

# RL11/RLV11,RL01

CONTROLLER TEST PART 1  
MD-11-DZRLA-A

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

DEC 1977  
**digital**  
MADE IN USA

The microfiche card contains a grid of 150 small frames, arranged in 10 columns and 15 rows. Each frame contains a small, illegible image or diagram, likely representing a test procedure or data for the RL11/RLV11,RL01 controller. The frames are arranged in a regular grid pattern across the left side of the card.

11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZRLA-A-D  
PRODUCT NAME: RL11/RLV11 RLO1 CONTROLLER TEST (PART 1)  
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 1) IS A PDP-11 (LSI-11) BASED PROGRAM THAT WILL TEST THE CONTROLLER. IT STARTS BY TESTING BASIC INTERFACE LOGIC, REGISTER MANIPULATION AND FUNCTIONALITY WHICH INCLUDES NOOP, GET STATUS, READ HEADERS AND SEEK OPERATIONS. 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)  
 : - 8 RLO1 DRIVES  
 1 - 8 RLO1K CARTRIDGES WITH BAD SECTOR FILE  
 KW11P, KW11L (OPTIONAL)  
 LINEPRINTER (OPTIONAL)

#### 1.2.2 SOFTWARE REQUIREMENTS

MAINDEC-11-DZR\_A-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 PROGRAM:

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

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

## 2.3.1.1 TESTS SWITCH (/TESTS:&lt;TEST-LIST&gt;)

<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:&lt;PASS-CNT&gt;)

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DEIRED 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:&lt;FLAG-LIST&gt;)

<FLAG-LIST> IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR
IER	INHIBIT ERROR REPORTING
IBE	INHIBIT BASIC ERROR REPORTS
IXE	INHIBIT EXTENDED ERROR REPORTS
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER
PNT	PRINT NUMBER OF TEST BEING EXECUTED
BOE	BELL ON ERROR
JAM	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.

SEQ 0005

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

<INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS. SEE EXAMPLE AT END OF 2.3.1.

#### 2.3.1.5 EFFECT OF COMMAND

THE EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE PARAMETER DIALOGUE, THE SOFTWARE PARAMETER DIALOGUE, AND THEN THE DIAGNOSTIC TESTS THEMSELVES.

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:HGE=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

```
*****
RESTART/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.

SEQ 0006

### 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 WHI 4 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 (&lt;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

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

## 2.3.5.1 TESTS, PASS, AND FLAGS SWITCHES

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

## 2.3.5.2 EFFECT OF COMMAND

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
HICORE 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

-----

\*\*\*\*\*  
 DROPPED 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

-----

\*\*\*\*\*  
 PRINT  
 \*\*\*\*\*

### 2.3.8.1 EFFECT OF COMMAND

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

### 2.3.9 DISPLAY COMMAND

-----

\*\*\*\*\*  
 DISPLAY UNITS: <UNIT-LIST>  
 \*\*\*\*\*

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

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

### 2.3.9.2 EFFECT OF COMMAND

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

### 2.3.10 FLAGS COMMAND

-----

\*\*\*\*\*  
 FLAGS)  
 \*\*\*\*\*

#### 2.3.10.1 EFFECT OF COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

### 2.3.11 ZFLAGS COMMAND

-----

\*\*\*\*\*  
 ZFLAGS)  
 \*\*\*\*\*

#### 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 ONE OF THE THREE OPERATOR DIALOGS (HARDWARE, HARDWARE, OR SOFTWARE QUESTIONS) CAUSES TC 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 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 RL11 CONTROLLER.

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (O) 330?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

BR LEVEL (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 (0) 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

#### 2.3.15 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

-----

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

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

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

# L01

SEQ 0011

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 A CONSTANT 75 IN TABLES 21 THRU 64, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 21,22,23,...,49 IN TABLES 21 THRU 49, AND GETS A 49 IN SLOT 50, AND GETS THE VALUES 51,52,53,...,64 IN TABLES 51 THRU 64. SLOT THREE GETS THE VALUE 77 IN TABLES 21 THRU 64.

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

## 2.4 EXECUTION TIMES

ONE PASS OF THE PROGRAM TAKES APPROXIMATELY 45 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

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

## 4.0 PERFORMANCE AND PROGRESS REPORTS

### 4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

### 4.2 PROGRESS REPORTS



THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

SEQ 0013

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)  
 BIT 11 - DATA LATE (WITH BIT 10 CLEAR)  
 BIT 10 - HEADER CRC (WITH BIT 10 SET)  
 BIT 9 - DATA CRC (WITH BIT 10 CLEAR)  
 BIT 8 - OPERATION INCOMPLETE  
 BIT 7/8 - 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

RAMP - MULTIPURPOSE REGISTER

-----

FOR READ/WRITE FUNCTION

-----

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

FOR REHD 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

C02

SEQ 0015

## 6.0 TEST SUMMARIES

## TEST 1 - RLCS ADDRESSABILITY

THIS TEST WILL CHECK THAT THE CONTROL AND STATUS REGISTER CAN BE ACCESSED. IF THE REGISTER CANNOT BE ACCESSED THE PROCESSOR WILL TRAP TO LOCATION 4, WHICH IS SET UP TO HANDLE THE TRAP.

## TEST 2 - RLBA ADDRESSABILITY

THIS TEST WILL CHECK THAT THE BUS ADDRESS REGISTER CAN BE ACCESSED. IF THE REGISTER CANNOT BE ACCESSED THE PROCESSOR WILL TRAP TO LOCATION 4, WHICH IS SET UP TO HANDLE THE TRAP.

## TEST 3 - RLDA ADDRESSABILITY

THIS TEST WILL CHECK THAT THE DISK ADDRESS REGISTER CAN BE ACCESSED. IF THE REGISTER CANNOT BE ACCESSED THE PROCESSOR WILL TRAP TO LOCATION 4, WHICH IS SET UP TO HANDLE THE TRAP.

## TEST 4 - RLMP ADDRESSABILITY

THIS TEST WILL CHECK THAT THE MULTIPURPOSE REGISTER CAN BE ACCESSED. IF THE REGISTER CANNOT BE ACCESSED THE PROCESSOR WILL TRAP TO LOCATION 4, WHICH IS SET UP TO HANDLE THE TRAP.

## TEST 5 - READ WRITE OF RLCS

THIS TEST WILL ATTEMPT TO WRITE RLCS BITS 9-1 AND READ THEM BACK. WALKING AND GROWING 0'S AND 1'S ARE USED. BIT 7 (CONTROLLER READY) IS ALWAYS WRITTEN AS A 1 SO NOT TO INITIATE A FUNCTION. BITS 15, 14 AND 0 ARE TREATED AS DON'T CARE FOR THIS TEST.

## TEST 6 - READ WRITE OF RLBA

THIS TEST WILL ATTEMPT TO WRITE RLBA BITS 15-0 AND READ THEM BACK. WALKING AND GROWING 0'S AND 1'S ARE USED. BIT 0 ON A RL11 SHOULD ALWAYS COME BACK AS A 0, WHILE ON AN RLV11 IT IS LOADABLE.

## TEST 7 - READ WRITE OF RLDA

THIS TEST WILL ATTEMPT TO WRITE RLDA BITS 15-0 AND READ THEM BACK. WALKING AND GROWING 0'S AND 1'S ARE USED.

## TEST 8 - BIS OF RLCS

THIS TEST WILL USE THE 11 INSTRUCTION "BIS" TO SHOW THAT A READ-MODIFY-WRITE SEQUENCE OF THE RLCS WORKS. BITS 9-1 ARE

USED. BIT SETTING IN WALKING AND GROWING 0'S AND 1'S. BIT 7 (CONTROLLER READY) IS ALWAYS SET. BITS 15, 14 AND 1 ARE DON'T CARES.

## TEST 9 - BIC OF RLCS

THIS TEST WILL USE THE 11 INSTRUCTION "BIC" TO SHOW THAT A READ-MODIFY-WRITE SEQUENCE OF THE RLCS WORKS. BITS 9-1 ARE USED. BIT CLEARING IN WALKING AND GROWING 0'S AND 1'S. BIT 7 (CONTROLLER READY) IS ALWAYS SET. BITS 15, 14 AND 1 ARE DON'T CARES.

## TEST 10 - BIS OF RLBA

THIS TEST WILL USE THE 11 INSTRUCTION "BIS" TO SHOW THAT A READ-MODIFY-WRITE SEQUENCE OF THE RLBA WORKS. BITS 15-0 ARE BIT SET USING GROWING AND WALKING 0'S AND 1'S. BIT 0 CANNOT SET ON A RL11, BUT CAN ON A RLV11.

## TEST 11 - BIC OF RLBA

THIS TEST WILL USE THE 11 INSTRUCTION "BIC" TO SHOW THAT A READ-MODIFY-WRITE SEQUENCE OF THE RLBA WORKS. BITS 15-0 ARE BIT CLEARED USING GROWING AND WALKING 0'S AND 1'S.

## TEST 12 - BIS OF RLDA

THIS TEST WILL USE THE 11 INSTRUCTION "BIS" TO SHOW THAT A READ-MODIFY-WRITE SEQUENCE OF THE RLDA WORKS. BITS 15-0 ARE BIT SET USING GROWING AND WALKING 0'S AND 1'S.

## TEST 13 - BIC OF RLDA

THIS TEST WILL USE THE 11 INSTRUCTION "BIC" TO SHOW THAT A READ-MODIFY-WRITE SEQUENCE OF THE RLDA WORKS. BITS 15-0 ARE BIT CLEARED USING GROWING AND WALKING 0'S AND 1'S.

## TEST 14 - BUS RESET OF RLCS

THIS TEST WILL VERIFY THAT THE BUS RESET OF THE PROCESSOR WILL CLEAR ALL BITS OF THE RLCS WITH THE EXCEPTION OF BIT 7 (CONTROLLER READY), BIT 0 (DRIVE READY) AND BIT 15 (COMPOSITE ERROR) IF BIT 14 (DRIVE ERROR) IS SET.

## TEST 15 - BUS RESET OF RLBA

THIS TEST WILL VERIFY THAT THE BUS RESET OF THE PROCESSOR WILL CLEAR ALL BITS OF THE RLBA.

## TEST 16 - BUS RESET OF RLDA

THIS TEST WILL VERIFY THAT THE BUS RESET OF THE PROCESSOR WILL



CLEAR ALL BITS OF THE RLDA.

SEG 0019

TEST 17 - UNIQUENESS OF RLCS

THIS TEST WILL VERIFY THAT WHEN THE RLCS (XXXXX0) IS ADDRESSED ONLY THAT REGISTER IS EFFECTED. BOTH THE RLBA AND THE RLDA ARE SET UP WITH KNOWN DATA, THE RLDA IS WRITTEN, THEN THE RLBA AND RLDA ARE VERIFIED THAT THEY DID NOT CHANGE.

TEST 18 - UNIQUENESS OF RLBA

THIS TEST WILL VERIFY THAT WHEN THE RLBA (XXXXX2) IS ADDRESSED ONLY THAT REGISTER IS EFFECTED. BOTH THE RLCS AND RLDA ARE WRITTEN WITH KNOWN DATA, THE RLBA IS WRITTEN, THEN THE RLCS AND RLDA ARE VERIFIED THAT THEY DID NOT CHANGE.

TEST 19 - UNIQUENESS OF RLDA

THIS TEST WILL VERIFY THAT WHEN THE RLDA (XXXXX4) IS ADDRESSED ONLY THAT REGISTER IS EFFECTED. BOTH THE RLCS AND RLBA ARE WRITTEN WITH KNOWN DATA, THE RLDA IS WRITTEN, THEN THE RLCS AND RLBA ARE VERIFIED THAT THEY DID NOT CHANGE.

TEST 20 - UNIQUENESS OF RLMP

THIS TEST WILL VERIFY THAT WHEN THE RLMP (XXXXX6) IS ADDRESSED ONLY THAT REGISTER IS EFFECTED. THE RLCS, RLBA AND RLDA ARE WRITTEN WITH KNOWN DATA, THE RLMP IS WRITTEN, THEN THE RLCS, RLBA AND RLDA ARE VERIFIED THAT THEY DID NOT CHANGE.

TEST 21 - NOOP FUNCTION

THIS TEST WILL VERIFY THE OPERATION OF THE NOOP (0) FUNCTION ON PDP-11'S ONLY. SINCE ON AN LSI-11 IT IS A MAINTENANCE FUNCTION. THE ABILITY OF CONTROLLER READY TO RESET AND NO ERRORS ARE CHECKED.

TEST 22 - TEST NOOP DOES NOTHING

THIS TEST WILL CHECK THAT THE NOOP FUNCTION WILL NOT DISTURB ANY REGISTERS OF THE CONTROLLER.

TEST 23 - TEST OF INTERRUPT

THIS TEST WILL CAUSE AN INTERRUPT FROM THE CONTROLLER USING NOOP (RL11 ONLY) TO CHECK THE INTERRUPT LOGIC AND VECTOR.

TEST 24 - TEST PRIORITY BR LEVEL

THIS TEST WILL CHECK THAT THE PROPER PRIORITY IS ON THE BOARD. WE VERIFY THAT ABOVE THE LEVEL THE BOARD WILL NOT INTERRUPT AND BELOW IT, IT WILL.

## TEST 25 - GET STATUS FUNCTION

THIS TEST WILL VERIFY THAT THE GET STATUS FUNCTION (2) WILL COMPLETE CORRECTLY. THE RLDA IS SET UP AND GET STATUS IS ISSUED. CONTROLLER READY IS CHECKED AS WELL AS ERROR BITS. (FIRST TEST A DRIVE MUST BE PRESENT.)

## TEST 26 - GET STATUS FUNCTION INTERRUPT

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

## TEST 27 - GET STATUS FUNCTION GENERATES OPI

THIS TEST WILL PROVE THE ABILITY FOR OPI (OPERATION INCOM) TO SET AND THAT THE DRIVE COMMAND IS BEING TRANSMITTED CORRECTLY. THE COMMAND WORD (RLDA) IS SET UP WITH THE MARKER BIT ONLY. AN OPI IS EXPECTED TO RESULT, THIS IS CHECKED.

## TEST 28 - OPI UNDER INTERRUPT

THIS TEST WILL CHECK THE ABILITY OF AN OPI TO CAUSE AN INTERRUPT TO OCCUR. WE SEND ONLY THE MARKER BIT WITH THE GET STATUS COMMAND AND EXPECT AN OPI ERROR.

## TEST 30 - READ HEADER FUNCTION INTERRUPT

THIS TEST WILL CHECK THE ABILITY OF THE READ HEADER FUNCTION TO INTERRUPT ON COMPLETION.

## TEST 31 - REPEATED RD HDRS YIELD SAME CYL AND HD

THIS TEST WILL CHECK THAT ON REPEATED READ HEADERS THE CYLINDER AND HEAD BITS OF THE HEADER WORD (RLMP) ARE ALWAYS THE SAME.

## TEST 32 - CHECK OF HEADER CRC

THIS TEST WILL VERIFY THE HEADER CRC THAT FOLLOWS THE TWO HEADER WORDS IS ACTUALLY THE CORRECT CRC-16 CALCULATION OF THE TWO HEADER WORDS.

## TEST 33 - CHECK CONSECUTIVE HEADERS

THIS TEST WILL CHECK THAT HEADERS ARE CONSECUTIVE.

## TEST 34 - SEEK FUNCTION

THIS TEST WILL CHECK THE SEEK FUNCTION (3) TO RESET CONTROLLER READY AND POST NO ERRORS. COMMAND WORD IS LOADED WITH A ONE CYLINDER FORWARD SEEK.

## TEST 35 - CHECK DRIVE READY ON SEEK

THIS TEST WILL CHECK THAT DRIVE READY CLEARS AND RESETS ON ISSUANCE OF A SEEK COMMAND.

## TEST 36 - SEEK FUNCTION INTERRUPT

THIS TEST WILL CHECK THE ABILITY OF A SEEK COMMAND TO GENERATE AN INTERRUPT ON CONTROLLER READY RESETTING AND NOT ONE ON DRIVE READY RESETTING.

## TEST 37 - TEST DIFFERENCE WORD TRANSMISSION

THIS TEST WILL TRY TO VERIFY THAT BITS 14-7, 6, 2, 0 OF THE COMMAND WORD GET TRANSMITTED CORRECTLY. WE ISSUE SEEKS FROM TRACK 0 WITH COMMAND WORDS OF WALKING AND GROWING 0'S AND 1'S. ALL SEEKS ARE VERIFIED WITH A READ HEADER AND RETURN TO TRACK 0 BEFORE NEXT PATTERN IS ISSUED.

## TEST 38 - VERIFY HEAD SELECT 0 VIA RD HEADER

THIS TEST WILL VERIFY THAT HEAD 0 CAN BE SELECTED AND READ VIA READ HEADER.

## TEST 39 - VERIFY HEAD SELECT 1 VIA RD HEADER

THIS TEST WILL VERIFY THAT HEAD 1 CAN BE SELECTED AND READ VIA READ HEADER.

## TEST 40 - VERIFY HEAD SELECT 0 VIA GET STATUS

THIS TEST WILL VERIFY THE WORD RETURNED TO THE RLMP BY A GET STATUS CONTAINS THE RIGHT HEAD SELECT.

## TEST 41 - VERIFY HEAD SELECT 1 VIA GET STATUS

THIS TEST WILL VERIFY THE WORD RETURNED TO THE RLMP BY A GET STATUS CONTAINS THE RIGHT HEAD SELECT.

## TEST 42 - TEST TIME AT WHICH DP WD GETS

THIS TEST WILL CHECK THAT THE DIFFERENCE WORD (RLDA) ACTUALLY DOES GET TRANSMITTED PRIOR TO CONTROLLER READY RESETTING. THIS IS DONE BY ISSUING A SEEK, WAITING FOR CONTROLLER READY AND RE-LOADING THE RLDA. THE SEEK IS THEN VERIFIED TO SEE IF IT IS CORRECT.

## TEST 43 - EXTENSIVE CHECK OF CRC

THIS TEST WILL MORE EXTENSIVELY CHECK THE CRC LOGIC BY POSI-

TIJNING AT DIFFERENT POINTS ON THE PACK AND CHECKING THAT THE  
HEADER CRC RECEIVED IS CORRECT.

SEG 0021

TEST 44 - VERIFY GET STATUS WHILE DRDY IS LOW

THIS TEST WILL CHECK THE ABILITY TO PERFORM A GET STATUS WHILE  
THE DRIVE IS SEEKING.

7.0 PROGRAM LISTING  
-----







```

2825
2826
2827
2828 002100          DEVREG
      (5) 002100      000000      .WORD      0
      (2) 002102      000001      .BLKW
2829
2830 002104          DEVTYP <RLO1>
      (3) 002104      046122      030460      000      .ASCIZ  @RLO1@
      (2)          002112      .EVEN
2831 002112          BGNMOD  GLBEGAT
2832
2833 002112          EQUALS

```

BIT DIFINITIONS

```

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

```

```

BIT15== 100000
BIT14== 40000
BIT13== 20000
BIT12== 10000
BIT11== 4000
BIT10== 2000
BIT09== 1000
BIT08== 400
BIT07== 200
BIT06== 100
BIT05== 40
BIT04== 20
BIT03== 10
BIT02== 4
BIT01== 2
BIT00== 1

BIT9== BIT09
BIT8== BIT08
BIT7== BIT07
BIT6== BIT06
BIT5== BIT05
BIT4== BIT04
BIT3== BIT03
BIT2== BIT02
BIT1== BIT01
BIT0== BIT00

```

EVENT FLAG DEFINITIONS

```

EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
EF16:EF01 AVAILABLE FOR PROGRAM USE

```

```

000040
000037
000036
000035
000034

000020
000017

```

```

EF.START== 32.
EF.RESTART== 31.
EF.CONTINUE== 30.
EF.NEW== 29.
EF.PWR== 28.

EF15== 16.
EF15== 15.

```

```

: START COMMAND WAS ISSUED
: RESTART COMMAND WAS ISSUED
: CONTINUE COMMAND WAS ISSUED
: A NEW PASS HAS BEEN STARTED
: A POWER-FAIL/POWER-UP OCCURRED

```



```

2864 ;OFFSET FOR HARDWARE P-TABLE
2865
2866 000000 CSR=0
2867 000002 VECT=2
2868 000004 PRIOR=4
2869 000006 DRBT=6
2870 000010 CNT=10
2871
2872 ;OFFSET FOR SOFTWARE P-TABLE
2873
2874 000000 DLT=0
2875 000002 ELT=2
2876 000004 SIZE=4
2877
2878 002112 ENDMOD
2879
2880 002112 BGNMOD GLBDAT
2881
2882 .SBTTL GLOBAL DATA
2883
2884 002112 000000 PWRFLG: .WORD 0
2885 002114 000000 UUT: .WORD 00
2886 002116 000000 UNITST: .WORD 00
2887 002120 000000 RLCS: .WORD 00
2888 002122 000000 RLGA: .WORD 00
2889 002124 000000 RLDA: .WORD 00
2890 002126 000000 RLMP: .WORD 00
2891 002130 000000 BCSR: .WORD 00
2892 002132 000000 BPRIOR: .WORD 00
2893 002134 000000 BVEC: .WORD 00
2894 002136 000000 DRIVE: .WORD 00 ;DRIVE UNDER TEST
2895 002140 000000 B.CS: .WORD 00
2896 002142 000000 B.BA: .WORD 00
2897 002144 000000 B.DA: .WORD 00
2898 002146 000000 B.MP: .WORD 00
2899 002150 000000 DERFLG: .WORD 00
2900 002152 000000 F.CS: .WORD 00
2901 002154 000000 F.BA: .WORD 00
2902 002156 000000 F.DA: .WORD 00
2903 002160 000000 F.MP: .WORD 00
2904 002162 000000 F.MP1: .WORD 00
2905 002164 000000 F.MP2: .WORD 00
2906 002166 000000 PFLG: .WORD 00 ;PROCESSOR TYPE 0=UNIBUS 1=Q-BUS
2907 002170 000000 TRPFLG: .WORD 00
2908 002172 000000 INTFLG: .WORD 00 ;INTERRUPT OCCURANCE FLAG
2909 002174 000000 LDCSR: .WORD 00 ;LOCATION TO FORM RLCS
2910 002176 000077 SECMSK: .WORD 77 ;MASK OUT SECTOR
2911 002200 120001 XPOLY: .WORD 120001
2912 002202 000004 ERRVEC: .WORD 4
2913 002204 000000 BCCFBK: .WORD 00
2914 002206 000000 CALBCC: .WORD 00 ;LOCATION USED BY "SIMBCC"
2915 002210 000000 TEMP2: .WORD 00 ;LOCATION USED BY "SIMBCC"
2916 002212 000000 TEMP3: .WORD 00 ;LOCATION USED BY "SIMBCC"
2917 002214 000000 TEMP4: .WORD 00 ;LOCATION USED BY "SIMBCC"
2918 002216 000000 TMPO: .WORD 00
2919 002220 000000 TMP1: .WORD 00

```

```

2920 002222 000000
2921 002224 000000
2922 002226 000000
2923 002230 000000
2924 002232 177700
2925 002234 000050
2926 002236 000047
2927 002240 000000
2928 002242 077600
2929 002244 000000
2930 002246 000000
2931 002250 000000
2932 002252 000000
2933 002254 000000
2934
2935
2936
2937
2938
2939 002256 000000
2940 002260 000001
2941 002262 000003
2942 002264 000007
2943 002266 000017
2944 002270 000037
2945 002272 000077
2946 002274 000177
2947 002276 000377
2948 002300 000777
2949 002302 001777
2950 002304 003777
2951 002306 007777
2952 002310 017777
2953 002312 037777
2954 002314 077777
2955 002316 177777
2956 002320 177776
2957 002322 177774
2958 002324 177770
2959 002326 177760
2960 002330 177740
2961 002332 177700
2962 002334 177600
2963 002336 177400
2964 002340 177000
2965 002342 176000
2966 002344 174000
2967 002346 170000
2968 002350 160000
2969 002352 140000
2970 002354 100000
2971
2972 002356 000000
2973 002360 000001
2974 002362 000002
2975 002364 000004

```

```

TMP2: .WORD 0
GDDAT: .WORD 0
BDDAT: .WORD 0
FIRST: .WORD 0 ; FIRST SECTOR READ
CYLSK: .WORD 177700 ; MASK CYLINDER AND HEAD SELECT
MXSEC1: .WORD 40. ; MAX SECTOR ADDRESS +1
MAXSEC: .WORD 39. ; MAX SECTOR ADDRESS
DWORD: .WORD 0 ; DIFFERENCE WORD (SEEK)
MAXCYL: .WORD 77600 ; MAXIMUM CYLINDER ADDRESS
SVHD: .WORD 0 ; SAVE CURRENT HEAD SELECT
ERRLMT: .WORD 0
WHY: .WORD 0 ; REASON FOR DROP IN AUTOSIZE
T.CNTRL: .WORD 0
TMPFNC: .WORD 0

```

```

.SBTL PATTERNS FOR REGISTER R'W
; PATTERNS USED FOR LOADING/READING REGISTERS

```

```

BEGPAT: 0 ; GROWING 1
1
3
7
17
37
77
177
377
777
1777
3777
7777
17777
37777
77777
177777
177776 ; GROWING 0
177774
177770
177760
177740
177700
177600
177400
177000
176000
174000
170000
160000
140000
100000
000000
1 ; WALKING 1
2
4

```

2976	002366	000010	10
2977	002370	000020	20
2978	002372	000040	40
2979	002374	000100	100
2980	002376	000200	200
2981	002400	000400	400
2982	002402	001000	1000
2983	002404	002000	2000
2984	002406	004000	4000
2985	002410	010000	10000
2986	002412	020000	20000
2987	002414	040000	40000
2988	002416	100000	100000
2989	002420	177777	177777
2990	002422	177776	177776
2991	002424	177775	177775
2992	002426	177773	177773
2993	002430	177767	177767
2994	002432	177757	177757
2995	002434	177737	177737
2996	002436	177677	177677
2997	002440	177577	177577
2998	002442	177377	177377
2999	002444	176777	176777
3000	002446	175777	175777
3001	002450	173777	173777
3002	002452	167777	167777
3003	002454	157777	157777
3004	002456	137777	137777
3005	002460	077777	077777
3006	002462	177777	177777
3007	002464	000000	000000

;WALKING 0

ENDPAT: 000000

.SBTTL PATTERNS FOR DIFFERENCE WORD

3010			
3011	002466	000200	
3012	002470	000400	
3013	002472	001000	
3014	002474	002000	
3015	002476	004000	
3016	002500	010000	
3017	002502	020000	
3018	002504	040000	
3019	002506	077600	
3020	002510	077400	
3021	002512	076600	
3022	002514	075600	
3023	002516	073600	
3024	002520	067600	
3025	002522	057600	
3026	002524	037600	
3027	002526	077600	
3028	002530	000200	
3029	002532	000600	
3030	002534	001600	
3031	002536	003600	

SKLST: .WORD BIT7  
 .WORD BIT8 ;SHIFTING 1  
 .WORD BIT9  
 .WORD BIT10  
 .WORD BIT11  
 .WORD BIT12  
 .WORD BIT13  
 .WORD BIT14  
 .WORD 77600 ;SHIFTING 0  
 .WORD 77400  
 .WORD 76600  
 .WORD 75600  
 .WORD 73600  
 .WORD 67600  
 .WORD 57600  
 .WORD 37600  
 .WORD 77600  
 .WORD 200  
 .WORD 600 ;GROWING 1  
 .WORD 1600  
 .WORD 3600

3032 002540 007600  
 3033 002542 017600  
 3034 002544 037600  
 3035 002546 077600  
 3036 002550 077400  
 3037 002552 077000  
 3038 002554 076000  
 3039 002556 074000  
 3040 002560 070000  
 3041 002562 060000  
 3042 002564 040000  
 3043 002566 000000

QUAMAX: .WORD 7600  
 .WORD 17600  
 HALMAX: .WORD 37600  
 .WORD 77600  
 .WORD 77400  
 .WORD 77000  
 .WORD 76000  
 .WORD 74000  
 .WORD 70000  
 .WORD 60000  
 .WORD 40000  
 SEND: .WORD 00000

:GROWING 0

3044  
 3045  
 3046  
 3047 002570 000000  
 3048 002572 000002  
 3049 002574 000004  
 3050 002576 000010  
 3051 002600 000020  
 3052 002602 000040  
 3053 002604 000100  
 3054 002606 000400  
 3055 002610 001000  
 3056 002612 001576  
 3057 002614 001574  
 3058 002616 001570  
 3059 002620 001560  
 3060 002622 001540  
 3061 002624 001500  
 3062 002626 001400  
 3063 002630 001576  
 3064 002632 001574  
 3065 002634 001566  
 3066 002636 001556  
 3067 002640 001536  
 3068 002642 001436  
 3069 002644 001136  
 3070 002646 000076  
 3071 002650 000006  
 3072 002652 000016  
 3073 002654 000036  
 3074 002656 000076  
 3075 002660 000176  
 3076 002662 000576  
 3077 002664 001576  
 3078 002666 000000  
 3079 002670 000240

:PATTERNS FOR TEST OF RLCS

CSPAT: .WORD 0  
 .WORD BIT1  
 .WORD BIT2  
 .WORD BIT3  
 .WORD BIT4  
 .WORD BIT5  
 .WORD BIT6  
 .WORD BIT8  
 .WORD BIT9  
 .WORD 1576  
 .WORD 1574  
 .WORD 1570  
 .WORD 1560  
 .WORD 1540  
 .WORD 1500  
 .WORD 1400  
 .WORD 1576  
 .WORD 1574  
 .WORD 1566  
 .WORD 1556  
 .WORD 1536  
 .WORD 1436  
 .WORD 1136  
 .WORD 76  
 .WORD 6  
 .WORD 16  
 .WORD 36  
 .WORD 76  
 .WORD 176  
 .WORD 576  
 .WORD 1576  
 .WORD 0  
 HDRBUF: .BLKW 160.  
 ENDMOD

:SHIFTING 1

:GROWING 0

:SHIFT 0

:GROWING 1

3081  
 3082 003370  
 3083  
 3084  
 3088 003370 047516 041440 047117  
 3089 003406 047516 042040 044522  
 3090 003431 040 051104 000126

BGNMOD GLBTXT  
 .SBTTL GLOBAL TEXT

NORES: .ASCIZ /NO CONTROLLER/  
 NODRY: .ASCIZ /NO DRIVE CONNECTED/  
 DEMES: .ASCIZ /DRV/

3091	003436	047040	046530	000	NXMMES:	.ASCIZ	/NXM/
3092	003443	040	050117	000111	OPIMES:	.ASCIZ	/OPI/
3093	003450	044040	051103	000103	HCRCMES:	.ASCIZ	/HCRC/
3094	003456	044040	043116	000	HNFMES:	.ASCIZ	/HNF/
3095	003463	040	041504	000113	DCKMES:	.ASCIZ	/DCK/
3096	003470	042040	052114	000	DLTMES:	.ASCIZ	/DLT/
3097	003475	015	000012		MSCRLF:	.ASCIZ	<15><12>
3098	003480	000015			LF:	.ASCIZ	<15>
3099	003485	041440	046517	000120	COMP:	.ASCIZ	/COMP/
3100	003490	047506	041522	042105	OPIERR:	.ASCIZ	/FORCED OPI(GET STATUS) CAUSED OTHER ERRORS/
3101	003563	116	047517	020120	NOPMES:	.ASCIZ	/NOOP OPERATION-FLAG MODE/
3102	003614	047516	050117	047440	NOPINT:	.ASCIZ	/NOOP OPERATION-INTR. MODE/
3103	003646	051127	052111	020105	WCKM:	.ASCIZ	/WRITE CHECK OPERATION-FLAG MODE/
3104	003706	051127	052111	020105	WCKINT:	.ASCIZ	/WRITE CHECK OPERATION-INTR. MODE/
3105	003747	122	040505	020104	RDMES:	.ASCIZ	/READ HEADER OPERATION-FLAG MODE/
3106	004007	122	040505	020104	RHDINT:	.ASCIZ	/READ HEADER OPERATION-INTR. MODE/
3107	004050	042523	045505	047440	SEKMES:	.ASCIZ	/SEEK OPERATION-FLAG MODE/
3108	004101	123	042505	020113	SEKINT:	.ASCIZ	/SEEK OPERATION-INTR. MODE/
3109	004133	107	052105	051440	GSTMES:	.ASCIZ	/GET STATUS OPERATION-FLAG MODE/
3110	004172	042507	020124	052123	GSTINT:	.ASCIZ	/GET STATUS OPERATION-INTR. MODE/
3111	004231	103	035123	000040	ARLCS:	.ASCIZ	/CS: /
3112	004236	041040	035101	000040	ARLBA:	.ASCIZ	/BA: /
3113	004244	042040	035101	000040	ARLDA:	.ASCIZ	/DA: /
3114	004252	046440	035120	000040	ARLMP:	.ASCIZ	/MP: /
3115	004260	042502	047506	042522	BEREG:	.ASCIZ	/BEFORE COMMAND: /
3116	004301	124	046511	020105	AFREG:	.ASCIZ	/TIME OF ERROR: /
3117	004322	047503	052116	047522	CRTIM:	.ASCIZ	/CONTROLLER TIMED OUT/
3118	004347	104	044522	042526	ORTIM:	.ASCIZ	/DRIVE READY TIMED OUT/
3119	004375	103	047101	047040	EM1:	.ASCIZ	/CAN NOT ADDRESS RLCS/
3120	004422	040503	020116	047516	EM2:	.ASCIZ	/CAN NOT ADDRESS RLBA/
3121	004447	103	047101	047040	EM3:	.ASCIZ	/CAN NOT ADDRESS RLDA/
3122	004474	040503	020116	047516	EM4:	.ASCIZ	/CAN NOT ADDRESS RLMP/
3123	004521	122	041514	020123	EM5:	.ASCIZ	%RLCS READ/WRITE ERROR (BIT 0 DON'T CARE) %
3124	004572	046122	040502	051040	EM6:	.ASCIZ	%RLBA READ/WRITE ERROR %
3125	004620	046122	040504	051040	EM7:	.ASCIZ	%RLDA READ/WRITE ERROR %
3126	004646	050117	020111	047527	EM11:	.ASCIZ	/OPI WOULD NOT GENERATE INTERRUPT/
3127	004707	116	020117	047111	EM13:	.ASCIZ	/NO INTERRUPT FROM NOCP(O)/
3128	004741	116	047517	024120	EM14:	.ASCIZ	/NOOP(O) MODIFIED RLMP/
3129	004767	116	047517	024120	EM15:	.ASCIZ	/NOOP(O) MODIFIED RLBA/
3130	005015	116	047517	024120	EM16:	.ASCIZ	/NOOP(O) MODIFIED RLDA/
3131	005043	111	052116	051105	EM17:	.ASCIZ	/INTERRUPT PRIORITY FAILURE/
3132	005076	042507	020124	052123	EM30:	.ASCIZ	/GET STATUS WOULD NOT INTERRUPT/
3133	005135	122	046514	020120	EM32:	.ASCIZ	/RLMP CONTAINED WRONG STATUS/
3134	005171	117	044520	042040	EM33:	.ASCIZ	/OPI DID NOT SET-GSTAT WITHOUT GS BIT/
3135	005236	050117	020111	044504	EM34:	.ASCIZ	/OPI DID NOT SET-GSTAT WITHOUT GS AND MK BITS/
3136	005313	122	040505	020104	EM37:	.ASCIZ	/READ HEADER WOULD NOT INTERRUPT/
3137	005353	102	042101	041440	EM41:	.ASCIZ	/BAD CYLINDER OR HEAD SELECT IN REPEATED READ HEADER TEST/
3138	005444	040502	020104	042510	EM42:	.ASCIZ	/BAD HEADER CRC ON READ HEADER/
3139	005502	042523	052103	051117	EM43:	.ASCIZ	/SECTOR ADDRESS OUT OF SEQUENCE DURING CONSECUTIVE READ HEADERS/
3140	005601	127	044522	044524	EM44:	.ASCIZ	/WRITING RLMP MODIFIED RLCS/
3141	005634	051127	052111	047111	EM45:	.ASCIZ	/WRITING RLMP MODIFIED RLBA/
3142	005667	127	044522	044524	EM46:	.ASCIZ	/WRITING RLMP MODIFIED RLDA/
3143	005722	042523	045505	053440	EM47:	.ASCIZ	/SEEK WOULD NOT INTERRUPT/
3144	005733	104	044522	042526	EM52:	.ASCIZ	/DRIVE READY CAUSED EXTRANEIOUS INTERRUPT/
3145	006023	102	042101	051440	EM54:	.ASCIZ	/BAD SEEK-TEST OF DIFFERENCE WORD/
3146	006062	040502	020104	042510	EM55:	.ASCIZ	/BAD HEAD SELECT VIA RD HDR/



3147	006115	102	042101	044040	EM56:	.ASCIZ	/BAD HEAD SELECT VIA GET STATUS/
3148	006154	047514	042101	047111	EM57:	.ASCII	/LOADING RLDA BEFORE DRIVE READY ON SEEK <<15>> <<12>
3149	006225	104	044522	042526		.ASCIZ	/DRIVE READY DID NOT SET/
3150	006255	102	052111	051440	EM61:	.ASCIZ	/BIT SET INSTRUCTION ON RLCS YIELDED WRONG RESULT/
3151	006336	044502	020124	046103	EM62:	.ASCIZ	/BIT CLEAR INSTRUCTION ON RLCS YIELDED WRONG RESULT/
3152	006421	102	052111	051440	EM63:	.ASCIZ	/BIT SET INSTRUCTION ON RLBA YIELDED WRONG RESULT/
3153	006502	044502	020124	046103	EM64:	.ASCIZ	/BIT CLEAR INSTRUCTION ON RLBA YIELDED WRONG RESULT/
3154	006565	102	052111	051440	EM65:	.ASCIZ	/BIT SET INSTRUCTION ON RLDA YIELDED WRONG RESULT/
3155	006646	044502	020124	046103	EM66:	.ASCIZ	/BIT CLEAR INSTRUCTION ON RLDA YIELDED WRONG RESULT.
3156	006731	102	051525	051040	EM67:	.ASCIZ	/BUS RESET DID NOT CLEAR RLCS/
3157	006766	052502	020123	042522	EM70:	.ASCIZ	/BUS RESET DID NOT CLEAR RLBA/
3158	007023	102	051525	051040	EM71:	.ASCIZ	/BUS RESET DID NOT CLEAR RLDA/
3159	007060	051127	052111	047111	EM72:	.ASCIZ	/WRITING RLCS MODIFIED RLBA/
3160	007113	127	044522	044524	EM73:	.ASCIZ	/WRITING RLCS MODIFIED RLDA/
3161	007146	051127	052111	047111	EM74:	.ASCIZ	/WRITING RLBA MODIFIED RLCS/
3162	007200	051127	052111	047111	EM75:	.ASCIZ	/WRITING RLBA MODIFIED RLDA/
3163	007232	051127	052111	047111	EM76:	.ASCIZ	/WRITING RLDA MODIFIED RLCS/
3164	007265	127	044522	044524	EM77:	.ASCIZ	/WRITING RLDA MODIFIED RLBA/
3165	007320	046122	051503	041440	EM101:	.ASCIZ	/RLCS CONTAINED FOLLOWING ERROR(S):
3166	007365	000170			EM102:	.BLKB	120.
3167							
3168		007556				.EVEN	
3169							
3170						ENDMOD	
3171						.SBTTL	GLOBAL ERRORS
3172							
3173						BGNMOD	GLBERR
3174							
3175						BGNMSG	ERRO
3176							
3177							
3178							
3179							
3180							
3181	007556	004737	010102			JSR	PC,LINE1
3182	007562	004737	010136			JSR	PC,LINE2
3183							
3184	007566	004537	012166			JSR	RS,CKERLT ;CHECK ERROR LIMIT
3185	007572					ENDMSG	
3186	007572				L10000:	EMT	C\$MSG
3187	007572	104023					
3188							
3189	007574				BGNMSG	ERR1	
3190							
3191	007574	004737	010102			JSR	PC,LINE1
3192							
3193	007600	004537	012166			JSR	RS,CKERLT ;CHECK ERROR LIMIT
3194	007604					ENDMSG	
3195	007604				L10001:	EMT	C\$MSG
3196	007604	104023					

3194					
3195	007606			BGNMSG	ERR2
3196					
3197	007606	004737	010102		JSR PC,LINE1
3198	007612				PRINTB #FRMT4,GDDAT,BDDAT
(9)	007612	013746	002226		MOV BDDAT,-(SP)
(8)	007616	013746	002224		MOV GDDAT,-(SP)
(7)	007622	012746	010560		MOV #FRMT4,-(SP)
(6)	007626	012746	000003		MOV #3,-(SP)
(5)	007632	010600			MOV SP,RO
(4)	007634	104014			EMT C\$PNTB
(4)	007636	062706	000010		ADD #10,SP
3199					
3200	007642	004537	012166		JSR R5,CKERLT ;CHECK ERROR LIMIT
3201	007646				ENDMSG
(3)	007646			L10002:	
(3)	007646	104023			EMT C\$MSG
3202					
3203	007650			BGNMSG	ERR3
3204					
3205	007650	004737	010102		JSR PC,LINE1
3206	007654	004737	010136		JSR PC,LINE2
3207	007660				PRINTB #FRMT5,TMPO,BDDAT,GDDAT
(10)	007660	013746	002224		MOV GDDAT,-(SP)
(9)	007664	013746	002226		MOV BDDAT,-(SP)
(8)	007670	013746	002216		MOV TMPO,-(SP)
(7)	007674	012746	010616		MOV #FRMT5,-(SP)
(6)	007700	012746	000004		MOV #4,-(SP)
(5)	007704	010600			MOV SP,RO
(4)	007706	104014			EMT C\$PNTB
(4)	007710	062706	000012		ADD #12,SP
3208					
3209	007714	004537	012166		JSR R5,CKERLT ;CHECK ERROR LIMIT
3210	007720				ENDMSG
(3)	007720			L10003:	
(3)	007720	104023			EMT C\$MSG
3211					
3212	007722			BGNMSG	ERR4
3213					
3214	007722	004737	010102		JSR PC,LINE1
3215	007726	004737	010136		JSR PC,LINE2
3216	007732				PRINTB #FRMT4,GDDAT,BDDAT
(9)	007732	013746	002226		MOV BDDAT,-(SP)
(8)	007736	013746	002224		MOV GDDAT,-(SP)
(7)	007742	012746	010560		MOV #FRMT4,-(SP)
(6)	007746	012746	000003		MOV #3,-(SP)
(5)	007752	010600			MOV SP,RO
(4)	007754	104014			EMT C\$PNTB
(4)	007756	062706	000010		ADD #10,SP
3217					
3218	007762	004537	012166		JSR R5,CKERLT ;CHECK ERROR LIMIT
3219	007766				ENDMSG
(3)	007766			L10004:	
(3)	007766	104023			EMT C\$MSG
3220					
3221	007770			BGNMSG	ERR5

H03

01 ERR MACY11 30.1046) 30-OCT-77 16:51 PAGE 84-1  
01 P11 05-OCT-77 10:41 GLOBAL ERRORS

SEQ 0033

007770	004737	010102	JSR	PC,LINE1	
007774	004537	012166	JSR	RS,CKERLT	:CHECK ERROR LIMIT
010000			ENDMSG		
010000	104023		L10005: EMT	C\$MSG	
010002			BGNMSG	ERR6	
010002	004737	010102	JSR	PC,LINE1	
010006	004737	010360	JSR	PC,LINE3	
010012	004737	010136	JSR	PC,LINE2	
010016			15: PRINTB	#FRMT99	
010016	012746	010613	MOV	#FRMT99, -(SP)	
010022	012746	000001	MOV	#1, -(SP)	
010026	010600		MOV	SP, R0	
010030	104014		EMT	C\$PNTB	
010032	062706	000004	ADD	#4, SP	
010036	004537	012166	JSR	RS,CKERLT	:CHECK ERROR LIMIT
010042			ENDMSG		
010042	104023		L10006: EMT	C\$MSG	
010044			BGNMSG	ERR7	
010044	004737	010102	JSR	PC,LINE1	
010050			PRINTB	#FRMT6, BDDAT	
010050	013746	002226	MOV	BDDAT, -(SP)	
010054	012746	010667	MOV	#FRMT6, -(SP)	
010060	012746	000002	MOV	#2, -(SP)	
010064	010600		MOV	SP, R0	
010066	104014		EMT	C\$PNTB	
010070	062706	000006	ADD	#6, SP	
010074	004537	012166	JSR	RS,CKERLT	
010100			ENDMSG		
010100	104023		L10007: EMT	C\$MSG	
010102			LINE1: PRINTB	#FRMT1, RLCS, B, DRIVE+1	
010102	005046		CLR	-(SP)	
010104	153716	002137	BISB	DRIVE+1, (SP)	
010110	013746	002120	MOV	RLCS, -(SP)	
010114	012746	010432	MOV	#FRMT1, -(SP)	
010120	012746	000003	MOV	#3, -(SP)	
010124	010600		MOV	SP, R0	
010126	104014		EMT	C\$PNTB	
010130	062706	000010	ADD	#10, SP	
010134	000207		RTS	PC	
010136			LINE2: PRINTB	#FRMT2, #BEREG, #APLCS, B, CS, #ARLBA, B, BA	
010136	012746	002142	MOV	B, BA, -(SP)	

01010142	012746	004236	MOV	#ARLBA, -(SP)
01010146	013746	002140	MOV	B. CS, -(SP)
01010152	012746	004231	MOV	#ARLCS, -(SP)
01010156	012746	004260	MOV	#BEREG, -(SP)
01010162	012746	010472	MOV	#FRMT2, -(SP)
01010166	012746	000006	MOV	#6, -(SP)
01010172	010600		MOV	SP, RO
01010174	104014		EMT	C\$PNTB
01010176	062706	000016	ADD	#16, SP
01020202			PRINTB	#FRMT2A, #ARLDA, B. DA, #ARLMP, B. MP
01020206	013746	002146	MOV	B. MP, -(SP)
01020208	012746	004252	MOV	#ARLMP, -(SP)
01020212	013746	002144	MOV	B. DA, -(SP)
01020216	012746	004244	MOV	#ARLDA, -(SP)
01020222	012746	010511	MOV	#FRMT2A, -(SP)
01020226	012746	000005	MOV	#5, -(SP)
01020232	010600		MOV	SP, RO
01020234	104014		EMT	C\$PNTB
01020236	062706	000014	ADD	#14, SP
01020242			PRINTB	#FRMT2, #AFREG, #ARLCS, E. CS, #ARLBA, E. BA
01020246	013746	002154	MOV	E. BA, -(SP)
01020252	012746	004236	MOV	#ARLBA, -(SP)
01020256	013746	002152	MOV	E. CS, -(SP)
01020262	012746	004231	MOV	#ARLCS, -(SP)
01020266	012746	004301	MOV	#AFREG, -(SP)
01020272	012746	010472	MOV	#FRMT2, -(SP)
01020276	012746	000006	MOV	#6, -(SP)
01030300	010600		MOV	SP, RO
01030302	104014		EMT	C\$PNTB
01030306	062706	000016	ADD	#16, SP
01030308			PRINTB	#FRMT2B, #ARLDA, E. DA, #ARLMP, E. MP, E. MP1, E. MP2
01030312	013746	002164	MOV	E. MP2, -(SP)
01030316	013746	002162	MOV	E. MP1, -(SP)
01030322	013746	002160	MOV	E. MP, -(SP)
01030326	012746	004252	MOV	#ARLMP, -(SP)
01030332	013746	002156	MOV	E. DA, -(SP)
01030336	012746	004244	MOV	#ARLDA, -(SP)
01030342	012746	010524	MOV	#FRMT2B, -(SP)
01030346	012746	000007	MOV	#7, -(SP)
01030350	010600		MOV	SP, RO
01030352	104014		EMT	C\$PNTB
01030356	062706	000020	ADD	#20, SP
01030360	000207		RTS	PC
01030360			LINE3: PRINTB	#FRMT3, #EM101
01030364	012746	007320	MOV	#EM101, -(SP)
01030370	012746	010553	MOV	#FRMT3, -(SP)
01030374	012746	000002	MOV	#2, -(SP)
01030376	010600		MOV	SP, RO
01040400	104014		EMT	C\$PNTB
01040404	062706	000006	ADD	#6, SP
01040404			PRINTB	#FRMT3, #EM102
01040410	012746	007365	MOV	#EM102, -(SP)
01040414	012746	010553	MOV	#FRMT3, -(SP)
01040420	012746	000002	MOV	#2, -(SP)
01040420	010600		MOV	SP, RO

```

(4) 010422 104014 EMT C$PNTB
010424 062706 ADD #6,SP
010430 000207 RTS PC

010432 040445 047503 052116 FRMT1: .ASCIZ //A%CONTROLLER: %06%A DRIVE: %01/
010472 047045 052045 052045 FRMT2: .ASCIZ //N%T%T%06%T%06/
010511 045 022524 033117 FRMT2A: .ASCIZ //T%06%T%06/
010524 052045 047445 022466 FRMT2B: .ASCIZ //T%06%T%06%A %06%A %06/
010553 045 022516 000124 FRMT3: .ASCIZ //N%T/
010560 047045 040445 054105 FRMT4: .ASCII //N%AEXP'D: %06%A REC'D: %06/
010613 045 000116 FRMT99: .ASCIZ //N/
010616 047045 040445 040514 FRMT5: .ASCIZ //N%ALAST: %06%A PRES: %06%A EXP'D: %06%N/
010667 045 022516 040501 FRMT6: .ASCIZ //N%AT PROCESSOR LEVEL %06%N/
010724 040445 051105 047522 FRMT11: .ASCIZ //A%ERROR LIMIT EXCEEDED-DROPPED%N/
010765 045 022516 042101 FRMT12: .ASCIZ //N%ADrive DID NOT RECOVER FROM POWER FAILURE%N/
011044 047045 052045 040445 FRMT13: .ASCIZ //N%T%A - WILL NOT TEST%N/

011076 .EVEN

011076 ENDMOD

011076 BGNMOD HPTCODE

011076 BGNHW
011076 000005 .WORD L10010-L$HW/2
011100 174400 .WORD 174400 :CSR
011102 000330 .WORD 330 :VECTOR
011104 000240 .WORD 240 :PRIORITY
011106 000000 .WORD 0 :DRIVE .BITS 8,9,10)
011110 000001 .WORD 1 :RL11 = 1, PLV11 = 0

011112 ENDSW
011112 L10010:

011112 ENDMOD

011112 BGNMOD SPTCODE

011112 BGNSW
011112 000003 .WORD L10011-L$SW/2

011114 000000 DROP: .WORD 0
011116 000012 MERLMT: .WORD 10.
011120 000000 T.SIZE: .WORD 0

011122 ENDSW
011122 L10011:

011122 ENDMOD

011122 BGNMOD DSPCODE

```



```

(3) 011254 104033      EMT      CSRESET
3326 011256 012700 000034  REDEF    #EF.PWR      ;POWER UP????
(3) 011256 012700 000034  MOV      #EF.PWR,RO
(3) 011262 104050      EMT      CSREFG
3327 011264 103004      BNCOMPLET NOPWR      ;NO,BRANCH
(2) 011264 103004      BCC      NOPWR
3328 011265 013737 002014 002112  MOV      LSUNIT,PWRFLG ;YES, SET POWER FLAG
3329 01127 000472      BR       CONT        ;GO TO CONTINUE POINT
3330 011276 012700 000037      NOPWR: REDEF    #EF.RESTART ;RESTART
(3) 011276 012700 000037  MOV      #EF.RESTART,RO
(3) 011302 104050      EMT      CSREFG
3331 011304 103410      BCOMPLET START
(2) 011304 103410      BCS      START
3332 011306 012700 000035      REDEF    #EF.NEW      ;NEW PASS????
(3) 011306 012700 000035  MOV      #EF.NEW,RO
(3) 011312 104050      EMT      CSREFG
3333 011314 103404      BCOMPLET START      ;YES, THEN RE INIT
(2) 011314 103404      BCS      START
3334 011316 012700 000040      REDEF    #EF.START    ;START???
(3) 011316 012700 000040  MOV      #EF.START,RO
(3) 011322 104050      EMT      CSREFG
3335 011324 103007      BNCOMPLET CONTINUE
(2) 011324 103007      BCC      CONTINUE
3336 011326 013737 002014 002114  START: MOV      LSUNIT,UUT
3337 011334 012737 177777 002116  MOV      #-1,UNITST
3338 011342 000404      BR       NXT
3339
3340 011344 012700 000036      CONTINUE: REDEF    #EF.CONTINUE ;CONTINUE????
(3) 011344 012700 000036  MOV      #EF.CONTINUE,RO
(3) 011350 104050      EMT      CSREFG
3341 011352 103443      BCOMPLET CONT
(2) 011352 103443      BCS      CONT
3342
3343 011354 005737 002114      NXT:    TST      UUT      ;DONE ALL UUT'S
3344 011360 001006      BNE     IS            ;NO
3345 011362 012737 177777 002116  MOV      #-1,UNITST
3346 011370 013737 002014 002114  MOV      LSUNIT,UUT
3347
3348 011376 005237 002116      IS:    INC      UNITST
3349 011402 005337 002114      DEC     UUT
3350 011406 013700 002116      REST:  GPHARD  UNITST,RO
(3) 011406 013700 002116  MOV      UNITST,RO
(3) 011412 104042      EMT      CSGPHRD
3351 011414 103406      BCOMPLET IS
(2) 011414 103406      BCS     IS
3352 011416 005737 002112      TST     PWRFLG      ;POWER FLAG TO 0
3353 011422 001754      BEQ     NXT        ;YES, DONT DEC IT
3354 011424 005337 002112      DEC     PWRFLG
3355 011430 000751      BR      NXT        ;GET NEXT ONE
3356 011432 012037 002130      IS:    MOV      (RO)+,BCSR
3357 011436 012037 002134      MOV     (RO)+,BVEC
3358 011442 012037 002132      MOV     (RO)+,BPRIOR
3359 011446 012037 002136      MOV     (RO)+,DRIVE
3360 011452 012037 002252      MOV     (RO)+,T.CNTRL ;GET CONTROLLER TYPE
3361
3362 011456 005037 002246      CLR     ERRLMT      ;INIT ERROR COUNT

```

```

3363 011462 013700 002130      CONT:  MOV      BCSR,RO
3364 011466 010037 002120      MOV      RO,RLCS
3365 011472 062700 000002      ADD      #2,RO
3366 011476 010037 002122      MOV      RO,RLBA
3367 011502 062700 000002      ADD      #2,RO
3368 011506 010037 002124      MOV      RO,RLDA
3369 011512 062700 000002      ADD      #2,RO
3370 011516 010037 002126      MOV      RO,RLMP
3371 011522 005737 002112      TST      PWRFLG
3372 011526 001064          BNE      5$
3373 011530 005737 011120      TST      T.SIZE          ;DO WE WANT TO CHECK UNITS??
3374 011534 001461          BEQ      5$              ;NO
3375
3376 011536 005037 002170      CLR      TRPFLG          ;CLR OUT TRAP FLAG
3377 011542          SETVEC  ERRVEC,#TRPHAN,#340 ;SETJP VECTOR TO CATCH NON-EXIST
(7) 011542 012746 000340      MOV      #340,-(SP)
(6) 011546 012746 013302      MOV      #TRPHAN,-(SP)
(5) 011552 013746 002202      MOV      ERRVEC,-(SP)
(4) 011556 012746 000003      MOV      #3,-(SP)
(3) 011562 104037          EMT      C$SVEC
(2) 011564 062706 003010      ADD      #10,SP
3378 011570 005777 170324      TST      @RLCS          ;ACCESS CONTROLLER
3379 011574          CLRVEC  ERRVEC          ;RELEASE VECTOR
(3) 011574 013700 002202      MOV      ERRVEC,RO
(3) 011600 104036          EMT      C$CVEC
3380 011602 005737 002170      TST      TRPFLG          ;DID IT TRAP
3381 011606 001404          BEQ      7$              ;NO, CHECK IT'S DRIVE
3382 011610 012737 003370 002250      MOV      #NORES,WHY     ;SETUP ERR MESS
3383 011616 000415          BR       8$
3384
3385 011620 012777 000200 170272 7$:  MOV      #200,@RLCS     ;CONTROLLER READY
3386 011626 053777 002136 170264      BIS      DRIVE,@RLCS    ;SELECT DRIVE
3387 011634 032777 000001 170256      BIT      #1,@RLCS      ;DRIVE THERE
3388 011642 001016          BNE      5$              ;YES
3389 011644 012737 003406 002250      MOV      #NODRY,WHY     ;SETUP ERR MESS
3390 011652          PRINTB #FRMT13,WHY
(8) 011652 013746 002250 8$:  MOV      WHY,-(SP)
(7) 011656 012746 011044      MOV      #FRMT13,-(SP)
(6) 011662 012746 000002      MOV      #2,-(SP)
(3) 011666 010600      MOV      SP,RO
(4) 011670 104014      EMT      C$PNTB
(4) 011672 062706 000006      ADD      #6,SP
3391 011676 000434          BR       6$
3392
3393 011700 005737 002112 5$:  TST      PWRFLG          ;RECENT POWER FAILURE??
3394 011704 001457          BEQ      END            ;NO
3395
3396          ;THERE WAS A RECENT POWER FAILURE, THEREFORE WE WILL WAIT
3397          ;SIXTY SECONDS FOR THE DRIVE TO COME READY
3398
3399 011706 012701 000074          MOV      #60,R1         ;SIXTY SECOND TIMEOUT
3400 011712 012777 000200 170200      MOV      #200,@RLCS    ;SET CRDY
3401 011720 053777 002136 170172      BIS      DRIVE,@RLCS   ;SET IN DRIVE SELECT
3402 011726 032777 000001 170164 2$:  BIT      #DRDY,@RLCS   ;DRIVE READY??
3403 011734 001023          BNE      3$              ;YES, THEN START TEST
3404

```



```

3405 011736          WAITMS #10.          ;WAIT A SECOND
(3) 011736 012700 000012  MOV #10.,RO
(3) 011742 104026          EMT CSWTM
3406 011744 005301          DEC R1          ;SIXTY SECONDS GONE BY
3408 011746 001367          BNE 2$         ;NO, GO BACK
3409
3410 011750          PRINTB #FRMT12          ;DROPPING DRIVE
(7) 011750 012746 010765  MOV #FRMT12, -(SP)
(6) 011754 012746 000001  MOV #1, -(SP)
(3) 011760 010600          MOV SP,RO
(4) 011762 104014          EMT C$PNTB
(4) 011764 062706 000004  ADD #4,SP
3411 011770 004737 010102 6$: JSR PC,LINE1  ;GIVE DRIVE INFO
3412 011774          DODU UNITST    ;TELL SUPERVISOR TO DROP IT
(3) 011774 013700 002116  MOV UNITST,RO
(3) 012000 104053          EMT C$DODU
3413 012002          DOCLN          ;FORCE AN ABORT
(3) 012002 104044          EMT C$DCLN
3414
3415
3416 012004 012777 000013 170112 3$: MOV #13,DRLOA  ;SETUP DR RST
3417 012012 012777 000204 170100  MOV #204,DRLCS ;GS FUNC
3418 012020 053777 002136 170072  BIS DRIVE,DRLCS ;SELECT DRIVE
3419 012026 042777 000200 170064  BIC #200,DRLCS  ;ISSUE IT
3420 012034 032777 000200 170056 4$: BIT #200,DRLCS ;WAIT FOR READY
3421 012042 001774          BEQ 4$
3422
3423 END: SETVEC BVEC,#INTSRV,#340
(7) 012044 012746 000340  MOV #340, -(SP)
(6) 012050 012746 013310  MOV #INTSRV, -(SP)
(5) 012054 013746 002134  MOV BVEC, -(SP)
(4) 012060 012746 000003  MOV #3, -(SP)
(3) 012064 104037          EMT C$SVEC
(2) 012066 062706 000010  ADD #10,SP
3424 012072 005037 002166  CLR PFLG      ;CLR PROCESSOR FLAG
3425 012076          READBUS      ;Q-BUS
(3) 012076 104007          EMT C$RDBU
3426 012100          BNCOMPLETE 1$
(2) 012100 103002          BCC 1$
3427 012102 005237 002166 1$: INC PFLG      ;NO, Q-BUS THEN
3428 012106          ENDINIT
(3) 012106 104011  L10012: EMT C$INIT
3429
3430          ENDMOD
3431
3432          BGNMOD CLNCODE
3433          BGNCLN
3434
3435 012110          SETPRI #PRI07
(3) 012110 012700 000340  MOV #PRI07,RO
(3) 012114 104041          EMT C$SPRI
3436
3437
3438

```

012116	032777	000200	167774	1\$:	BIT	#CRDY, @RLCS	
012124	001774				BEQ	1\$	
012126	042777	000100	167764		BIC	#INTEN, @RLCS	
012134					CLRVEC	BVEC	
012134	013700	002134			MOV	BVEC, R0	
012140	104036				EMT	C\$CVEC	
012142	005737	002112			TST	PWRFLG	;TREAT POWER FAILURE
012146	001402				BEQ	2\$	
012150	005337	002112			DEC	PWRFLG	
012154				2\$:			
012154				L10013:	ENDCLN		
012154	104012				EMT	C\$CLEAN	
012156					ENDMOD		
012156				BGNMOD	DRPCODE		
012156					BGNDU		
012156	000240				NOP		
012160					ENDDU		
012160				L10014:			
012160	104055				EMT	C\$DU	
012162					ENDMOD		
012162				BGNMOD	ADDCODE		
012162					BGNAU		
012162	000240				NOP		
012164					ENDAU		
012164				L10015:			
012164	104054				EMT	C\$AU	
012166					ENDMOD		
				.SBTTL	GLOBAL SUBROUTINES		
012166				BGNMOD	GLBSUB		
012166				CKERLT:	INLOOP	*	
012166	104020				EMT	C\$INLP	

```

3486 012170
(2) 012170 103427
3487 012172 005737 011114
3488 012176 001424
3489 012200 005237 002246
3490 012204 023737 002246 011116
3491 012212 002416
3492
3493 012214
(1) 012214 012746 010724
(6) 012220 012746 000001
(3) 012224 010600
(4) 012226 104017
(4) 012230 062706 000004
3494 012234 004737 010102
3495 012240
(3) 012240 013700 002116
(3) 012244 104053
(4) 012246
(3) 012246 104044
3497 012250
3498 012250 000205
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000

```

```

BCOMPLETE 99$
BCS 99$
TST DROP
BEQ 99$
INC ERRLMT
CMP ERRLMT, MERLMT
BLT 99$

PRINTF #FRMT11
MOV #FRMT11, -(SP)
MOV #1, -(SP)
MOV SP, RO
EMT C$PNTF
ADD #4, SP
JSR PC, LINE1
DODU UNITST ;DROP THE UNIT
MOV UNITST, RO
EMT C$DODU
DOCLN
EMT C$DCLN

99$:
RTS R5

```

.SBTTL ROUTINE TO CHECK FOR CONTROLLER ERRORS

```

*****
*THIS ROUTINE WILL CHECK RLCS FOR ERRORS AND PRINT THEM
*ACCORDINGLY. IT WILL MERGE THE ERROR PRINTOUT WITH THE TEST
*ERROR MESSAGE.
*
*EXAMPLE: RLCS CONTAINED FOLLOWING ERROR(S):
*          DRV OPI HCRC HNF
*          SEEK UNDER INTERRUPT
*
*
*ROUTINE USES RO, R1 AND PICKS HEADER FROM R3
*
*          CALL JSR R5, CHERR
*
*

```

```

012252 005037 002150
012256 032737 176000 002152
012264 001001
012266 000205
012270 023727 002254 000004
012276 002401
012300 000414
012302 023727 002254 000002
012310 001410
012312 013700 002152
012316 042700 001777

```

```

CHERR: CLR DERFLG ;CLEAR OUT DRIVE ERROR FLAG
BIT #176000, E.CS ;ANY ERRORS SET
BNE 199$ ;IF YES, INVESTIGATE
RTS R5 ;NO, EXIT
199$: CMP TMPFNC, #GSTAT ;FUNCTION-NOP, RESET, GETSTATUS
BLT 98$ ;YES, GO CHECK IF ONLY DRIVE ERROR
BR 1$ ;YES SERVICE ERROR
98$: CMP TMPFNC, #WRCHK
IS
BEQ 1$
MOV E.CS, RO ;GET E.CS
BIC #1777, RO

```

```

3533 012322 022700 140000 CMP #140000,R0 ;DRIVE ERROR ALONE?
3534 012326 001001 BNE 1$ ;NO, GO SERVICE
3535 012330 000205 2$: RTS R5 ;YES, EXIT
3536
3537 012332 012701 007365 1$: MOV #EM102,R1 ;GET START OF STRING
3538 012336 005737 002152 TST E.CS ;IS COMPOSITE ERROR SET? (BETTER BE
3539 012342 100003 BPL 99$ ;IT'S NOT SOMETHING IS WRONG
3540 012344 004537 013016 JSR R5, FIX ;YES, PUT "COMP" IN STRING
3541 012350 003502 COMP ;"COMP"
3542 012352 032737 040000 002152 99$: BIT #DERR,E.CS ;DRIVE ERROR SET?
3543 012360 001405 BEQ 3$ ;NO, CONTINUE
3544 012362 005237 002150 INC DERFLG ;SET DRV ERROR FLAG
3545 012366 004537 013016 JSR R5, FIX ;YES, PUT "DRV" INTO STRING
3546 012372 003431 DEMES ;"DRV"
3547 012374 032737 020000 002152 3$: BIT #NXM,E.CS ;NON-EXISTENT MEMORY ERROR?
3548 012402 001403 BEQ 4$ ;NO, CONTINUE
3549 012404 004537 013016 JSR R5, FIX ;YES, PUT "NXM" INTO STRING
3550 012410 003436 NXMMES ;"NXM"
3551 012412 032737 002000 002152 4$: BIT #OPI,E.CS ;IS OPI SET?
3552 012420 001422 BEQ 6$ ;NO, GO CHECK BITS 11 & 12
3553 012422 004537 013016 JSR R5, FIX ;PUT "OPI" INTO STRING
3554 012426 003443 OPIMES ;"OPI"
3555 012430 032737 004000 002152 BIT #BIT11,E.CS ;HEADER CRC ERROR?
3556 012436 001403 BEQ 5$ ;NO, GO CHECK HEADER NOT FOUND
3557 012440 004537 013016 JSR R5, FIX ;GO PUT "HCRC" IN STRING
3558 012444 003450 HCRCMES ;"HCRC"
3559 012446 032737 010000 002152 5$: BIT #BIT12,E.CS ;HEADER NOT FOUND?
3560 012454 001422 BEQ 8$ ;NO, GO PUT "CRLF" IN STRING
3561 012456 004537 013016 JSR R5, FIX ;PUT "HNF" IN STRING
3562 012462 003456 HNFMES ;"HNF"
3563 012464 000416 BR 8$ ;PUT "CRLF" IN STRING
3564 012466 032737 004000 002152 6$: BIT #BIT11,E.CS ;DATA CRC ERROR?
3565 012474 001403 BEQ 7$ ;NO, GO CHECK DATA LATE
3566 012476 004537 013016 JSR R5, FIX ;PUT "DCK" IN STRING
3567 012502 003463 DCKMES ;"DCK"
3568 012504 032737 010000 002152 7$: BIT #BIT12,E.CS ;DATA LATE ERROR?
3569 012512 001403 BEQ 8$ ;NO, GO PUT IN "CRLF"
3570 012514 004537 013016 JSR R5, FIX ;PUT "DLT" IN STRING
3571 012520 003470 DLTMES ;"DLT"
3572 012522 004537 013016 8$: JSR R5, FIX
3573 012526 003475 MSCRLF
3574 012530 004537 013016 JSR R5, FIX
3575 012534 000000 RESTMS: .WORD 0 ;HEADER FROM TEST
3576 012536 105011 CLAB (R1) ;PUT TERMINATOR IN
3577
3578 012540 ERROF 300, LF, ERR6
3579 ( ) 012540 104462 TRAP T$ERRCODE
3580 ( ) 012542 000454 .WORD 300
3581 ( ) 012544 003500 .WORD LF
3582 ( ) 012546 010002 .WORD ERR6
3583
3584 012550 000205 RTS R5 ;EXIT ROUTINE
3585
3586 .SBTTL LOAD RLCS
3587 *****
3588 ;* ROUTINE TO LOAD RLCS WITH FUNCTION TO BE PERFORMED

```

```

3585          : *      CALL:   JSR      RS,LDFUNC
3586          : *      .WORD          ;BITS TO BE LOADED, FUNCTION
3587          : *      ;AND INTR ENABLE ONLY
3588          : *
3589          : *
3590          : *
3591 012552 012537 002174 LDFUNC: MOV      (R5)+,LDCSR      ;GET BITS TO LOAD
3592 012556 005737 002150      TST      DERFLG
3593 012562 001424          BEQ      98$
3594 012564 013746 002140      MOV      B,CS, -(SP)
3595 012570 012777 000013 167326      MOV      #13,ARLOA
3596 012576 012737 000004 002140      MOV      #GSTAT,B,CS
3597 012604 053737 002136 002140      BIS      DRIVE,B,CS
3598 012612 013777 002140 167300      MOV      B,CS,ARLCS
3599 012620 012637 002140          MOV      (SP)+,B,CS
3600 012624 032777 000200 167266 99$: BIT      #200,ARLCS
3601 012632 001774          BEQ      99$
3602 012634 010346 98$: MOV      R3, -(SP)      ;SAVE R3
3603 012636 042737 177661 002174      BIC      #177661,LDCSR      ;CLEAR ALL BUT FUNC & INTR EN
3604 012644 013737 002174 012770      MOV      LDCSR,FNDFNC      ;SAVE FUNCTION
3605 012652 042737 000100 012770      BIC      #INTEN,FNDFNC      ;ONLY FUNCTION
3606 012660 013737 012770 002254      MOV      FNDFNC,TMPFNC
3607 012666 012703 012772          MOV      #HDRLST,R3      ;GET HEADER LIST
3608 012672 006237 012770          ASR      FNDFNC      ;ALIGN TO RIGHT
3609 012676 001404          BEQ      2$
3610 012700 022323 1$: CMP      (R3)+,(R3)+      ;BUMP R3 BY 4
3611 012702 005337 012770          DEC      FNDFNC      ;FOUND IT
3612 012706 001374          BNE      1$      ;NO KEEP LOOKING
3613 012710 032737 000100 002174 2$: BIT      #INTEN,LDCSR      ;YES DO WE WANT FLAG OR INTR
3614 012716 001401          BEQ      3$      ;FLAG BRANCH
3615 012720 005723          TST      (R3)+      ;INTR POINT TO THAT ONE
3616 012722 011303 3$: MOV      (R3),R3      ;SET HEADER
3617 012724 010337 012534          MOV      R3,RESTMS      ;SET UP HEADER
3618 012730 053737 002136 002174      BIS      DRIVE,LDCSR      ;SELECT DRIVE
3619 012736 052737 000200 002174 4$: BIS      #200,LDCSR      ;CONTROLLER READY
3620 012744 013777 002174 167146      MOV      LDCSR,ARLCS
3621 012752 004537 013030          JSR      R5,BEFORE
3622 012756 042777 000200 167134 5$: BIC      #200,ARLCS
3623 012764 012603          MOV      (SP)+,R3      ;RESTORE R3
3624 012766 000205          RTS      R5      ;EXIT
3625
3626 012770 000000 FNDFNC: .WORD 0
3627
3628 012772 003563 HDRLST: NOPMES
3629 012774 003614      NOPINT
3630 012776 003646      WCKMES
3631 013000 003706      WCKINT
3632 013002 004133 OKHDR: GSTMES
3633 013004 004172      GSTINT
3634 013006 004050      SEKMES
3635 013010 004101      SEKINT
3636 013012 003747      RHMES
3637 013014 004007      RHDINT
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000

```

```

:*****
:ROUTINE TO MOVE ASCII STRINGS

```

```

3641          : *USES REGISTERS R1 - WHERE STRING IS BEING BUILT
3642          : *
3643          : *      CALL      JSR      R5, FIX
3644          : *      .WORD      .          ; ADDRESS OF STRING TO MOVE
3645
3646 013016 012500  FIX:  MOV      (R5)+, R0      ; GET ADDRESS AND MOVE RETURN
3647 013020 112021  IS:  MOVB   (R0)+, (R1)+    ; GET BYTE AND UPDATE
3648 013022 001376      BNE     IS          ; WATCH 0 BYTE TERMINATOR
3649 013024 105741      TSTB   -(R1)       ; BACK UP OVER ZERO BYTE
3650 013026 000205      RTS      R5          ; EXIT
3651
3652
3653          : LOAD REGISTERS BEFORE FUNCTION
3654          : CALL:  JSR      R5, BEFORE
3655
3656 013030 017737 167064 002140  BEFORE: MOV      @RLCS, B. CS      ; READ CS
3657 013036 017737 167060 002142      MOV      @RLBA, B. BA      ; READ BA
3658 013044 017737 167054 002144      MOV      @RLDA, B. DA      ; READ DA
3659 013052 017737 167050 002146      MOV      @RLMP, B. MP      ; READ MP
3660 013060 000205      RTS      R5
3661
3662
3663          : LOAD REGISTERS AT ERROR
3664          : CALL:  JSR      R5, AFTER
3665
3666 013062 017737 167032 002152  AFTER:  MOV      @RLCS, E. CS      ; READ CS
3667 013070 017737 167026 002154      MOV      @RLBA, E. BA      ; READ BA
3668 013076 017737 167022 002156      MOV      @RLDA, E. DA      ; READ DA
3669 013104 017737 167016 002160      MOV      @RLMP, E. MP      ; READ MP
3670 013112 017737 167010 002162      MOV      @RLMP, E. MP1     ; READ MP
3671 013120 017737 167002 002164      MOV      @RLMP, E. MP2     ; READ MP
3672 013126 000205      RTS      R5
3673
3674
3675
3676          .SBTTL  ROUTINE TO CALCULATE CRC
3677
3678          : ROUTINE WILL CALCULATE A CRC-16 CRC ON A WORD OF
3679          : 1-16 BITS IN LENGTH. RESULT IS RETURNED IN "CALBCC"
3680
3681          :
3682          :      CALL:  JSR      R5, SIMBCC
3683          :      .WORD      .          ; NUMBER OF BITS (1-16)
3684          :      .WORD      .          ; DATA FOR CRC CALCULATION
3685          :      .WORD      .          ; PREVIOUS OR STARTING CRC
3686          :      .          ; (SHOULD BE ZEROED FOR START)
3687
3688          :
3689          :      ROUTINE USES R0, R1, R2
3690
3691 013130 010046  SIMBCC: MOV      R0, -(SP)      ; SAVE R0
3692 013132 010146      MOV      R1, -(SP)      ; SAVE R1
3693 013134 010246      MOV      R2, -(SP)      ; SAVE R2
3694 013136 012537 002210      MOV      (R5)+, TEMP2    ; GET NUMBER OF BITS
3695 013142 012537 002212      MOV      (R5)+, TEMP3    ; GET DATA FOR CRC CALCULATION
3696 013146 012537 002214      MOV      (R5)+, TEMP4    ; GET STARTING CRC
3697 013152 005037 002204  IS:  CLR      BCCFBK        ;
3698 013156 013700 002214      MOV      TF1P4, R0      ; GET PRESENT CRC
3699 013162 006037 002212      ROR      TEMP3         ; ROTATE NEW DATA
    
```

```

3697 013166 005500 ADC RO ;MERGE NEW WITH OLD
3699 013170 032700 000001 BIT #1,RO ;BIT 0 SET
3699 013174 001402 BEQ 2$ ;IF NOT CONTINUE
3700 013176 005137 002204 COM BCCFBK ;
3701 013202 013700 002200 2$: MOV XPOLY,RO ;GET CRC POLYNOMIAL (CRC-16.)
3702 013206 005100 COM RO ;COMPLIMENT POLYNOMIAL
3703 013210 040037 002204 BIC RO,BCCFBK ;
3704 013214 000241 CLC ;CLEAR CARRY
3705 013216 006037 002214 ROR TEMP4
3706 013222 013700 002204 MOV BCCFBK,RO
3707 013226 013701 002214 MOV TEMP4,R1
3708 013232 010102 MOV R1,R2
3709 013234 040100 BIC R1,RO
3710 013236 043702 002204 BIC BCCFBK,R2
3711 013242 050200 BIS R2,RO
3712 013244 043737 002200 002214 BIC POLY,TEMP4
3713 013252 050037 002214 BIS RO,TEMP4
3714 013256 005337 002210 DEC TEMP2
3715 013262 001333 BNE 1$
3716 013264 013737 002214 002206 MOV TEMP4,CALBCC
3717 013272 012602 MOV (SP)+,R2
3718 013274 012601 MOV (SP)+,R1
3719 013276 012600 MOV (SP)+,RO
3720 013300 000205 RTS R5 ;RETURN

```

```

;ROUTINE TO SET FLAG IF TRAP OCCURRED
;"TRPHAN" IS IN LOCATION 4.

```

```

3721 013302 005237 002170 TRPHAN: INC TRPFLG ;INDICATE TRAP
3722 013306 000002 RTI ;RETURN
3723 013310 BGNSRV
3724 013310 005237 002172 INTSRV: INC INTFLG ;INDICATE INTERRUPT
3725 013314 ENDSRV
3726 013314 L10016: RTI

```

```

;ROUTINE TO WAIT FOR DRIVE READY

```

```

3727 013316 010146 WTDROY: MOV R1, -(SP) ;SAVE R1
3728 013320 012701 003720 MOV #200,R1 ;TIME OUT OF 200 MILLISECONDS
3729 013324 032777 000001 166566 1$: BIT #DROY,DRCS ;DRIVE READY?
3730 013332 001011 BNE 2$ ;YES, EXIT
3731 013334 WAITUS #1 ;WAIT A WHILE
3732 013334 012700 000001 MOV #1,RO
3733 013340 04027 EMT CSWTU
3734 013342 005301 DEC R1 ;CHECK IF TIME UP
3735 013344 001367 BNE 1$ ;NO, GO CHECK DRIVE READY
3736 013346 ERADF 200,DRIM,ERRS ;DRIVE READY DID NOT SET
3737 013346 04462 TRAP T$ERRCODE

```

```

(5) 013350 000310 .WORD 200
(5) 013352 004347 .WORD DRTIM
(5) 013354 007770 .WORD ERR5
(5) 013356 012601 2$: MOV (SP)+,R1 ;RESTORE
(5) 013360 000205 RTS R5 ;EXIT

:ROUTINE TO WAIT FOR CONTROLLER READY
WTCRDY: MOV R1, -(SP) ;SAVE R1
MOV #800, R1 ;WAIT 800 MILLISECONDS
1$: BIT #CRDY, RLC5 ;CONTROLLER READY
BNE 2$ ;YES EXIT
WAITUS #1 ;WAIT A WHILE
MOV #1, R0
EMT C$WTU
DEC R1 ;CHECK IF TIME UP
BNE 1$ ;NO GO BACK

013412 004537 013062 JSR R5, AFTER ;GET REGISTERS

013416 010146 ERRDF 100, CRTIM, ERR6 ;CONTROLLER TIMED OUT
013416 104462 TRAP T$ERRCODE
013420 000144 .WORD 100
013422 004322 .WORD CRTIM
013424 010002 .WORD ERR6

013426 000402 BR 3$ ;EXIT

013430 004537 013062 2$: JSR R5, AFTER ;GET REGISTERS
013434 012601 3$: MOV (SP)+, R1
013436 000205 RTS R5 ;EXIT

013440 ENDMOD

.SBTTL **TEST 1** - RLCS ADDRESSABILITY
BGNTST ;****START OF TEST****
STARS
:*****
:TEST TO SEE IF WE CAN ADDRESS THE CONTROL
:AND STATUS REGISTER. IF WE TRAP WE WILL REPORT
:THE ERROR AND ABORT. AFTER THIS TEST WE ONLY KNOW
:THAT WE CAN ADDRESS THE REGISTER.
STARS
:*****

013440 005037 002170 1$: CLR TRPFLG ;CLEAR TRAP OCCURANCE
013444 2$: SETVEC ERRVEC, #TRPHAN, #340 ;SET TO CATCH TRAP
013444 MOV #340, -(SP)
013446 012746 000340 MOV #TRPHAN, -(SP)
013450 012746 013302 MOV ERRVEC, -(SP)
013454 013746 002202

```



```

(4) 013460 012746 000003      MOV      #3, -(SP)
(3) 013464 104037      EMT      C$SVEC
013466 062706 000010      ADD      #10, SP
3790 013472 005777 166422      TST      @RLCS          ;ADDRESS RLCS
3791 013476      CLRVEC  ERRVEC          ;RELEASE TRAP VECTOR
(3) 013476 013700 002202      MOV      ERRVEC, R0
013502 104036      EMT      C$CVEC
3793 013504 005737 002170      TST      TRPFLG          ;TRAP OCCURRED???
3794 013510 001407      BEQ      3$              ;NO, IKAY PROCEED
3795 013512 013737 002120 002224      MOV      RLCS, GDDAT      ;SET UP ERROR DATA
3796 013520      ERRSF  0, EM1, ERR1      ;BUS TIMEOUT IN ADDRESSING RLCS
(3) 013520 104461      TRAP    T$ERCODE
(5) 013522 000000      .WORD  0
(5) 013524 004375      .WORD  EM1
(5) 013526 007574      .WORD  ERR1
3798 013530      3$:   CKLOOP              ;CHECK IF /FL:LOE IS SET
(3) 013530 104006      EMT      C$CLP1          ;****END OF TEST****
3799 013532      ENDTST
(3) 013532 104001      L10017: EMT      C$SETST
3800
3801
3802
3803
3804 013534      .SBTTL  **TEST 2** - PLBA ADDRESSABILITY
3805      BGNTST                    ;****START OF TEST****
3806
3807 013534      STARS
(2) 3808      ;*****
3809      ;TEST TO SEE IF WE CAN ADDRESS THE BUS ADDRESS
3810      ;REGISTER. IF WE TRAP WE WILL REPORT THE ERROR
3811      ;AND ABORT. AFTER THIS TEST WE ONLY KNOW THAT
3812 013534      ;WE CAN ADDRESS THE REGISTER.
3813      ;*****
3814 013534 005037 002170      1$:   CLR      TRPFLG          ;CLEAR TRAP OCCURANCE
3815 013540 012746 000340      2$:   SETVEC  ERRVEC, #TRPHAN, #340 .SET TO CATCH TRAP
(7) 013540 012746 013302      MOV      #340, -(SP)
(6) 013544 012746 013302      MOV      #TRPHAN, -(SP)
(5) 013550 013746 002202      MOV      ERRVEC, -(SP)
(4) 013554 012746 000003      MOV      #3, -(SP)
(3) 013560 104037      EMT      C$SVEC
(2) 013562 062706 000010      ADD      #10, SP
3816 013566 005777 166330      TST      @RLBA          ;ADDRESS RLBA
3817 013572      CLRVEC  ERRVEC          ;RELEASE TRAP VECTOR
(3) 013572 013700 002202      MOV      ERRVEC, R0
(3) 013576 104036      EMT      C$CVEC
3819 013600 005737 002170      TST      TRPFLG          ;TRAP OCCURRED???
3820 013604 001407      BEQ      3$              ;NO, CONTINUE
3821 013606 013737 002122 002224      MOV      RLBA, GDDAT      ;SETUP ERROR DATA
3822 013614      ERRSF  1, EM2, ERR1      ;BUS TIMEOUT IN ADDRESSING R. BA

```

(3) 013614 104461  
(5) 013616 000001  
(5) 013620 004422  
(5) 013622 007574  
3824 013624 104006  
(3) 013624 104006  
3825 013626  
(3) 013626  
(3) 013626 104001  
3826  
3827  
3828  
3829  
3830 013630  
3831 013630  
(3)  
3832  
3833  
3834  
3835  
3836 013630  
(2)  
3837  
3838  
3839 013630 005037 002170  
3840 013634  
(7) 013634 012746 000340  
(6) 013640 012746 013302  
(5) 013644 013746 002202  
(4) 013650 012746 000003  
(3) 013654 104037  
(2) 013656 062706 000010  
3841  
3842 013662 005777 166236  
3843 013666  
(3) 013666 013700 002202  
(3) 013672 104036  
3844 013674 005737 002170  
3845 013700 001407  
3846  
3847 013702 013737 002124 002224  
3848 013710  
(3) 013710 104461  
(5) 013712 000002  
(5) 013714 004447  
(5) 013716 007574  
3849 013720  
(3) 013720 104006  
3850 013722  
(3) 013722  
(3) 013722 104001  
3851  
3852  
3853  
3854  
3855 013724

TRAP T\$ERCODE  
.WORD 1  
.WORD EM2  
.WORD ERR1  
3\$: CKLOOP ;CHECK IF /FL:LOE IS SET  
EMT C\$CLP1  
ENDTST ;\*\*\*\*END OF TEST\*\*\*\*  
L10020: EMT C\$ETST

.SBTTL \*\*TEST 3\*\* - RLDA ADDRESSABILITY

BGNTST ;\*\*\*\*START OF TEST\*\*\*\*  
STARS  
:\*\*\*\*\*  
:TEST TO SEE IF WE CAN ADDRESS THE DISK ADDRESS  
:REGISTER IF WE TRAP WE WILL REPORT THE ERROR  
:AND ABORT. AFTER THIS TEST WE ONLY KNOW THAT  
:WE CAN ADDRESS THE REGISTER.  
STARS  
:\*\*\*\*\*

1\$: CLR TRPFLG ;CLEAR TRAP OCCURANCE  
2\$: SETVEC ERRVEC, #TRPHAN, #340 ;SET TO CATCH TRAP  
MOV #340, -(SP)  
MOV #TRPHAN, -(SP)  
MOV ERRVEC, -(SP)  
MOV #3, -(SP)  
EMT C\$SVEC  
ADD #10, SP  
TST @RLDA ;ADDRESS RLDA  
CLRVEC ERRVEC ;RELEASE TRAP VECTOR  
MOV ERRVEC, R0  
EMT C\$CVEC  
TST TRPFLG ;TRAP OCCURRED???  
BEQ 3\$ ;NO, CONTINUE  
MOV RLDA, GDDAT ;SETUP ERROR INFO  
ERRSF 2, EM3, ERR1 ;BUS TIMEOUT IN ADDRESSING RLDA  
TRAP T\$ERCODE  
.WORD 2  
.WORD EM3  
.WORD ERR1  
3\$: CKLOOP ;CHECK IF /FL:LOE IS SET  
EMT C\$CLP1  
ENDTST ;\*\*\*\*END OF TEST\*\*\*\*  
L10021: EMT C\$ETST

.SBTTL \*\*TEST 4\*\* - RLMP ADDRESSABILITY

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

3856 013724  
 (2)  
 3857  
 3858  
 3859  
 3860  
 3861 013724  
 (2)  
 3862  
 3863  
 3864 013724 005037 002170  
 3865 013730  
 (7) 013730 012746 000340  
 (6) 013734 012746 013302  
 (5) 013740 013746 002202  
 (4) 013744 012746 000003  
 (3) 013750 104037  
 (2) 013752 062706 000010  
 3866  
 3867 013756 005777 166144  
 3868 013762  
 (3) 013762 013700 002202  
 (3) 013766 104036  
 3869 013770 005737 002170  
 3870 013774 001407  
 3871 013776 013737 002126 002224  
 3872  
 3873 014004  
 (3) 014004 104461  
 (5) 014006 000003  
 (5) 014010 004474  
 (5) 014012 007574  
 3874 014014  
 (3) 014014 104006  
 3875 014016  
 (3) 014016  
 (3) 014016 104001  
 3876  
 3877  
 3878  
 3879  
 3880 014020  
 3881  
 3882  
 3883  
 3884 014020  
 (2)  
 3885  
 3886  
 3887  
 3888  
 3889 014020  
 (2)  
 3890  
 3891  
 3892 014020 012703 002570

```

STARS
:*****
:TEST TO SEE IF WE CAN ADDRESS THE MULTIPURPOSE
:REGISTER. IF WE TRAP WE WILL REPORT THE ERROR AND
:ABORT. AFTER THIS TEST WE ONLY KNOW THAT WE CAN
:ADDRESS THE REGISTER.
STARS
:*****

1$: CLR TRPFLG ;CLEAR TRAP OCCURANCE
2$: SETVEC ERRVEC, #TRPHAN, #340 ;SET UP TO CATCH TRAP
MOV #340, -(SP)
MOV #TRPHAN, -(SP)
MOV ERRVEC, -(SP)
MOV #3, -(SP)
EMT C$SVEC
ADD #10, SP

TST @RLMP ;ADDRESS RLMP
CLRVEC ERRVEC ;RELEASE TRAP VECTOR
MOV ERRVEC, R0
EMT C$CVEC
TST TRPFLG ;TRAP OCCURRED???
BEQ 3$ ;NO, CONTINUE
MOV RLMP, GDDAT ;SET UP ERROR INFO

ERRSF 3, EM4, ERR1 ;BUS TIMEOUT IN ADDRESSING RLMP
TRAP TS$RCODE
.WORD 3
.WORD EM4
.WORD ERR1

3$: CKLOOP ;CHECK IF /FL:LOE IS SET
EMT C$CLP1

ENDTST ;****END OF TEST****
L10022: EMT C$SETST

.SBTTL **TEST 5** - READ WRITE OF RLCS
BGNTST ;****START OF TEST****

STARS
:*****
:TEST THAT WE CAN WRITE/READ BITS 8,9 AND BITS 6-1
:OF THE CONTROL AND STATUS REGISTER. BITS 15-10 AND 0
:ARE DON'T CARE BITS AT THIS TIME AND BIT 7
:((CONTROLLER READY) IS ALWAYS WRITTEN TO A ONE.
STARS
:*****

MOV #C$PAT, R3 ;SET UP TABLE POINTER OF PATTERNS

```

```

3893
3894 014024 BGNSEG :****START OF SEGMENT****
(3) 014024 104004 EMT CSBSEG
3895
3896 014026 CSTEST:
3897 014026 011337 002224 MOV (R3),GDDAT ;GET PATTERN INTO GDDAT
3898 014032 052737 000200 002224 BIS #200,GDDAT ;INSURE GO IS SET
3899 014040 013777 002224 166052 MOV GDDAT,RRLCS ;LOAD RLCS (CONTROL AND STATUS)
3900 014046 032777 040000 166044 BIT #DERR,RRLCS ;IF DRIVE ERROR PRESENT
3901 014054 001403 BEQ 99$ ;THEN EXPECT DRIVE AND
3902 014056 052737 140000 002224 BIS #ERR!DERR,GDDAT ;COMPOSITE ERROR
3903 014064 017737 166030 002226 99$: MOV RRLCS,BDDAT ;READ RLCS BACK
3904 014072 042737 000001 002226 BIC #DRDY,BDDAT ;IGNORE DRIVE READY
3905 014100 023737 002224 002226 CMP GDDAT,BDDAT ;DID WE READ WHAT WE LOADED
3906 014106 001404 BEQ 1$ ;YES, THEN BRANCH
3907
3908 014110 ERROF 4. EMS ERR2 ;WRONG DATA IN RLCS
(3) 014110 104462 TRAP T$ERRCODE
(5) 014112 000004 .WORD 4
(5) 014114 004521 .WORD EMS
(5) 014116 007606 .WORD E-2
3909 014120 1$: ESCAPE SEG ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
(3) 014120 104010 EMT C$ESCAPE
(3) 014122 000012 .WORD 10000$-.
3910
3911
3912 014124 005723 TST (R3)+ ;BUMP FOR NEXT PATTERN
3913 014126 020327 002666 CMP R3,#CSEND ;CHECK FOR END
3914 014132 001335 BNE CSTEST ;NOT END, LOAD NEXT PATTERN
3915
3916 014134 ENDSEG :****END OF SEGMENT****
(3) 014134 10000$
3917 014136 104005 EMT C$ESEG ;****END OF TEST****
(3) 014136
(3) 014136 104001 EMT C$ETST
3918
3919
3920 .SBTTL **TEST 6** - READ WRITE OF RLBA
3921
3922 014140 BGNST ;****START OF TEST****
3923 014140
3924 STARS
(2) ;*****
3925 ;TEST THAT WE CAN WRITE/READ BITS IS THRU 1 OF THE
3926 ;BUS ADDRESS REGISTER. FOUR PATTERNS ARE USED: GROWING 1. SHIFTING 1.
3927 ;GROWING 0 AND SHIFTING 0. BIT 0 IS ALSO LOADED BUT
3928 ;SHOULD ALWAYS COME BACK AS 0
3929 014140 STARS
(2) ;*****
3930
3931 014140 012703 002256 MOV #BEGPAT,R3 ;GET START OF PATTERN LIST
3932 014144 BGNSEG :****START OF SEGMENT****
(3) 014144 104004 EMT CSBSEG
3933
3934 BATEST:

```

M04

JCTERR MACY11 30,1046, 30-OCT-77 16:51 PAGE 84-19  
 C:RLAA.F11 05-OCT-77 10:41 \*\*\*TEST 6\*\* - READ WRITE OF RLBA

SEG 0051

3935	014146	011337	002224		MOV	(R3),GDDAT	:GET PATTERN TO SEND
3936	014152	005737	002252		TST	T.CNTR	:RL11??
3937	014156	001403			BEQ	2\$	:NO
3938	014160	042737	000001	002224	BIC	#BIT0,GDDAT	:KEEP RLBA EVEN (UNIBUS)
3939	014166	013777	002224	165726	MOV	GDDAT,RALBA	:LOAD PATTERN TO BUS ADDRESS
3940	014174	017737	165722	002226	MOV	RALBA,BDDAT	:READ IT BACK
3941	014202	023737	002224	002226	CMP	GDDAT,BDDAT	:IS IT CORRECT?
3942	014210	001404			BEQ	1\$	:IF SO, BRANCH
3943							
3944	014212				ERRDF	5,EM6,ERR2	:DATA WRONG IN RLBA
(3)	014212	104462			TRAP	T\$ERRCODE	
(5)	014214	000005			.WORD	5	
(5)	014216	004572			.WORD	EM6	
(5)	014220	007606			.WORD	ERR2	
3945	014222				ESCAPE	SEG	:IF /FL:LOE SET LOOP, ELSE EXIT SEG
(3)	014222	104010			EMT	C\$ESCAPE	
(3)	014224	000012			.WORD	10000\$-	

```

3947
3948
3949 014226 005723          TST      (R3)+          ;BUMP FOR NEXT PATTERN
3950 014230 020327 002464  CMP      R3,#ENDPAT    ;CHECK FOR END
3951 014234 001344          BNE      BATEST        ;NOT END, BRANCH FOR NEXT
3952
3953 014236          ENDSEG          ;****END OF SEGMENT****
   (3) 014236 10000$:
   (3) 014236 104005  EMT      C$ESEG
3954 014240          ENDTST          ;****END OF TEST****
   (3) 014240 L10024:
   (3) 014240 104001  EMT      C$ETST
3955
3956
3957 .SBTTL **TEST 7** - READ WRITE OF RLDA
3958
3959 014242          BGNTEST          ;****START OF TEST****
3960
3961 014242          STARS
   (2) ;*****
3962 ;TEST THAT WE CAN WRITE/READ THE DISK ADDRESS REGISTER
3963 ;ALL BIT POSITIONS ARE WRITTEN USING FOUR PATTERNS:
3964 ;GROWING 1, SHIFTING 1, GROWING 0 AND SHIFTING 0
3965 014242          STARS
   (2) ;*****
3966
3967
3968 014242 012703 002256          BGNSEG  MOV      #BEGPAT,R3      ;SET UP POINTER TO PATTERN LIST
3969 014246          EMT      C$BSEG      ;****START OF SEGMENT****
   (3) 014246 104004
3970 014250          DATEST:
3971 014250 011337 002224          MOV      (R3),GDDAT      ;GET PATTERN
3972 014254 013777 002224 165642  MOV      GDDAT,RALDA     ;LOAD PATTERN IN DA
3973
3974 014262 017737 165636 002226  MOV      RALDA,BDDAT     ;READ PATTERN BACK
3975 014270 023737 002224 002226  CMP      GDDAT,BDDAT     ;IS IT CORRECT?
3976 014276 001404          BEQ      1$              ;BRANCH IF CORRECT
3977
3978 014300          ERRDF  6,EM7,ERR2      ;WRONG DATA IN RLDA
   (3) 014300 104462          TRAP  T$ERRCODE
   (5) 014302 000006          .WORD  6
   (5) 014304 004620          .WORD  EM7
   (5) 014306 007606          .WORD  ERR2
3979 014310          1$:  ESCAPE  SEG          ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
   (3) 014310 104010          EMT      C$ESCAPE
   (3) 014312 000012          .WORD  10000$-.
3980
3981
3982 014314 005723          TST      (R3)+          ;BUMP POINTER
3983 014316 020327 002464  CMP      R3,#ENDPAT    ;AT END OF PATTERNS?
3984 014322 001352          BNE      DATEST        ;NO, BRANCH BACK
3985
3986 014324          ENDSEG          ;****END OF SEGMENT****
   (3) 014324 10000$:
   (3) 014324 104005  EMT      C$ESEG
3987 014326          ENDTST          ;****END OF TEST****

```

(3) 014326  
 (3) 014326 104001  
 39988  
 39989  
 39990  
 39991  
 39992 014330  
 39993 014330  
 (2)  
 39994  
 39995  
 39996  
 39997  
 39998 014330  
 (2)  
 39999  
 40000  
 40001 014330 012703 002570  
 40002 014334  
 (3) 014334 104004  
 40003 014336  
 40004 014336 012777 UU02UU 165554  
 40005 014344 011337 002224  
 40006 014350 052737 000200 002224  
 40007 014356 051377 165536  
 40008 014362 032777 040000 165530  
 40009 014370 001403  
 40010 014372 052737 140000 002224  
 40011 014400 017737 165514 002226  
 40012 014406 042737 000001 002226  
 40013 014414 023737 002226 002224  
 40014 014422 001404  
 40015  
 40016 014424  
 (3) 014424 104462  
 (5) 014426 000007  
 (5) 014430 005255  
 (5) 014432 007606  
 40017 014434  
 (3) 014434 104010  
 (3) 014436 000012  
 40018  
 40019  
 40020 014440 005723  
 40021 014442 022703 002666  
 40022 014446 001333  
 40023  
 40024 014450  
 (3) 014450  
 (3) 014450 104005  
 40025 014452  
 (3) 014452  
 (3) 014452 104001  
 40026  
 40027  
 40028

L10025: EMT CSETST  
 .SBTTL \*\*TEST 8\*\* - BIS OF RLCS  
 BGNSTST ;\*\*\*\*START OF TEST\*\*\*\*  
 STARS  
 ;\*\*\*\*\*  
 ;TEST THAT WE CAN USE THE "BIS" INSTRUCTION ON THE CONTROL  
 ;AND STATUS REGISTER. BITS 8,9 AND 6-1 ARE TESTED TO  
 ;SET INDIVIDUALLY AS WELL AS COLLECTIVELY WITHOUT DESTROYING  
 ;ANY PREVIOUS DATA PATTERN  
 STARS  
 ;\*\*\*\*\*  
 BGNSEG MOV #CSPAT,R3 ;GET BEGINNING OF LIST  
 ;\*\*\*\*START OF SEGMENT\*\*\*\*  
 EMT CSBSEG  
 1\$: MOV #CRDY,RLCS ;INSURE GO IS THERE  
 MOV (R3),GDDAT ;SET UP EXPECTED RLCS  
 BIS #CRDY,GDDAT ;IN GDDAT  
 BIS (R3),RLCS ;BIT SET PATTERN IN RLCS  
 BIT #DERR,RLCS ;IF ERROR BIT SET THEN  
 BEQ 99\$ ;EXPECT IT ON THE READ  
 BIS #ERR:DERR,GDDAT ;BACK  
 99\$: MOV RLCS,BDDAT ;READ RLCS TO CHECK "BIS"  
 BIC #DRDY,BDDAT ;CLEAR OUT DRIVE READY  
 CMP BDDAT,GDDAT ;DID BIS WORK?  
 BEQ 2\$ ;BRANCH IF OKAY  
 ERADF 7,EM61,ERR2 ;WRONG DATA IN RLCS  
 TRAP T\$ERRCODE  
 .WORD 7  
 .WORD EM61  
 .WORD ERR2  
 2\$: ESCAPE SEG ;IF /FL:LOE SET LOOP, ELSE EXIT SEG  
 EMT C\$ESCAPE  
 .WORD 10000\$- ;BIT OR CLEARED OTHER BIT  
 TST (R3)+ ;GET NEXT PATTERN  
 CMP #CSEND,R3 ;AT END OF LIST  
 BNE 1\$ ;NO GO BACK FOR TEST OF  
 ;NEXT PATTERN  
 ;\*\*\*\*END OF SEGMENT\*\*\*\*  
 ENDSEG  
 10000\$: EMT C\$ESEG  
 ;\*\*\*\*END OF TEST\*\*\*\*  
 ENDTST  
 L10026: EMT CSETST  
 .SBTTL \*\*TEST 9\*\* - BIC OF RLCS

```

4029
4030 014454          BGNTST          ;****START OF TEST****
4031
4032 014454          STARS
4033          ;*****
4034          ;TEST THAT THE "BIC" INSTRUCTION WILL WORK ON THE
4035          ;CONTROL AND STATUS REGISTER.  BITS 8-9 AND 6-1 ARE
4036          ;TESTED.
4037          STARS
4038          ;*****
4039 014454 012703 002570      BGNSEG  MOV      #CSPAT,R3      ;GET BEGINNING OF PATTERNS
4040 014460          ;*****START OF SEGMENT****
4041 (3) 014460 104004          EMT      C$BSEG
4042 014462          1$:
4043 014462 012777 001776 165430      MOV      #1776,RLCS      ;SET ALL SETTABLE BITS
4044 014470 012737 001776 002224      MOV      #1776,GDDAT     ;SET UP EXPECT DATA IN
4045 014476 041337 002224          BIC      (R3),GDDAT      ;GDDAT
4046 014502 041377 165412          BIC      (R3),RLCS      ;CLEAR BITS IN RLCS VIA "BIC"
4047 014506 032777 040000 165404      BIT      #DERR,RLCS     ;IF DRIVE ERROR BIT SET
4048 014514 001403          BEQ      99$            ;EXPECT IT SET WHEN WE
4049 014516 052737 140000 002224      BIS      #ERR!DERR,GDDAT ;READ IT BACK
4050 014524 017737 165370 002226      99$:  MOV      RLCS,BDDAT     ;MOVE RLCS TO BDDAT FOR COMPARE
4051 014532 042737 000001 002226      BIC      #DRDY,BDDAT     ;CLEAR DRIVE READY
4052 014540 023737 002226 002224      CMP      BDDAT,GDDAT    ;DID "BIC" WORK PROPERLY
4053 014546 001404          BEQ      2$            ;BRANCH IF OKAY
4054          ERRDF - B. EM62,ERR2      ;WRONG DATA IN RLCS
4055          TRAP  T$ERRCODE
4056          .WORD B
4057          .WORD EM62
4058          .WORD ERR2
4059 (3) 014560          2$:  ESCAPE  SEG      ;IF /FL:LOE SET LOOP. ELSE EXIT SEG
4060 014560 104010          EMT      C$ESCAPE
4061 014562 000012          .WORD  10000$-
4062
4063 014564 005723          TST      (R3)+          ;GET NEXT PATTERN
4064 014566 020327 002666      CMP      R3,#CSEND      ;AT END OF LIST
4065 014572 001333          BNE      1$            ;NO, GO BACK WITH NEXT PATTERN
4066 014574          ENOSEG
4067 (3) 014574 10000$:          EMT      C$ESEG      ;****END OF SEGMENT****
4068 014574 104005          EMT      C$ESEG
4069 014576          ENDTST
4070 (3) 014576 100027:      EMT      C$ETST      ;****END OF TEST****
4071 014576 104001          EMT      C$ETST
4072
4073          .SBTTL  **TEST 10** - BIS OF ALBA
4074
4075 014600          BGNTST          ;****START OF TEST****
4076
4077 014600          STARS
4078          ;*****
4079          ;TEST THAT THE "BIS" INSTRUCTION WILL WORK ON THE BUS
4080          ;ADDRESS REGISTER.  BITS 15-0 ARE LOADED.  ONLY BITS 15-1

```



:ARE EXPECTED BACK. FOUR PATTERNS ARE USED: GROWING I, SHIFTING I,  
:GROWING O, AND SHIFTING O.  
STARS  
:\*\*\*\*\*

014600  
014600 012703 002256  
014604 104004  
014606 005077 165310  
014612 011337 002224  
014616 005737 002252  
014622 001403  
014624 042737 000001 002224  
014632 051377 165264  
014636 017737 165260 002226  
014644 023737 002224  
014652 001404  
  
014654  
014654 104462  
014656 000011  
014660 006421  
014662 007606  
014664  
014664 104010  
014666 000012  
  
014670 005723  
014672 020327 002464  
014676 001343  
014700  
014700 104005  
014702  
014702 104001

BGNSEG MOV #BEGPAT,R3 :GET START OF LIST  
:\*\*\*\*START OF SEGMENT\*\*\*\*  
EMT C\$BSEG  
1\$: CLR @RLBA :CLEAR "BA"  
MOV (R3),GDDAT :SET EXPECTED  
TST T,CNTLR :RL11  
BEQ 3\$ :NO  
BIC #1,GDDAT :BIT 0 CAN'T SET IN RLBA (JNIBLS  
3\$: BIS (R3),@RLBA :BIS RLBA WITH PATTERN  
MOV @RLBA,BDDAT :READ "BA"  
CMP BDDAT,GDDAT :DID RLBA LOAD PROPERLY?  
BEQ 2\$ :BRANCH IF YES  
  
ERRDF 9,EM63,ERR2 :WRONG DATA IN RLBA  
TRAP T\$ERRCODE  
.WORD 9  
.WORD EM63  
.WORD ERR2  
2\$: ESCAPE SEG :IF FL:OE SET LOOP, ELSE EXIT SEG  
EMT C\$ESCAPE  
.WORD 10000\$-  
  
TST (R3)+ :GET NEXT PATTERN  
CMP R3,#ENDPAT :DID WE COMPLETE LIST  
BNE 1\$ :NO, GO BACK FOR NEXT.  
:\*\*\*\*END OF SEGMENT\*\*\*\*  
ENDSEG 10000\$:  
EMT C\$ESEG  
ENDTST :\*\*\*\*END OF TEST\*\*\*\*  
L10030: EMT C\$ESET

.SBTTL \*\*TEST 11\*\* - BIC OF RLBA

014704  
014704  
014704  
014704 012703 002256  
014710  
014710 104004

BGNST :\*\*\*\*START OF TEST\*\*\*\*  
STARS  
:\*\*\*\*\*  
:TEST THAT THE "BIC" INSTRUCTION WILL WORK ON THE BUS  
:ADDRESS REGISTER. BITS 15-1 ARE TESTED WITH 4 PATTERNS  
:GROWING I, SHIFTING I, GROWING O AND SHIFTING O.  
STARS  
:\*\*\*\*\*  
BGNSEG MOV #BEGPAT,R3 :GET START OF LIST  
:\*\*\*\*START OF SEGMENT\*\*\*\*  
EMT C\$BSEG

```

4112 014712 15: MOV #2, @RLBA ;SET RLBA TO ALL 1'S (BIT 0=C)
4113 014713 012777 177776 165202 MOV #2, @GDDAT ;SET UP EXPECTED RESULTS
4114 014720 012737 177776 002224 BIC (R3), @GDDAT ;IN GDDAT
4115 014726 041337 002224 BIC (R3), @RLBA ;BIC RLBA
4116 014732 041377 165164 MOV @RLBA, @BDDAT ;READ RLBA
4117 014736 017737 165160 002226 CMP @BDDAT, @GDDAT ;BIC WORK OKAY?
4118 014744 023737 002226 002224 BEQ 2$ ;IF YES BRANCH
4119 014752 001404
4120 014754 ERRDF 10, EM64, ERR2 ;WRONG DATA IN RLBA
4121 014754 104462 TRAP T$ERRCODE
4122 014756 000012 .WORD 10
4123 014760 006502 .WORD EM64
4124 014762 007606 .WORD ERR2
4125 014764 2$: ESCAPE SEG ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
4126 014764 104010 EMT C$ESCAPE
4127 014766 000012 .WORD 10000$-
4128 014770 005723 TST (R3)+ ;GET NEXT PATTERN
4129 014772 020327 002464 CMP R3, #ENDPAT ;HAVE WE COMPLETED LIST
4130 014776 001345 BNE 1$ ;NO, GO BACK FOR NEXT
4131 015000 ENOSEG ;****END OF SEGMENT****
4132 015000 10000$: EMT C$ESEG
4133 015000 104005 ENDTST ;****END OF TEST****
4134 015002 L10031: EMT C$ETST
4135 015002 104001
4136 .SBTTL **TEST 12** - BIS OF RLDA
4137 BGNSTST ;****START OF TEST****
4138 015004 STARS
4139 015004 ;*****
4140 015004 ;TEST THAT THE "BIS" INSTRUCTION WILL WORK ON THE DISK ADDRESS
4141 015004 ;REGISTER. BITS 15-0 ARE TESTED WITH 4 PATTERNS, GROWING 1,
4142 015004 ;SHIFTING 1, GROWING 0, AND SHIFTING 0.
4143 015004 STARS
4144 015004 ;*****
4145 015004 012703 002256 BGNSEG MOV #BEGPAT, R3 ;GET START OF LIST
4146 015010 104004 EMT C$BSEG ;****START OF SEGMENT****
4147 015012 15: CLR @RLDA ;CLEAR "DA"
4148 015012 005077 165106 MOV (R3), @GDDAT ;SET EXPECTED
4149 015016 011337 002224 BIS (R3), @RLDA ;BIS RLDA
4150 015022 051377 165076 MOV @RLDA, @BDDAT ;READ RLDA
4151 015026 017737 165072 002226 CMP @BDDAT, @GDDAT ;IS RLDA CORRECT
4152 015034 023737 002226 002224 BEQ 2$ ;IF OKAY BRANCH
4153 015042 001404
4154 015044 ERRDF 11, EM65, ERR2 ;WRONG DATA IN RLDA
4155 015044 104462 TRAP T$ERRCODE
4156 015046 000012 .WORD 11

```

```

(5) 015050 006565 .WORD EM65
(5) 015052 007606 .WORD ERR2
4153 015054 104010 2$: ESCAPE SEG ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
(3) 015054 104010 EMT C$ESCAPE
(3) 015056 000012 .WORD 10000$-.
4154 015060 005723 TST (R3)+ ;GET NEXT PATTERN
4155 015062 020327 002464 CMP R3,#ENDPAT ;HAVE WE FINISHED?
4156 015066 001351 BNE 1$ ;NO GO BACK
4158 015070 104005 ENOSEG ;****END OF SEGMENT****
(3) 015070 104005 EMT C$ESEG
4159 015072 104001 ENOTST ;****END OF TEST****
(3) 015072 104001 EMT C$ETST

.SBTTL **TEST 13** - BIC OF RLDA
4160 015074 BGNTST ;****START OF TEST****
4161 015074 STARS
4162 ;*****
4163 ;TEST THAT THE "BIC" INSTRUCTION WORKS ON THE DISK
4164 ;ADDRESS REGISTER. ALL BITS ARE TESTED WITH FOUR
4165 ;PATTERNS: GROWING 1, SHIFTING 1, GROWING 0 AND SHIFTING 0
4166 ;*****
(2) 015074
4167
4168
4169
4170 015074
(2)
4171
4172
4173 015074 012703 002256 BGNSEG MOV #BEGPAT,R3 ;GET START OF LIST
4174 015100 104004 EMT C$BSEG ;****START OF SEGMENT****
(3) 015102 1$:
4175 015102 012777 177777 165014 MOV #-1,2RLDA ;SET RLDA TO ALL 1'S
4176 015110 012737 177777 002224 MOV #-1,GDDAT ;SET EXPECTED DATA
4177 015116 041337 002224 BIC (R3),GDDAT ;SET EXPECTED DATA
4178 015122 041377 164776 BIC (R3),2RLDA ;"BIC" RLDA
4179 015126 017737 164772 002226 MOV 2RLDA,BDDAT ;READ RLDA
4180 015134 023737 002224 002226 CMP GDDAT,BDDAT ;DID "BIC" WORK?
4181 015142 001404 BEQ 2$ ;IF IT DID BRANCH
4182
4183 ERROF 12,EM66,ERR2 ;WRONG DATA IN RLDA
4184 015144 104462 TRAP T$ERRCODE
(5) 015146 000014 .WORD 12
(5) 015150 006646 .WORD EM66
(5) 015152 007606 .WORD ERR2
4185 015154 104010 2$: ESCAPE SEG ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
(3) 015154 104010 EMT C$ESCAPE
(3) 015156 000012 .WORD 10000$-.
4186 015160 005723 TST (R3)+ ;GET NEXT PATTERN
4187 015162 020327 002464 CMP R3,#ENDPAT ;DONE?
4188 015166 001345 BNE 1$ ;NO GO BACK
4189 015170 104005 ENOSEG ;****END OF SEGMENT****
(2) 015170 104005 EMT C$ESEG

```

015170 104005  
015172 104001

ENDTST EMT CSESEG :\*\*\*\*END OF TEST\*\*\*\*  
L10033:  
EMT CSETST

.SBTTL \*\*TEST 14\*\* - BUS RESET OF RLCS

015174  
015174

BGNTST :\*\*\*\*START OF TEST\*\*\*\*

STARS

:\*\*\*\*\*  
:TEST THAT A BUS RESET WILL CLEAR THE PROPER BITS  
:OF THE CONTROL AND STATUS REGISTER. THOSE BITS ARE  
:BITS 6-1,8,9,10,11,12,13,15. BIT 15 WILL CLEAR ONLY  
:IF BIT 14 (DRIVE ERROR IS NOT SET). BIT 0 (DRIVE READY)  
:IS A DON'T CARE. IF AT THE START UP THIS TEST BIT  
:14 (DRIVE ERROR) IS SET WE WILL INSIST IF IS THERE AFTER  
:THE "RESET" ALONG WITH BIT 15 (COMPOSITE ERROR). BITS  
:15-10 ARE NOT WRITEABLE.

015174

STARS

:\*\*\*\*\*

015174 104033  
015174 005300  
015200 104041  
015202 012777 000377 164710  
015210 012737 000200 002224  
015216 032777 040000 164674  
015224 001403  
015226 052737 140000 002224  
015234 012700 000100  
015240 104033  
015242 005300  
015244 001376  
015246 017737 164646 002226  
015254 042737 000001 002226  
015262 023737 002226 002224  
015270 001404

SETPRI #PRI07 :PRIORITY TO SEVEN  
MOV #PRI07,RO  
EMT C\$SPRI  
MOV #377,@RLCS :LOAD ALL RLCS LOADABLE BITS  
MOV #CRDY,GDDAT :SETUP EXPECTED  
BIT #DERR,@RLCS :DRIVE ERR SET?  
BEQ IS :IF NOT DON'T EXPECT IT  
BIS #DERR!ERR,GDDAT :IT'S SET, INIT BETTER NOT CLR  
1\$: MOV #100,RO :SET UP A WAIT LOOP  
BRESE :BUS RESET  
EMT C\$RESET  
2\$: DEC RO :WAIT IN CASE OF DRIVE ERROR  
BNE 2\$  
MOV @RLCS,BDDAT :READ RLCS  
BIC #DRDY,BDDAT :CLEAR OUT DRDY - DON'T CARE  
CMP BDDAT,GDDAT :DID INIT WORK  
BEQ 3\$ :YES. BRANCH

015272 104462  
015274 000015  
015276 006731  
015300 007606

ERRDF 13,EM67,ERR2 :WRONG DATA IN RLCS  
TRAP T\$ERRCODE  
.WORD 13  
.WORD EM67  
.WORD ERR2

015302  
015302  
015302 104001

3\$: ENDTST :\*\*\*\*END OF TEST\*\*\*\*  
L10034:  
EMT CSETST

.SBTTL \*\*TEST 15\*\* - BUS RESET OF RLBA

015304

BGNTST :\*\*\*\*START OF TEST\*\*\*\*

```

015304 STARS
:*****
:TEST THAT A BUS RESET WILL CLEAR THE ENTIRE
:BUS ADDRESS REGISTER. THE BUS ADDRESS IS LOADED WITH 177776
:AND IS EXPECTED TO BE ZERO AFTER THE RESET
015304 STARS
:*****
015304 012777 177776 164610 MOV #2,RLBA :SET BA TO ALL 1'S
015312 005737 002252 TST T.CNTRL :RL11??
015316 001403 BEQ 2$ :NO
015320 052777 000001 164574 BIS #1,RLBA
015326 005037 002224 2$: CLR GDDAT :CLEAR EXPECTED DATA
015332 BRESET :ISSUE BUS INIT
015338 104033 EMT CSRESET
015344 017737 164562 002226 MOV RLBA,BDDAT :READ RLBA
015348 001404 BEQ 1$ :IF CLEAR BRANCH
015344 ERDF 14,EM70.ERR2 :WRONG DATA IN RLBA
015344 TRAP TSEARCHCODE
015346 .WORD 14
015350 .WORD EM70
015354 .WORD ERR2
1$:
015354 ENDTST :****END OF TEST****
015354 L10035: EMT CSETST
015354 104001
    
```

.SBTTL \*\*TEST 16\*\* - BUS RESET OF RLDA

```

015356 BGNTST :****START OF TEST****
015356 STARS
:*****
:TEST THAT A BUS RESET WILL CLEAR THE ENTIRE
:DISK ADDRESS REGISTER. THE DISK ADDRESS IS LOADED WITH 177777
:AND IS EXPECTED TO BE ZERO AFTER THE RESET.
015356 STARS
:*****
015356 012777 177777 164540 MOV #1,RLDA :SET DA TO ALL 1'S
015364 005037 002224 CLR GDDAT :CLEAR EXPECTED
015370 BRESET :ISSUE BUS INIT
015376 104033 EMT CSRESET
015382 017737 164526 002226 MOV RLDA,BDDAT :READ RLDA
015400 001404 BEQ 1$ :IF CLEAR BRANCH
015402 ERDF 15,EM71.ERR2 :WRONG DATA IN RLDA
015402 TRAP TSEARCHCODE
015404 .WORD 15
015406 .WORD EM71
    
```



10316  
10317  
10318  
10319  
10320  
10321  
10322  
10323  
10324  
10325  
10326  
10327  
10328  
10329  
10330  
10331  
10332  
10333  
10334  
10335  
10336  
10337  
10338  
10339  
10340  
10341  
10342  
10343  
10344  
10345  
10346  
10347  
10348  
10349  
10350  
10351  
10352  
10353  
10354  
10355  
10356  
10357  
10358  
10359  
10360

015536  
015536  
015536 104001

ENDTST ;\*\*\*\*END OF TEST\*\*\*\*  
L10037: EMT CSETST

.SBTTL \*\*TEST 18\*\* - UNIQUENESS OF RLBA

015540  
015540

BGNTST ;\*\*\*\*START OF TEST\*\*\*\*  
STARS  
:\*\*\*\*\*  
:TEST THE UNIQUENESS OF THE BUS ADDRESS REGISTER. THE  
:RLCS AND RLDA ARE LOADED WITH XXX20X AND 177777  
:RESPECTIVELY. THE RLBA IS THEN WRITTEN TO INSURE  
:THAT NEITHER THE RLCS OR RLDA ARE MODIFIED  
:BY WRITING THE RLBA.  
STARS  
:\*\*\*\*\*

015540

015540 012737 000200 002224  
015546 032777 040000 164344  
015554 001403  
015556 052737 140000 002224  
015564 013777 002224 164326  
015572 012777 177777 164324  
015600 005077 164316

MOV #CRDY,GDDAT ;CONTROLLER READY  
BIT #DERR,RLCS ;IF DRIVE ERROR IS  
BEQ 99\$ ;SET THEN EXPECT IT  
BIS #ERR!DERR,GDDAT ;SET WHEN WE READ IT.  
99\$: MOV GDDAT,RLCS ;LOAD RLCS  
MOV #-1,RLDA ;LOAD RLDA  
CLR RLBA ;CLEAR RLBA

;CHECK IF RLCS IS OKAY

015604 017737 164310 002226  
015612 042737 000001 002226  
015620 023737 002226 002224  
015626 001404

MOV RLCS,BDDAT ;READ RLCS  
BIC #DRDY,BDDAT ;IGNORE DRIVE READY  
CMP BDDAT,GDDAT ;CS OK?  
BEQ IS ;YES, GO CHECK DA

015630  
015630 104462  
015632 000022  
015634 007146  
015636 007606  
015640  
015640 104006

ERRDF 18,EM74,ERR2 ;BA MODIFIED CS  
TRAP T\$ERRCODE  
.WORD 18  
.WORD EM74  
.WORD ERR2  
IS: CKLOOP ;CHECK IF FL:LOE IS SET  
EMT CSCLP1

015642 022777 177777 164254  
015650 001412  
015652 012737 177777 002224  
015660 017737 164240 002226

CMP #-1,RLDA ;IS RLDA OKAY?  
BEQ 2\$ ;IF OKAY BRANCH  
MOV #-1,GDDAT ;SET UP EXPECTED  
MOV RLDA,BDDAT ;READ RLDA

015666  
015666 104462  
015670 000023  
015672 007200  
015674 007606

ERRDF 19,EM75,ERR2 ;BA MODIFIED DA  
TRAP T\$ERRCODE  
.WORD 19  
.WORD EM75  
.WORD ERR2

K05

OUTERR MACY11 30(1046 30-OCT-77 14:51 PAGE 85-10  
DZRLAA.P11 05-OCT-77 10:41

\*\*TEST 18\*\* - UNIQUENESS OF RLBA

SEG 0062

4359 015676  
4360 015676  
(3) 015676  
(3) 015676 104001

2\$:  
ENDTST :\*\*\*\*END OF TEST\*\*\*\*  
L10040: EMT C\$ETST

4361  
4362  
4363  
4364  
4365 015700  
4366  
4367  
4368 015700  
(2)

.SBTTL \*\*TEST 19\*\* - UNIQUENESS OF RLDA  
BGNTST :\*\*\*\*START OF TEST\*\*\*\*

4369  
4370  
4371  
4372  
4373  
4374 015700  
(2)

STARS  
:\*\*\*\*\*  
:TEST THE UNIQUENESS OF THE DISK ADDRESS REGISTER. THE RLCS  
:AND RLBA ARE LOADED WITH XXX20X AND 177776  
:RESPECTIVELY. THE RLDA IS THEN WRITTEN TO INSURE  
:THAT NEITHER THE RLCS OR THE RLBA ARE MODIFIED  
:BY WRITING THE RLDA.  
STARS  
:\*\*\*\*\*

4375  
4376  
4377 015700 012737 000200 002224  
4378 015706 032777 040000 164204  
4379 015714 001403  
4380 015716 052737 140000 002224  
4381 015724 013777 002224 164166  
4382 015732 012777 177776 164162  
4383 015740 005077 164160

99\$: MOV #CRDY,GDDAT :CONTROLLER READY  
BIT #DERR,@RLCS :IF DRIVE ERROR SET  
BEQ 99\$ :THEN EXPECT IT LATER  
BIS #ERR!DERR,GDDAT  
99\$: MOV GDDAT,@RLCS :LOAD CS  
MOV #-2,@RLBA :LOAD BA WITH ALL 1'S  
CLR @RLDA :CLEAR RLDA

4384  
4385  
4386  
4387 015744 017737 164150 002226  
4388 015752 042737 000001 002226  
4389 015760 023737 002224 002226  
4390 015766 001404

;CHECK IF RLCS IS OKAY  
MOV @RLCS,BDDAT :READ RLCS  
BIC #DRDY,BDDAT :IGNORE DRIVE READY  
CMP GDDAT,BDDAT :RLCS OKAY?  
BEQ 1\$ :YES, THEN BRANCH

4391 015770  
(3) 015770 104462  
(5) 015772 000024  
(5) 015774 007232  
(5) 015776 007606  
4393 016000  
(3) 016000 104006

ERRDF 20,EM76,ERR2 :DA MODIFIED CS  
TRAP T\$ERRCODE  
.WORD 20  
.WORD EM76  
.WORD ERR2  
1\$: CKLOOP :CHECK IF /FL:LOE IS SET  
EMT C\$CLP1

4394  
4395 016002 022777 177776 164112  
4396 016010 001412  
4397  
4398 016012 012737 177776 002224  
4399 016020 017737 164076 002226  
4400

CMP #-2,@RLBA :IS RLBA OKAY?  
BEQ 2\$ :BRANCH IF OKAY  
MOV #-2,GDDAT :SET UP EXPECTED  
MOV @RLBA,BDDAT :READ RLBA

4401 016026  
(3) 016026 104462  
(5) 016030 000025  
(5) 016032 007265  
(5) 016034 007606

ERRDF 21,EM77,ERR2 :DA MODIFIED BA  
TRAP T\$ERRCODE  
.WORD 21  
.WORD EM77  
.WORD ERR2



016036 25:

016036 ENDTST ;\*\*\*\*END OF TEST\*\*\*\*  
016036 L10041:  
016036 104001 EMT CSETST

.SBTTL \*\*TEST 20\*\* - UNIQUENESS OF RLMP  
016040 BGNST ;\*\*\*\*START OF TEST\*\*\*\*

016040 STARS  
:\*\*\*\*\*  
:TEST THE UNIQUENESS OF THE MULTI-PURPOSE REGISTER  
:WE WILL WRITE THE RLCS, RLBA, AND THE RLDA, THEN THE  
:RLMP IS WRITTEN. WE THEN GO BACK AND VERIFY THE CONTENTS  
:OF THE RLCS, RLBA, RLDA.  
STARS  
:\*\*\*\*\*

016040 012737 000200 002224 MOV #CRDY, GDDAT ;CONTROLLER READY  
016046 032777 040000 164044 BIT #DERR, @RLCS ;IF DRIVE ERROR SET  
016054 001403 BEQ 99\$ ;THE EXPECT IT LATER  
016056 052737 140000 002224 BIS #ERR!DERR, GDDAT  
016064 013777 002224 164026 99\$: MOV GDDAT, @RLCS ;LOAD CS  
016072 012777 177776 164022 MOV #-2, @RLBA ;LOAD BA WITH ALL 1'S  
016100 012777 177777 164016 MOV #-1, @RLDA ;LOAD RLDA  
016106 005077 164014 CLR @RLMP ;WRITE RLMP

;CHECK IF RLCS IS OKAY

016112 017737 164002 002226 MOV @RLCS, BDDAT ;READ RLCS  
016120 042737 000001 002226 BIC #DRDY, BDDAT ;IGNORE DRIVE READY  
016126 023737 002224 002226 CMP GDDAT, BDDAT ;RLCS OKAY?  
016134 001404 BEQ IS ;YES, THEN BRANCH

016136 ERRDF 201, EM44, ERR2 ;MP MODIFIED CS  
016136 104462 TRAP T\$ERRCODE  
016140 000311 .WORD 201  
016142 005601 .WORD EM44  
016144 007606 .WORD ERR2

IS: CKLOOP ;CHECK IF /FL:LOE IS SET  
EMT C\$CLP1

016150 022777 177776 163744 CMP #-2, @RLBA ;IS RLBA OKAY?  
016156 001412 BEQ 25 ;BRANCH IF OKAY

016160 012737 177776 002224 MOV #-2, GDDAT ;SET UP EXPECTED  
016166 017737 163730 002226 MOV @RLBA, BDDAT ;READ RLBA

016174 ERRDF 211, EM45, ERR2 ;MP MODIFIED BA  
016174 104462 TRAP T\$ERRCODE  
016176 000323 .WORD 211  
016200 005634 .WORD EM45

MOS

```

(5) 016202 007606
4446 016204
(3) 016204 104006
4447 016206 022777 177777 163710
4448 016214 001412
4449
4450 016216 017737 163702 002226
4451 016224 012737 177777 002224
4452
4453 016232
(2) 016232 104462
(5) 016234 000324
(5) 016236 005667
(5) 016240 007606
4454
4455 016242
4456
4457
4458 016242
(3) 016242
(3) 016242 104001
4459
4460
4461 016244
4462
4463
4464
4465 016244
(2)
4467
4468
4469
4470 016244
(2)
4472
4473
4474 016244 005737 002252
4475 016250 001410
4476
4477 016252 004537 012552
4478 016256 000000
4480 016260 004537 013362
4481 016264
(3) 016264 104006
4482
4483 016266 004537 012252
4484
4485 016272
4486 016272
(3) 016272
(3) 016272 104001
4487
4488
  
```

```

      .WORD  ERR2
2$:  CKLOOP
      EMT  CSCLP1      ;CHECK IF /FL:LOE IS SET
      CMP  #-1,RLDA   ;DISK ADDRESS OKAY
      BEQ  3$         ;YES, CONTINUE

      MOV  RLDA,BDDAT ;SET UP BAD
      MOV  #-1,GDDAT  ;SET UP EXPECTED

      ERDF 212,EM46,ERR2 ;MP MODIFIED DA
      TRAP T$ERCODE

      .WORD 212
      .WORD EM46
      .WORD ERR2

3$:

      ENDTST
L10042:
      EMT  C$ETST
      .SBTTL **TEST 21** - NOOP FUNCTION(RL11 ONLY)
      B$NTST
      ;****START OF TEST****

      STARS
      ;*****
      ;TEST THAT NOOP WILL FUNCTION. WE WILL ISSUE THE
      ;NOOP AND WAIT FOR CONTROLLER READY TO SET. A
      ;TIMEOUT OF 200 MILLISECS IS ALLOWED. DRIVE 0 IS ALWAYS
      ;SELECTED SINCE THE DRIVE IS NOT NECESSARY.
      STARS
      ;*****

      TST  T.CNTRL      ;RLV11??
      BEQ  99$         ;YES SKIP TEST

      JSR  R5,LDFUNC   ;ISSUE FUNCTION OF FOLLOWING WORD
      NOOP
      JSR  R5,WTCRDY   ;NOOP(0) FUNCTION
      ;WAIT FOR CONTROLLER READY HIGH
2$:  CKLOOP
      EMT  CSCLP1      ;CHECK IF /FL:LOE IS SET

      JSR  R5,CHERR    ;CHECK CONTROLLER FOR ERRORS

99$:
      ENDTST
L10043:
      EMT  C$ETST
  
```

# N05

OUTERR MACY11 30(1046) 30-OCT-77 16:51 PAGE 85-13  
 DZRLAA.P11 05-OCT-77 10:41

\*\*TEST 22\*\* - TEST NOOP DOES NOTHING

SEQ 0065

```

4489
4490
4491 016274
4492
4493 016274
4494 (2)
4495
4496 016274
4497 (2)
4498 016274 005737 002252
4499 016300 001476
4500
4501 016302 012777 000001 163614
4502 016310 012777 000002 163604
4503 016316 005077 163604
4504 016322 017737 163600 002224
4505
4506 016330 004537 012552
4507 016334 000000
4508 016336 004537 013362
4509 016342
4510 (3) 016342 104006
4511 016344 004537 012252
4512 016350
4513 (3) 016350 104010
4514 (3) 016352 000124
4515
4516 016354 017737 163546 002226
4517 016362 023737 002224 002226
4518 016370 001404
4519 016372
4520 (3) 016372 104462
4521 (5) 016374 000312
4522 (5) 016376 004741
4523 (5) 016400 007606
4524
4525 016402
4526 (3) 016402 104006
4527 016404 012737 000002 002224
4528 016412 017737 163504 002226
4529 016420 023737 002224 002226
4530 016426 001404
4531
4532 016430
4533 (3) 016430 104462
4534 (5) 016432 000313
4535 (5) 016434 004767
4536 (5) 016436 007606
4537
4538 016440
4539 (3) 016440 104006
  
```

```

.SBTTL **TEST 22** - TEST NOOP DOES NOTHING
BGNTST ;****START OF TEST****
STARS
;*****
;TEST THAT ISSUING A NOOP FUNCTION DOES NOTHING. THIS IS DONE BY WRITING
;THE RLBA, AND RLDA, READING THE RLMP AND MAKING SURE NOTHING CHANGES.
STARS
;*****
TST T.CNTRL ;RLV11??
BEQ 3$
MOV #1, @RLDA ;LOAD DISK ADDRESS
MOV #2, @RLBA ;LOAD BUS ADDRESS
CLR @RLMP
MOV @RLMP, GDDAT ;READ RLMP
JSR R5, LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
NOOPD
JSR R5, WTCRDY ;WAIT FOR CONTROLLER READY HIGH
CKLOOP ;CHECK IF /FL:LOE IS SET
EMT C$CLP1
JSR R5, CHERR ;CHECK CONTROLLER FOR ERRORS
ESCAPE TST ;IF /FL:LOE SET LOOP, ELSE EXIT TST
EMT C$ESCAPE
.WORD L10044-
MOV @RLMP, BDDAT ;READ RLMP
CMF GDDAT, BDDAT ;RLMP OK?
BEQ 1$
ERRDF 202, EM14, ERR2
TRAP T$ERRCODE
.WORD 202
.WORD EM14
.WORD ERR2
1$: CKLOOP ;CHECK IF /FL:LOE IS SET
EMT C$CLP1
MOV #2, GDDAT ;SET UP EXP'D BA
MOV @RLBA, BDDAT ;READ BA
CMP GDDAT, BDDAT ;BA OK?
BEQ 2$ ;YES
ERRDF 203, EM15, ERR2
TRAP T$ERRCODE
.WORD 203
.WORD EM15
.WORD ERR2
2$: CKLOOP ;CHECK IF /FL:LOE IS SET
EMT C$CLP1
  
```

```

4530
4531 016442 012737 000001 002224      MOV      #1,GDDAT      ;SET UP EXP'D DA
4532 016450 017737 163450 002226      MOV      @RLOA,BDDAT  ;READ DA
4533 016456 023737 002224 002226      CMP      GDDAT,BDDAT  ;DA OKAY
4534 016464 001404
4535
4536 016466
(2) 016466 104462      ERDF     204,EM16,ERR2
(5) 016470 000314      TRAP    T$ERCODE
(5) 016472 005015      .WORD   2C4
(5) 016474 007606      .WORD   EM16
      .WORD   ERR2
4537
4538 016476      3$:
4539
4540 016476      ENDTST
(2) 016476      L10044:      ;****END OF TEST****
(3) 016476 104001      EMT      C$ETST
4541
4542
4543      .SBTTL  **TEST 23** - TEST OF INTERRUPT
4544
4545 016500      BGNST      ;****START OF TEST****
4546
4547 016500
(2)
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557 016500
(2)
4558
4559
4560 016500 005737 002252      TST     T,CNTRL
4561 016504 001426      BEQ     99$
4562
4563 016506 005037 002172      CLR     INTFLG      ;CLEAR INTERRUPT OCCURANCE FLAG
4564 016512
(3) 016512 012700 000000      SETPRI  #PRI00      ;SET PSW TO 0
(3) 016516 104041      MOV     #PRI00,RO
4565 016520 004537 012552      EMT     C$SPRI
4566 016524 000100      JSR     R5,LDFUNC   ;ISSUE FUNCTION OF FOLLOWING WORD
4567 016526 004537 013362      JSR     R5,WTCRDY  ;NOOP AND INTERRUPT ENABLE
4568 016532 005737 002172      TST     INTFLG     ;WAIT FOR CONTROLLER READY HIGH
4569 016536 001004      BNE     2$        ;DID INTERRUPT OCCUR
4570 016540
(2) 016540 104462      ERDF     22,EM13,ERR0 ;IF SO BRANCH
(5) 016542 000026      TRAP    T$ERCODE
(5) 016544 004707      .WORD   22
(5) 016546 007556      .WORD   EM13
4571 016550 005037 002172      .WORD   ERR0
      2$:      CLR     INTFLG

```

```

4572 016554 CKLOOP :CHECK IF /FL:LOE IS SET
(3) 016554 104006 EMT C$CLP1
4573 016556 004537 012252 JSR R5,CHERR :CHECK CONTROLLER FOR ERRORS
4574
4575 016562 99$:
4576 016562 ENOTS:
(3) 016562 L10045: :****END OF TEST****
4577 016562 104001 EMT C$ETST
4578
4579 .SBTTL **TEST 24** - TEST PRIORITY BR LEVEL
4580
4581 016564 BGNTST :****START OF TEST****
4582 016564
4583 STARS
4584 :*****
4585 :TEST THAT PRIORITY GIVEN IS ACTUAL PRIORITY OF CONTROLLER. WE KNOW
4586 :THE BOARD WILL INTERRUPT. WE WILL START TRYING TO INTERRUPT AT 7
4587 :AND WORK DOWN TIL IT DOES INTERRUPT.
4588 016564 STARS
4589 :*****
4590 016564 005737 002252 TST T,CNTRL :RLV11??
4591 016570 001456 BEQ 6$ :YES, SKIP TEST
4592
4593 016572 012737 000340 002226 MOV #340,BDDAT :SET UP INITIAL OF 7
4594 016600 013737 002132 002224 MOV BPRIOR,GDDAT :GET GIVEN PRIORITY
4595
4596 016606 BGNSEG :****START OF SEGMENT****
(3) 016606 104004 EMT C$BSEG
4597
4598 016610 005037 002172 $$: CLR INTFLG :CLEAR INTERRUPT OCCURANCE
4599 016614 002226 SETPRI BDDAT :SET PRIORITY
(3) 016614 013700 MOV BDDAT,RO
(3) 016620 104041 EMT C$SPRI
4600
4601 016622 004537 012552 JSR R5,LDFUNC :ISSUE FUNCTION OF FOLLOWING WORD
4602 016626 000100 NOOPO!INTEN
4603
4604 016630 004537 013362 JSR R5,WTCRDY :WAIT FOR CONTROLLER READY HIGH
4605 016634 ESCAPE TST :IF /FL:LOE SET LOOP, ELSE EXIT TST
(3) 016634 104010 EMT C$ESCAPE
(3) 016636 000070 .WORD L10046-.
4606
4607 016640 004537 012252 JSR R5,CHERR :CHECK CONTROLLER FOR ERRORS
4608 016644 ESCAPE TST :IF /FL:LOE SET LOOP, ELSE EXIT TST
(3) 016644 104010 EMT C$ESCAPE
(3) 016646 000060 .WORD L10046-.
4609
4610 016650 023737 002226 002224 CMP BDDAT,GDDAT :SHOULD IT INTERRUPT
4611 016656 002012 BGE 1$ :NO, BRANCH
4612
4613 016660 005737 002172 TST INTFLG :DID INTERRUPT OCCUR
4614 016664 001004 BNE 2$ :YES, OK
4615

```

```

4616 016666 35: ERRDF 204, EM17, ERR7
      (3) 016666 104462 TRAP T$ERRCODE
      (5) 016670 000314 .WORD 204
      (5) 016672 005043 .WORD EM17
      (5) 016674 010044 .WORD ERR7

4618 016676 25: ESCAPE SEG ; IF /FL:LOE SET LOOP, ELSE EXIT SEG
      (3) 016676 104010 EMT C$ESCAPE
      (3) 016700 000014 .WORD 10000$.
4619 016702 000405 BR 4$
4620 016704 005737 002172 15: TST INTFLG ; DID INTERRUPT OCCUR
4621 016710 001772 BEQ 25 ; NO OK
4622 016712 000765 BR 35 ; YES, ERROR

4623 016714 ENDSEG ; ****END OF SEGMENT****
      (3) 016714 10000$:
      (3) 016714 104005 EMT C$ESEG
4625 016716 162737 000040 002226 45: SUB #40, BDDAT ; NEXT LEVEL
4626 016724 100331 BPL 5$

4627 016726 65: ; ****END OF TEST****
      (3) 016726 ENDTST
      (3) 016726 L10046: EMT C$SETST
      (3) 016726 104001

4630 .SBTTL **TEST 25** - GET STATUS FUNCTION
4631 BGNTST ; ****START OF TEST****
4632 016730
4633 STARS
4634 ; *****
4635 ; TEST GET STATUS FUNCTION. THE GET STATUS FUNCTION WILL
4636 ; WORK IF DRIVE IS LOADED AND READY OR NOT. THE ALDA
4637 ; IS LOADED WITH THE GET STATUS AND MARKER BITS (BITS 1,0)
4638 ; AND THE FUNCTION IS ISSUED. WE WAIT 200 MILLISECONDS
4639 ; FOR CONTROLLER READY. VERIFY THAT NO ERRORS OCCUR.
4640 STARS
4641 ; *****
4642 016730
4643
4644 016730 012777 000013 163166 MOV #GSBIT!MK!DRST, @ALDA ; SET GET STATUS AND MARKER BIT
4645 016736 004537 012552 JSR R5, LDFUNC ; ISSUE FUNCTION OF FOLLOWING WORD
4646 016742 000004 GSTAT ; GET STATUS
4647 016744 004537 013362 JSR R5, WTCRDY ; WAIT FOR CONTROLLER READY HIGH
4648 016750 25: CKLOOP ; CHECK IF /FL:LOE IS SET
      (3) 016750 104006 EMT C$CLP1
4649 016752 004537 012252 JSR R5, CHERR ; CHECK CONTROLLER FOR ERRORS
4650 016756 ENDTST ; ****END OF TEST****
      (3) 016756 L10047:
      (3) 016756 104001 EMT C$SETST

4651 .SBTTL **TEST 26** - GET STATUS FUNCTION INTERRUPT

```

```

4657
4658 016760          BGNTST          ;****START OF TEST****
4659
4660          :CHECK GET STATUS UNDER INTERRUPT
4661
4662
4663 016760 005037 002172          CLR          INTFLG          ;CLEAR INTERRUPT OCCURANCE
4664 016764          SETPRI          #PRI00          ;PSW TO LEVEL 0
4665 (3) 016764 012700 000000          MOV          #PRI00,RO
4666 (3) 016770 104041          EMT          C$SPRI
4667 016772 012777 000003 163124          MOV          #GSBIT!MK, @RLDA ;SET UP DA
4668 017000 004537 012552          JSR          R5, LDFUNC          ;ISSUE FUNCTION OF FOLLOWING WORD
4669 017004 000104          GSTAT!INTEN          ;GET STATUS INT ENABLE
4670 017006 004537 013362          JSR          R5, WTCRDY          ;WAIT FOR CONTROLLER READY HIGH
4671 (3) 017012          SETPRI          #PRI07
4672 (3) 017012 012700 000340          MOV          #PRI07,RO
4673 (3) 017016 104041          EMT          C$SPRI
4674 017020 005737 002172          TST          INTFLG          ;DID INTERRUPT OCCUR
4675 017024 001004          BNE          2$          ;YES-BRANCH
4676 (3) 017026 104462          ERRDF          2$, EM30, ERRO
4677 (5) 017030 000034          TRAP          T$ERCODE
4678 (5) 017032 005076          .WORD          28
4679 (5) 017034 007558          .WORD          EM30
4680 017036          .WORD          ERRO
4681 (3) 017036 104006          2$: CKLCOPI          ;CHECK IF /FL:LOE IS SET
4682 (3) 017036          EMT          C$CLP1
4683
4684 017040 004537 012252          JSR          R5, CHERR          ;CHECK CONTROLLER FOR ERRORS
4685
4686 017044          ENDTST          ;****END OF TEST****
4687 (3) 017044          L10050:
4688 (3) 017044 104001          EMT          C$SETST
4689
4690          .SBTTL **TEST 27** - GET STATUS FUNCTION GENERATES OPI W/O GS BIT
4691          BGNTST          ;****START OF TEST****
4692
4693          STARS
4694          ;*****
4695          ;VERIFY THAT GET STATUS FUNCTION WILL NOT COMPLETE
4696          ;WITHOUT SENDING OUT THE GET STATUS BIT IN THE RLDA.
4697          ;WE SET MARKER BUT NO GET STATUS BIT IN THE RLDA AND
4698          ;ISSUE A GET STATUS WE SHOULD RECIEVE AN OPI ERROR.
4699          ;VERIFY THAT CONTROLLER READY SETS AND OPI SETS
4700          STARS
4701          ;*****
4702
4703 017046 012777 000001 163050          MOV          #MK, @RLDA          ;SET ONLY MARKER BIT!!
4704 017054 004537 012552          JSR          R5, LDFUNC          ;ISSUE FUNCTION OF FOLLOWING WORD
4705 017060 000004          GSTAT          ;GET STATUS
4706 017062 004537 013362          JSR          R5, WTCRDY          ;WAIT FOR CONTROLLER READY HIGH
4707 017066 032737 074000 002152          BIT          #74000, E.CS
4708 017074 001405          BEQ          1$
4709 017076 012737 003510 012534          MOV          #OPIERR, RESTMS

```

```

017100 017104 004537 012252          JSR      R5,CHERR
017101 017110          1S:      CKLOOP
017110 017110 104006          EMT      C$CLP1
017112 017112 032737 002000 002152  BIT      #OPI,E.CS          ;IS OPI SET?
017120 017120 001004          BNE      2$              ;YES-BRANCH NO-CHECK TIMEOUT
017122 017122          ERDF     29,EM33,ERRO
017124 017124 104462          TRAP    T$ERCODE
017126 017126 000035          .WORD   29
017130 017130 005171          .WORD   EM33
017132 017132 007556          .WORD   ERRO
017132          2$:
017132          ENDTST          ;****END OF TEST****
017132 L10051:
017132 104001          EMT      C$ETST

.SBTTL **TEST 28** - OPI UNDER INTERRUPT
017134          BGNST          ;****START OF TEST****
017134          STARS
;*****
;FORCE AN OPI ERROR UNDER INTERRUPT TO VERIFY THAT
;AN INTERRUPT WILL OCCUR FROM OPI. THE OPI IS FORCED
;USING A GET STATUS WITHOUT THE GET STATUS BIT SET
;IN RLDA.
017134          STARS
;*****

017134          SETPRI  #PRI00
017134 012700 000000          MOV      #PRI00,R0
017140 104041          EMT      C$SPRI
017142 005037 002172          CLR      INTFLG
017146 012777 000001 162750          MOV      #MK,RLDA          ;SET ONLY MARKER BIT!!
017154 004537 012552          JSR      R5,LDFUNC          ;ISSUE FUNCTION OF FOLLOWING WORD
017160 000104          GSTAT:INTEN          ;GET STATUS
017162 004537 013362          JSR      R5,WTCRDY          ;WAIT FOR CONTROLLER READY HIGH
017166          SETPRI  #PRI07
017166 012700 000340          MOV      #PRI07,R0
017172 104041          EMT      C$SPRI
017174 005737 002172          TST      INTFLG          ;INTERRUPT OCCUR
017200 001004          BNE      2$

017202          ERDF     30,EM11,ERRO
017202 104462          TRAP    T$ERCODE
017204 000036          .WORD   30
017206 004646          .WORD   EM11
017210 007556          .WORD   ERRO

017212          2$:      CKLOOP          ;CHECK IF FL:LOE IS SET
017212 104006          EMT      C$CLP1

017214 032737 074000 002152          BIT      #74000,E.CS
017222 001405          BEQ     1$
017224 012737 003510 012534          MOV      #OPIERR,RESTMS

```



```

017232 004537 012252          JSR      R5,CHERR          ;ISSUE FUNCTION OF FOLLOWING WORD
017236          CKLOOP          ;READ HEADER
017240 104006          EMT      CSCLP1          ;WAIT FOR CONTROLLER READY HIGH
017246 032737 002000 002152  BIT      #OPI,E.CS          ;CHECK IF /FL:LOE IS SET
017250 001004          BNE     3$              ;CHECK CONTROLLER FOR ERRORS
017250          ERROF     31,EM33,ERRO ;CHECK CONTROLLER FOR ERRORS
017252          TRAP     T$ERCODE
017254          .WORD   31
017256          .WORD   EM33
017260          .WORD   ERRO
3$:
ENDTST          ;****END OF TEST****
L10052:        EMT      C$ETST
.SBTTL **TEST 29** - READ HEADER FUNCTION
017262          BGNTST          ;****START OF TEST****
017262          STARS
:*****
:CHECK THAT READ HEADER WORKS, THAT WE CAN ISSUE
:IT GET READY BACK WITHOUT ANY ERRORS SETTING.
:*****
017262          STARS
:*****
017262 004537 012552          JSR      R5,LDFUNC          ;ISSUE FUNCTION OF FOLLOWING WORD
017266 000010          RDHDR          ;READ HEADER
017270 004537 013362          JSR      R5,WTCRDY          ;WAIT FOR CONTROLLER READY HIGH
017274          CKLOOP          ;CHECK IF /FL:LOE IS SET
017274 104006          EMT      CSCLP1
017276 004537 012252          JSR      R5,CHERR          ;CHECK CONTROLLER FOR ERRORS
017302          ENDTST          ;****END OF TEST****
017302          L10053:
017302          EMT      C$ETST

```

H06

DATA MACY11 30-1046 30-OCT-77 16:51 PAGE 86  
DATA.F11 05-OCT-77 10:41 \*\*TEST 29\*\* - READ HEADER FUNCTION

SEQ 0072

4769

\*\*TEST 30\*\* - READ HEADER FUNCTION INTERRUPT

.SBTTL \*\*TEST 30\*\* - READ HEADER FUNCTION INTEPRUPT

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

STARS

:\*\*\*\*\*

:CHECK THAT READ HEADER WILL GENERATE AN INTERRUPT  
:UPON COMPLETION WITHOUT ANY ERRORS SETTING

STARS

:\*\*\*\*\*

```

017304
017304
017304
017300 012700 000000
017304 104041
017310 104041
017312 005037 002172
017316 004537 012552
017322 000110
017324 004537 013362
017330 012700 000340
017334 104041
017336 005737 002172
017342 001004
017344 104462
017346 000043
017350 005313
017354 007556
017356 104006

```

```

SETPRI #PRIO0 ;PSW TO 0
MOV #PRIO0,R0
EMT CSSPRI
CLR INTFLG ;CLEAR INTERRUPT OCCURANCE
JSR R5,LOFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
RDHDR:INTEN ;READ HEADER, INTR. ENA
JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY HIGH
SETPRI #PRIO7
MOV #PRIO7,R0
EMT CSSPRI
TST INTFLG ;INTERRUPT HAPPEN
BNE 2$ ;YES-CONTINUE
ERRDF 35,EM37,ERR0
TRAP T$ERRCODE
.WORD 35
.WORD EM37
.WORD ERR0
2$: CKLOOP ;CHECK IF FL:LOE IS SET
EMT C$CLP1

```

```

48093 017356 004537 012252 JSR RS,CHERR ;CHECK CONTROLLER FOR ERRORS
48094 017362 ENDTST ;****END OF TEST****
48095 017362 L10054:
48096 017362 104001 EMT C$ETST
48097
48098 .SBTTL **TEST 31** - REPEATED RD HDRS YIELD SAME CYL AND HD
48099
48000 017364 BGNST ;****START OF TEST****
48001
48002
48003 017364 STARS
48004 :*****
48005 :CHECKT THAT READ HEADERS WILL RELIABLY READ THE SAME
48006 :CYLINDER AND HEAD SELECT. WE WILL READ HEADERS VERIFYING
48007 :THAT WE ALWAYS READ THE SAME CYLINDER AND HEAD SELECT.
48008 STARS
48009 :*****
48100 017364 012701 000144 MOV #100,R1 ;SET UP TO DO 100 RD HDR'S
48101 017370 004537 012552 JSR RS,LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
48102 017374 000010 RDHDR ;READ HEADER
48103 017376 004537 013362 JSR RS,WTCRDY ;WAIT FOR CONTROLLER READY HIGH
48104 017402 99$: ESCAPE TST ;IF /FL:LOE SET LOOP, ELSE EXIT TST
48105 (3) 017402 104010 EMT C$ESCAPE
48106 (3) 017404 000122 .WORD L10055-.
48107
48108 017406 004537 012252 JSR RS,CHERR ;CHECK CONTROLLER FOR ERRORS
48109 017412 ESCAPE TST ;IF /FL:LOE SET LOOP, ELSE EXIT TST
48110 (3) 017412 104010 EMT C$ESCAPE
48111 (3) 017414 000112 .WORD L10055-.
48112
48113 017416 013737 002160 002224 MOV E.MP,GDDAT ;READ FIRST HEADER (ASSUME GOOD)
48114 017424 043737 002176 002224 BIC SECM$K,GDDAT ;MASK AWAY SECTOR BITS
48115 017432 BGNSEG ;****START OF SEGMENT****
48116 (3) 017432 104004 EMT C$BSEG
48117
48118 017434 004537 012552 2$: JSR RS,LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
48119 017440 000010 RDHDR
48120 017442 004537 013362 JSR RS,WTCRDY ;WAIT FOR CONTROLLER READY HIGH
48121 (3) 017446 97$: ESCAPE SEG ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
48122 (3) 017446 104010 EMT C$ESCAPE
48123 (3) 017450 000054 .WORD 10000$-.
48124
48125 017452 004537 012252 JSR RS,CHERR ;CHECK CONTROLLER FOR ERRORS
48126 017456 ESCAPE SEG ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
48127 (3) 017456 104010 EMT C$ESCAPE
48128 (3) 017460 000044 .WORD 10000$-.
48129
48130 017462 013737 002160 002226 MOV E.MP,BDDAT ;READ HEADER
48131 017470 043737 002176 002226 BIC SECM$K,BDDAT ;MASK AWAY SECTOR BITS
48132 017476 023737 002224 002226 CMP GDDAT,BDDAT ;IS HEADER CORRECT
48133 017504 001404 BEG 4$

```

```

4835
4836 017506 ERRDF 36,EM41,ERP4
      (3) 017506 104462 TRAP T$ERCODE
      (5) 017510 000044 .WORD 36
      (5) 017512 005353 .WORD EM41
      (5) 017514 007722 .WORD ERR4
4837
4838 017516 4$: CKLOOP ;CONSTANT CYL & HS
      (3) 017516 104006 EMT C$CLP1 ;CHECK IF /FL:LOE IS SET
4839
4840 017520 005301 DEC R1 ;PERFORM ALL READ HDR'S
4841 017522 001344 BNE 2$ ;IF NOT GO BACK AND DO ANOTHER
4842 017524 ENDOSEG ;****END OF SEGMENT****
      (3) 017524 10000$: EMT C$ESEG
      (3) 017524 104005 EMT C$ESEG
4843 017526 ENDTST ;****END OF TEST****
      (3) 017526 L1005$: EMT C$ETST
      (3) 017526 104001 EMT C$ETST
4844
4845
4846 .SBTTL **TEST 32** - CHECK OF HEADER CRC
4847
4848 017530 BGNTST ;****START OF TEST****
4849
4850 017530 STARS
      (2) ;*****
4851 ;CHECK THAT WE CAN READ THE HDCRC AFTER A
4852 ;READ HEADER AND THAT IT IS THE CORRECT CRC
4853 ;FOR THE HEADER.
4854 017530 STARS
      (2) ;*****
4855
4856
4857 017530 005037 017600 CLR 3$
4858 017534 004537 012552 JSR R5,LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
4859 017540 000010 RDHDR ;READ HEADER
4860 017542 004537 013362 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY HIGH
4861 017546 ESCAPE TST ;IF /FL:LOE SET LOOP, ELSE EXIT TST
      (3) 017546 104010 EMT C$ESCAPE
      (3) 017550 000114 .WORD L10056-.
4862
4863 017552 004537 012252 JSR R5,CHERR ;CHECK CONTROLLER FOR ERRORS
4864 017556 ESCAPE TST ;IF /FL:LOE SET LOOP, ELSE EXIT TST
      (3) 017556 104010 EMT C$ESCAPE
      (3) 017560 000104 .WORD L10056-.
4865
4866 017562 013737 002160 017576 MOV E.MP,2$ ;READ HEADER WORD
4867 017570 004537 013130 JSR R5,SIMBCC ;GO CALCULATE CRC
4868 017574 000020 16. ;16 BITS
4869 017576 000000 2$: .WORD 0 ;HEADER GOES HERE
4870 017600 000000 3$: .WORD 0 ;START WITH 0 CRC
4871 017602 013737 002206 017626 MOV CALBCC,5$
4872 017610 013737 002162 017624 MOV E.MP1,4$
4873 017616 004537 013130 JSR R5,SIMBCC ;GET SECOND HALF
4874 017622 000020 16.
4875 017624 000000 4$: .WORD 0

```

```

4876 017626 000000 5$: .WORD 0
4877 017630 013737 002206 002224 MOV CALBCC,GDDAT ;STORE CALCULATED CRC AS GOOD
4878 017636 013737 002164 002226 MOV E.MP2,BDDAT ;THIRD READ OF DB GETS CRC
4879 017644 023737 002224 002226 CMP GDDAT,BDDAT ;IS CRC CORRECT?
4880 017652 001404 BEQ 6$ ;IF SO CONTINUE
4881
4882 017654 ERRDF 37.EM42,ERR4
4883 (3) 017654 104462 TRAP T$EACODE
4884 (5) 017656 000045 .WORD 37
4885 (5) 017660 005444 .WORD EM42
4886 (5) 017662 007722 .WORD ERR4
4887 017664 6$:
4888 017664 ENDTST ;****END OF TEST****
4889 (3) 017664 L10056:
4890 (3) 017664 10400: EMT C$ETST

```

.SBTTL \*\*TEST 33\*\* - CHECK CONSECUTIVE HEADERS

```

4891 017666 BGNTST ;****START OF TEST****
4892
4893 017666 STARS
4894 (2) ;*****
4895 ;CHECK THAT THE HEADERS ARE CONSECUTIVE. WE WILL DO
4896 ;40 (FORTY) READ HEADERS AND STORE EACH. AFTER WE HAVE
4897 ;READ THE FORTIETH HEADER WE WILL VERIFY THAT
4898 ;THEY CAME IN SEQUENTIAL, THAT 0 FOLLOWS 39,
4899 ;THAT THERE WERE NO ERRORS.
4900 017666 STARS
4901 (2) ;*****

```

```

4902 017666 005037 002230 CLR FIRST ;CLEAR FIRST READ DONE FLAG
4903 017672 012703 002670 MOV #HDRBUF,R3 ;STORE HEADERS
4904 017676 012701 000050 MOV #40,R1 ;FOURTY HEADERS
4905 017702 012737 000210 002140 MOV #RDHOR!CRDY,B.CS
4906 017710 053737 002136 002140 BIS DRIVE,B.CS
4907 017716 013777 002140 162174 MOV B.CS,@RLCS
4908 017724 042777 000200 162166 2$: BIC #200,@RLCS
4909 017732 032777 000200 162160 1$: BIT #200,@RLCS ;DONE?
4910 017740 001774 BEQ 1$
4911 017742 017723 162152 MOV @RLCS,(R3)+
4912 017746 017723 162154 MOV @RLMP,(R3)+
4913 017752 017723 162150 MOV @RLMP,(R3)+
4914 017756 017723 162144 MOV @RLMP,(R3)+
4915 017762 005301 DEC R1 ;HAVE WE READ FOURTY HEADERS
4916 017764 001357 BNE 2$ ;GO BACK UNTIL FOURTY DONE
4917 017766 012703 002670 MOV #HDRBUF,R3 ;GET LIST OF HEADERS
4918 017772 012701 000050 MOV #40,R1 ;CHECK FOURTY OF THEM
4919 017776 011337 002152 MOV (R3),E.CS
4920 020002 005737 002152 TST E.CS
4921 020006 100016 BPL 99$
4922 020010 012737 003747 012534 MOV #RHDMES,RESTMS
4923 020016 005723 TST (R3)+

```

```

4924 020020 012337 002160      MOV      (R3)+,E.MP
4925 020024 012337 002162      MOV      (R3)+,E.MP1
4926 020030 012337 002164      MOV      (R3)+,E.MP2
4927 020034 004537 012252      JSR      R5,CHERR      ;CHECK CONTROLLER FOR ERRORS
4928 020040 000137 020202      JMP      7$
4929 020044 005723      99$:    TST      (R3)+
4930 020046 011337 002226      MOV      (R3),BDDAT    ;GET HEADER
4931 020052 005737 002230      TST      FIRST        ;IS THIS FIRST READ?
4932 020056 001007      BNE      4$           ;NO, BRANCH
4933 020060 012737 000001 002230      MOV      #1,FIRST     ;SET FIRST READ DONE FLAG
4934 020066 013737 002226 002224 3$:     MOV      BDDAT,GDDAT  ;SET UP NEXT READ EXPECTED
4935 020074 000435      BR       6$           ;GO SEE IF TEST IS DONE
4936 020076 005237 002224      4$:     INC      GDDAT        ;INCREMENT EXP'D HEADER
4937 020102 023737 002226 002224      CMP      BDDAT,GDDAT  ;IS NEW HEADER SEQUENTIAL?
4938 020110 001766      BEQ      3$           ;YES THEN BRANCH
4939 020112 033737 002176 002226      BIT      SECMSK,BDDAT ;IS NEW HEADER ZERO?
4940 020120 001015      BNE      5$           ;NO, THEN ERROR GO REPORT IT
4941 020122 013737 002224 002210      MOV      GDDAT,TEMP2  ;YES, CHECK IF LAST HEADER WAS
4942 020130 043737 002232 002210      BIC      CYLSK,TEMP2  ;MAX ADDRESS, IF SO BRANCH
4943 020136 023737 002234 002210      CMP      MXSEC1,TEMP2 ;STORE NEW DATA AS OLD
4944 020144 001750      BEQ      3$           ;AND PERFORM NEW RD HDR
4945 020146 043737 002176 002224      BIC      SECMSK,GDDAT ;EXPECTING ZERO SECTOR
4946 020154      5$:
4947 020154 005037 002230      CLR      FIRST        ;ERROR WILL MAKE US MISS
4948 020154 005037 002230      CLR      FIRST        ;NEXT SECTOR SEQUENTIALLY
4949 020154 005037 002230      CLR      FIRST        ;START OVER; CLEAR FIRST FLAG
4950 020160      ERROF   38,EM43,ERR2
4951 (3) 020160 104462      TRAP    TSEARCH
4952 (5) 020162 000046      .WORD   38
4953 (5) 020164 005502      .WORD   EM43
4954 (5) 020166 007606      .WORD   ERR2
4955 (3) 020170      6$:     CKLOOP  ;CHECK IF /FL:LOE IS SET
4956 (3) 020170 104006      EMT     CSCLP1
4957 020172 062703 000006      ADD     #6,R3
4958 020176 005301      DEC     R1
4959 020200 001321      BNE     99$          ;HAVE WE DONE THIS ENOUGH
4960 020202      7$:     ;NO, GO BACK DO IT AGAIN
4961 (3) 020202      ENDTST ;****END OF TEST****
4962 (3) 020202 104001      L10057: EMT     CSETST
4963 .SBTTL **TEST 34** - SEEK FUNCTION
4964 020204      BGNTST ;****START OF TEST****
4965 020204      STARS
4966 (2) ;*****
4967 ;CHECK THE SEEK FUNCTION RETURNS CONTROLLER READY
4968 ;WITH NO ERRORS. WE ISSUE A ONE TRACK IN WORD SEEK.
4969 020204 ;WE DO NOT CHECK THE RESULT FOR POSITION
4970 (2) STARS
4971 ;*****

```

```

4971
4972 020204 012777 000205 161712      MOV      #BIT7!MK!SIGN,ARLDA ;SET UP DA-DIFF=1,MARKER,TOWARDS
4973 020212 004537 012552                JSR      R5,LDFUNC          ;ISSUE FUNCTION OF FOLLOWING WORD
4974 020216 000006                SEEK                               ;SEEK
4975 020220 004537 013362                JSR      R5,WTCRDY         ;WAIT FOR CONTROLLER READY HIGH
4976 020224                WAITMS  #2.
(3) 020224 012700 000002                MOV      #2.,R0
(3) 020230 104026                EMT      C$WTM
4977 020232                2$:  CKLOOP                               ;CHECK IF /FL:LOE IS SET
(3) 020232 104006                EMT      C$CLP1
4978
4979
4980 020234 004537 012252                JSR      R5,CHERR          ;CHECK CONTROLLER FOR ERRORS
4981
4982 020240                ENDTST                               ;****END OF TEST****
(3) 020240                L10060:
(3) 020240 104001                EMT      C$ETST
4983
4984
4985                .SBTTL  **TEST 35** - CHECK DRIVE READY ON SEEK
4986
4987 020242                BGNTST                               ;****START OF TEST****
4988
4989
4990 020242                STARS
(2) (2) ;*****
4991                ;CHECK THE SEEK FUNCTION RETURNS DRIVE READY WITH
4992                ;NO ERRORS. WE ISSUE A ONE TRACK INWARD SEEK. WE DO
4993                ;NOT CHECK THE RESULT FOR POSITION
4994 020242                STARS
(2) (2) ;*****
4995
4996
4997
4998 020242 012777 000201 161654      MOV      #BIT7!MK,ARLDA ;SET DA, MARKER, DIFF=1.
4999 020250 004537 012552                JSR      R5,LDFUNC          ;ISSUE FUNCTION OF FOLLOWING WORD
5000 020254 000006                SEEK                               ;SEEK
5001 020256 004537 013362                JSR      R5,WTCRDY         ;WAIT FOR CONTROLLER READY HIGH
5002 020262                CKLOOP                               ;CHECK IF /FL:LOE IS SET
(3) 020262 104006                EMT      C$CLP1
5003
5004 020264 004537 012252                JSR      R5,CHERR          ;CHECK CONTROLLER FOR ERRORS
5005 020270                CKLOOP                               ;CHECK IF /FL:LOE IS SET
(3) 020270 104006                EMT      C$CLP1
5006
5007 020272 004537 013316                JSR      R5,WTCRDY         ;WAIT FOR DRIVE READY
5008 020276                CKLOOP                               ;CHECK IF /FL:LOE IS SET
(3) 020276 104006                EMT      C$CLP1
5009
5010 020300 004537 012252                JSR      R5,CHERR          ;CHECK CONTROLLER FOR ERRORS
5011
5012 020304                ENDTST                               ;****END OF TEST****
(3) 020304                L10061:
(3) 020304 104001                EMT      C$ETST
5013
5014

```



```

5015 .SBTTL **TEST 36** - SEEK FUNCTION INTERRUPT
5016
5017 020306 BGNTST ;****START OF TEST****
5018
5019
5020 020306 STARS
5021 (2) :*****
5022 :CHECK THAT CONTROLLER READY RESETTING WHEN THE SEEK IS
5023 :INITIATED CAUSES AN INTERRUPT BUT DRIVE READY WILL
5024 020306 :NOT. WE ALSO MONITOR FOR ANY ERROR BITS SETTING.
5025 (2) STARS
5026 :*****
5027
5028
5029 020306 005037 002172 CLR INTFLG
5030 020312 012700 000000 SETPRI #PRIO0 ;SET PSW TO 0
5031 (3) 020312 012777 000205 161576 MOV #PRIO0,R0
5032 (2) 020316 104041 EMT C$SPRI
5033 020320 004537 012552 MOV #BIT7:MK:SIGN, @RLDA ;SET UP RLDA
5034 020326 004537 013362 JSR R5, LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
5035 020332 000106 JSR R5, WTCRDY ;SEEK AND INTR. ENA.
5036 020334 004537 013362 JSR R5, WTCRDY ;WAIT FOR CONTROLLER READY HIGH
5037 020340 000240 NOP
5038 020342 005737 002172 1$: TST INTFLG ;DID INTERRUPT OCCUR
5039 020346 001004 BNE 2$ ;YES, GO CHECK DRDY
5040 (3) 020350 104462 ERRDF 40, EM47, ERRO
5041 (5) 020352 000050 TRAP T$ERCODE
5042 (5) 020354 005722 .WORD 40
5043 (5) 020356 007556 .WORD EM47
5044 020360 104006 2$: CKLOOP ERRO
5045 (3) 020360 104006 EMT C$CLP1 ;CHECK IF /FL:LOE IS SET
5046
5047
5048 020362 004537 012252 JSR R5, CHERR ;CHECK CONTROLLER FOR ERRORS
5049 020366 104006 CKLOOP ;CHECK IF /FL:LOE IS SET
5050 (3) 020366 104006 EMT C$CLP1
5051 020370 005037 002172 CLR INTFLG ;CLEAR INTERRUPT OCCURANCE
5052
5053
5054 020374 004537 013316 5$: JSR R5, WTCRDY ;WAIT FOR DRIVE READY
5055 020400 104006 CKLOOP ;CHECK IF FL:LOE IS SET
5056 (3) 020400 104006 EMT C$CLP1
5057
5058
5059 020402 012700 000340 SETPRI #PRIO7
5060 (3) 020402 104041 MOV #PRIO7, R0
5061 (3) 020406 005737 002172 EMT C$SPRI
5062 020410 001404 TST INTFLG ;DID DRIVE READY CAUSE INTERRUPT
5063 020414 001404 BEQ 6$ ;NO, CONTINUE
5064
5065
5066 020416 42, EM52, ERRO ERRDF
5067 (3) 020416 104462 TRAP T$ERCODE
5068 (5) 020420 000052 .WORD 42

```

```

(5) 020422 005753 .WORD EM52
(5) 020424 007556 .WORD ERRO
5056 020426 6$: CKLOOP ;CHECK IF /FL:LOE IS SET
(3) 020426 104006 EMT C$CLP1
5058 020430 ENDTST ;****END OF TEST****
(3) 020430 L10062: EMT C$SETST
(3) 020430 104001
5059
5060
5061 .SBTTL **TEST 37** - TEST DIFFERENCE WORD TRANSMISSION
5062 BGNTST ;****START OF TEST****
5063
5064
5065
5066
5067
5068 020432 STARS
(2) ;*****
5069 ;VERIFY THAT THE DIFFERENCE WORD LOADS AND IS
5070 ;TRANSMITTED CORRECTLY. WE WILL ISSUE SEEKS WITH THE
5071 ;DIFFERENCE WORD CONTAINING ALL OF THE BIT PATTERNS FLOATING 1,
5072 ;GROWING 1, GROWING 0 AND SHITING 0. THE SEEK WILL
5073 ;START FROM TRACK 0 EACH TIME AND WILL RETURN THERE
5074 ;EACH, THUS BOTH DIRECTIONS FOR PATTERNS WILL BE CHECKED.
5075 ;READ HEADERS ARE USED TO VERIFY THE SEEK CORRECTNESS.
5076 ;ERRORS ARE MONITORED AND REPORTED.
5077 020432 STARS
(2) ;*****
5078
5079
5080 020432 012703 002466 BGNSEG MOV #SKLST,R3 ;GET LIST OF DIFFERENCE WORDS
5081 020436 EMT C$BSEG ;****START OF SEGMENT****
(3) 020436 104004
5082 020440 1$: JSR R5,LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
5083 020440 004537 012552 RDHDR ;READ HEADER
5084 020444 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY HIGH
5085 020446 004537 013362 98$: CKLOOP ;CHECK IF /FL:LOE IS SET
(3) 020452 104006 EMT C$CLP1
5087
5088 020454 004537 012252 JSR R5,CHERP ;CHECK CONTROLLER FOR ERRORS
5089 020460 CKLOOP ;CHECK IF /FL:LOE IS SET
(3) 020460 104006 EMT C$CLP1
5090
5091 020462 013737 002160 002226 MOV E.MP,BDDAT ;READ HEADER
5092 020470 043737 002176 002226 BIC SECMSK,BDDAT ;CLEAR OUT SECTOR
5093 020476 001462 BEQ 99$ ;IF ON TRACK ZERO, H.S. ZERO. OK
5094
5095 ;NOT ON TRACK ZERO CALCULATE DIFFERENCE WORD AND PUT IT BACK
5096 ;ON ZERO.
5097
5098 020500 042737 000100 002226 BIC #RHHS,BDDAT ;CLEAR OUT HEAD SELECT
5099 020506 013777 002226 161410 MOV BDDAT,ALDA ;PUT CYLINDER AS DIFFERENCE WORD
5100 020514 052777 000001 161402 BIS #MK,ALDA ;SET MARKER BIT
5101 020522 004537 012552 JSR R5,LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD

```

S102	020526	000006			SEEK				:SEEK
S103	020530	004537	013362		JSR	R5,WTCRDY			:WAIT FOR CONTROLLER READY HIGH
S104	020534				CKLOOP				:CHECK IF /FL:LOE IS SET
(3)	020534	104006			EMT	C\$CLP1			
S105									
S106	020536	004537	012252		JSR	R5,CHERR			:CHECK CONTROLLER FOR ERRORS
S107	020542				CKLOOP				:CHECK IF /FL:LOE IS SET
(3)	020542	104006			EMT	C\$CLP1			
S108									
S109	020544	004537	013316		JSR	R5,WTCRDY			:WAIT FOR DRIVE READY
S110	020550			89%	CKLOOP				:CHECK IF /FL:LOE IS SET
(3)	020550	104006			EMT	C\$CLP1			
S111									
S112	020552	004537	012252		JSR	R5,CHERR			:CHECK CONTROLLER FOR ERRORS
S113	020556				CKLOOP				:CHECK IF /FL:LOE IS SET
(3)	020556	104006			EMT	C\$CLP1			
S114									
S115	020560	004537	012552		JSR	R5,LDFUNC			:ISSUE FUNCTION OF FOLLOWING WORD
S116	020564	000010			RCHDR				:READ HEADER
S117	020566	004537	013362		JSR	R5,WTCRDY			:WAIT FOR CONTROLLER READY HIGH
S118	020572			96%	CKLOOP				:CHECK IF /FL:LOE IS SET
(3)	020572	104006			EMT	C\$CLP1			
S119									
S120	020574	004537	012252		JSR	R5,CHERR			:CHECK CONTROLLER FOR ERRORS
S121	020600				CKLOOP				:CHECK IF /FL:LOE IS SET
(3)	020600	104006			EMT	C\$CLP1			
S122									
S123	020602	005037	002224		CLR	GDDAT			:CLEAR EXPECTED
S124	020606	013737	002226	002240	MOV	BDDAT,DWORD			:SAVE DIFFERENCE WORD
S125	020614	013737	002160	002226	MOV	E.MP,BDDAT			:READ HEADER
S126	020622	043737	002176	002226	BIC	SECMSK,BDDAT			:MASK OUT SECTOR BITS
S127	020630	001404			BEQ	5\$			:BRANCH IF ON ZERO TRACK
S128									
S129	020632				ERRDF	43,EMS4,ERR3			
S130	020632	104462			TRAP	T\$ERCODE			
S131	020634	000053			.WORD	43			
S132	020636	006023			.WORD	EMS4			
S133	020640	007650			.WORD	ERR3			
S134	020642			5%	CKLOOP				:CHECK IF /FL:LOE IS SET
(3)	020642	104006			EMT	C\$CLP1			
S135									
S136	020644	011377	161254		MOV	(R3),DRLDA			:GET DIFFERENCE WORD
S137	020650	052777	000005	161246	BIS	#SIGN,MK,DRLDA			:SET SIGN (TOWARDS SPINDLE) AND MARKER
S138	020656	004537	012552		JSR	R5,LDFUNC			:ISSUE FUNCTION OF FOLLOWING WORD
S139	020662	000006			SEEK				:SEEK
S140	020664	004537	013362		JSR	R5,WTCRDY			:WAIT FOR CONTROLLER READY HIGH
S141	020670				CKLOOP				:CHECK IF /FL:LOE IS SET
(3)	020670	104006			EMT	C\$CLP1			
S142									
S143	020672	004537	012252		JSR	R5,CHERR			:CHECK CONTROLLER FOR ERRORS
S144	020676				CKLOOP				:CHECK IF /FL:LOE IS SET
(3)	020676	104006			EMT	C\$CLP1			
S145									
S146	020700	004537	013316		JSR	R5,WTCRDY			:WAIT FOR DRIVE READY
S147	020704			87%	CKLOOP				:CHECK IF /FL:LOE IS SET
(3)	020704	104006			EMT	C\$CLP1			

```

S144
S145 020706 004537 012252 JSR R5,CHERR ;CHECK CONTROLLER FOR ERRORS
S146 020712 CKLOOP ;CHECK IF /FL:LOE IS SET
(3) 020712 104006 EMT C$CLP1
S147
S148 020714 004537 012552 JSR R5,LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
S149 020720 000010 RDHDR ;READ HEADER
S150
S151 020722 004537 013362 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY HIGH
S152 020726 CKLOOP ;CHECK IF /FL:LOE IS SET
(3) 020726 104006 EMT C$CLP1
S153
S154 020730 004537 012252 JSR R5,CHERR ;CHECK CONTROLLER FOR ERRORS
S155 020734 ESCAPE SEG ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
(3) 020734 104010 EMT C$ESCAPE
(3) 020736 000064 .WORD 10000$.
S156
S157 020740 011337 002224 MOV (R3),GDDAT ;GET EXPECTED CYLINDER
S158 020744 011337 002240 8$: MOV (R3),DWORD ;SET UP DIFFERENCE FOR SEEK
S159 020750 013737 002160 002226 MOV E.MP,BDDAT ;READ HEADER FROM RLMP
S160 020756 043737 002176 002226 BIC SECMSK,BDDAT ;CLEAR OUT SECTOR BITS
S161 020764 023737 002224 002226 CMP GDDAT,BDDAT ;DID SEEK GO TO THE RIGHT
S162 020772 001404 BEQ 9$ ;TRACK, IF SO, GO GET NEXT
S163
S164 020774 ERRDF 44,EMS4,ERR3
(3) 020774 104462 TRAP T$ERRCODE
(5) 020776 000054 .WORD 44
(5) 021000 006023 .WORD EMS4
(5) 021002 007650 .WORD ERR3
S165 021004 9$: CKLOOP ;CHECK IF /FL:LOE IS SET
(3) 021004 104006 EMT C$CLP1
S166
S167 021006 005723 TST (R3)+ ;BUMP PATTERN
S168 021010 020327 002566 CMP R3,#SKEND ;DID WE DO ALL PATTERNS?
S169 021014 001402 BEQ 10$ ;YES, GO TO NEXT TEST
S170 021016 000137 020440 JMP 1$ ;NO, GO BACK WITH NEXT PATTERN
S171
S172 021022 10$: .
S173
S174 021022 ENDSEG ;****END OF SEGMENT****
(3) 021022 10000$:
(3) 021022 104005 EMT C$SESEG
S175 021024 ENDTST ;****END OF TEST****
(3) 021024 104001 EMT C$ETST
S176
S177
S178 .SBTTL **TEST 38** - VERIFY HEAD SELECT 0 VIA RD HDR
S179
S180 021026 BGNST ;****START OF TEST****
S181
S182 ;
S183
S184 021026 STARS
(2) ;*****
S185 ;CHECK THAT WE CAN SELECT HEAD SELECT ZERO. ISSUE

```

```

S186          :SEEK TO HEAD SELECT 0 AND VERIFY WITH READ HEADER.
S187 021026   STARS
(2)          ::*****
S188
S189 021026   012777 000001 161070 99$:  MOV  #MK,R LDA    ;SET MARKER IN RLDA
S190 021034   005037 002224          CLR  GDDAT    ;SET EXPECTED
S191          ;LOAD HS=0 INTO RLDA
S192 021040   2$:
S193 021040   004537 012552          JSR  R5, LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
S194 021044   000006          SEEK      ;SEEK
S195 021046   004537 013362          JSR  R5, WTCRDY ;WAIT FOR CONTROLLER READY HIGH
S196 021052   104006          CKLOOP  ;CHECK IF /FL:LOE IS SET
(3) 021052   EMT  CSCLP1
S197
S198 021054   004537 012252          JSR  R5, CHERR  ;CHECK CONTROLLER FOR ERRORS
S199 021060   104006          CKLOOP  ;CHECK IF /FL:LOE IS SET
(3) 021060   EMT  CSCLP1
S200
S201 021062   004537 013316          JSR  R5, WTDROY ;WAIT FOR DRIVE READY
S202 021066   104006          CKLOOP  ;CHECK IF /FL:LOE IS SET
(3) 021066   EMT  CSCLP1
S203
S204 021070   004537 012252          JSR  R5, CHERR  ;CHECK CONTROLLER FOR ERRORS
S205 021074   104006          CKLOOP  ;CHECK IF /FL:LOE IS SET
(3) 021074   EMT  CSCLP1
S206
S207 021076   004537 012552          JSR  R5, LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
S208 021102   000010          RDHDR    ;READ HEADER
S209 021104   004537 013362          JSR  R5, WTCRDY ;WAIT FOR CONTROLLER READY HIGH
S210 021110   104006          CKLOOP  ;CHECK IF /FL:LOE IS SET
(3) 021110   EMT  CSCLP1
S211
S212 021112   004537 012252          JSR  R5, CHERR  ;CHECK CONTROLLER FOR ERRORS
S213 021116   104010          ESCAPE  TST      ;IF /FL:LOE SET LOOP, ELSE EXIT TST
(3) 021116   EMT  C$ESCAPE
(3) 021120   000036          .WORD  L10064-.
S214
S215 021122   013737 002160 002226   MOV  E, MP, BDDAT ;READ HEADER FOR HEAD SELECT
S216 021130   042737 177677 002226   BIC  #177677, BDDAT ;MASK ONLY HEAD SELECT
S217 021136   023737 002224 002226   CMP  GDDAT, BDDAT ;COMPARE HEAD SELECTS
S218 021144   001404          BEQ    $$      ;IF EQUAL CONTINUE
S219
S220 021146   104462          ERDF   45, EM55, ERR4
(3) 021146   TRAP  T$ERRCODE
(5) 021150   000055          .WORD  45
(5) 021152   006062          .WORD  EM55
(5) 021154   007722          .WORD  ERR4
S221
S222 021156   5$:
S223 021156   ENDTST
S224 021156   L10064: ;****END OF TEST****
(3) 021156   EMT  C$ETST
S225
S226
S227
S228
S229
S230
S231
S232
S233
S234
S235
S236
S237
S238
S239
S240
S241
S242
S243
S244
S245
S246
S247
S248
S249
S250
S251
S252
S253
S254
S255
S256
S257
S258
S259
S260
S261
S262
S263
S264
S265
S266
S267
S268
S269
S270
S271
S272
S273
S274
S275
S276
S277
S278
S279
S280
S281
S282
S283
S284
S285
S286
S287
S288
S289
S290
S291
S292
S293
S294
S295
S296
S297
S298
S299
S300
S301
S302
S303
S304
S305
S306
S307
S308
S309
S310
S311
S312
S313
S314
S315
S316
S317
S318
S319
S320
S321
S322
S323
S324
S325
S326
S327
S328
S329
S330
S331
S332
S333
S334
S335
S336
S337
S338
S339
S340
S341
S342
S343
S344
S345
S346
S347
S348
S349
S350
S351
S352
S353
S354
S355
S356
S357
S358
S359
S360
S361
S362
S363
S364
S365
S366
S367
S368
S369
S370
S371
S372
S373
S374
S375
S376
S377
S378
S379
S380
S381
S382
S383
S384
S385
S386
S387
S388
S389
S390
S391
S392
S393
S394
S395
S396
S397
S398
S399
S400

```

```

5228 021160          BGNTST                      :****START OF TEST****
5229
5230
5231 021160          STARS
(2)                :*****
5232                :CHECK THAT WE CAN SELECT HEAD SELECT ONE.  ISSUE
5233                :SEEK TO HEAD SELECT 1 AND VERIFY WITH READ HEADER.
5234 021160          STARS
(2)                :*****
5235
5236 021160 012777 000001 160736 99$:  MOV    #MK, @RLDA      ;SET MARKER IN RLDA
5237 021166 052777 000020 160730  BIS    #DAHS, @RLDA    ;LOAD HS=1 INTO RLDA
5238 021174 004537 012552          2$:  JSR    RS, LDFUNC     ;ISSUE FUNCTION OF FOLLOWING WORD
5239 021200 000006          SEEK
5240 021202 004537 013362          JSR    RS, WTCRDY    ;SEEK
5241 021206          CKLOOP      ;WAIT FOR CONTROLLER READY HIGH
5242 021206          EMT          ;CHECK IF /FL:LOE IS SET
(3)
5243
5244 021210 004537 012252          JSR    RS, CHERR     ;CHECK CONTROLLER FOR ERRORS
5245 021214          CKLOOP      ;CHECK IF /FL:LOE IS SET
(3)
5246 021214 104006          EMT          CSCLP1
5247 021216 004537 013316          JSR    RS, WTDYD     ;WAIT FOR DRIVE CLEAR
5248 021222          CKLOOP      ;CHECK IF /FL:LOE IS SET
(3)
5249 021222 104006          EMT          CSCLP1
5250 021224 004537 012252          JSR    RS, CHERR     ;CHECK CONTROLLER FOR ERRORS
5251 021230          CKLOOP      ;CHECK IF /FL:LOE IS SET
(3)
5252 021230 104006          EMT          CSCLP1
5253 021232 004537 012552          JSR    RS, LDFUNC     ;ISSUE FUNCTION OF FOLLOWING WORD
5254 021236 000010          RDHDR     ;READ HEADER
5255 021240 004537 013362          JSR    RS, WTCRDY    ;WAIT FOR CONTROLLER READY HIGH
5256 021244          CKLOOP      ;CHECK IF /FL:LOE IS SET
(3)
5257 021244 104006          EMT          CSCLP1
5258 021246 004537 012252          JSR    RS, CHERR     ;CHECK CONTROLLER FOR ERRORS
5259 021252          ESCAPE      TST
(3)
5260 021252 104010          EMT          ;IF /FL:LOE SET LOOP, ELSE EXIT TST
(3)
5261 021254 000044          .WORD    C$ESCAPE
5262
5263 021256 013737 002160 002226  MOV    E.MP, BDDAT   ;READ HEADER
5264 021264 042737 177677 002226  BIC    #177677, BDDAT ;MASK FOR H.S.
5265 021272 012737 000100 002224  MOV    #RHHS, GDDAT  ;SET EXPECTED
5266 021300 023737 002224 002226  CMP    GDDAT, BDDAT  ;CORRECT HEAD
5267 021306 001404          BEQ     SS           ;YES, CONTINUE
5268
5269 021310          ERRDF      46, EM55, ERR4
(3)
5270 021310 104462          TRAP    T$ERRCODE
(5)
5271 021312 000056          .WORD    46
(5)
5272 021314 006062          .WORD    EM55
(5)
5273 021316 007722          .WORD    ERR4
5274 021320          SS:
5275
5276 021320          ENDTST                      :****END OF TEST****

```

```

021320
021320 104001
021322
021322
021322
021322 012777 000001 160574
021330 005037 002224
021334 004537 012552
021340 000006
021342 004537 013362
021346 104006
021350 004537 012252
021354 104006
021356 004537 013316
021362 104006
021364 004537 012252
021370 104006
021372 012777 000003 160524
021400 004537 012552
021404 000004
021406 004537 013362
021412 104006
021414 004537 012252
021420 104010
021422 000036
021424 013737 002160 002226
021432 042737 177677 002226
021440 023737 002224 002226
021446 001404
021450
021450 :04462

```

```

L10065:      EMT      CSETST

.SBTTL  **TEST 40** - VERIFY HEAD SELECT 0 VIA GET STATUS
BGNTST      ;****START OF TEST****
STARS
:*****
:CHECK THAT WE CAN READ BACK HEAD SELECT 0 WITH
:A GET STATUS FUNCTION.  SELECT H.S. 0 WITH A SEEK
:VERIFY WITH GET STATUS
STARS
:*****
MOV      #MK, @RLDA      ;SET MARKER IN RLDA
          ;LOAD HS=0 INTO RLDA
2$:      CLR      GDDAT      ;SET UP EXP'D
3$:      JSR      R5, LDFUNC   ;ISSUE FUNCTION OF FOLLOWING WORD
          ;SEEK
          JSR      R5, WTCRDY  ;WAIT FOR CONTROLLER READY HIGH
          CKLOOP   ;CHECK IF /FL:LOE IS SET
          EMT      C$CLP1

          JSR      R5, CHERR    ;CHECK CONTROLLER FOR ERRORS
          CKLOOP   ;CHECK IF /FL:LOE IS SET
          EMT      C$CLP1

          JSR      R5, WTDROY   ;WAIT FOR DRIVE READY
          CKLOOP   ;CHECK IF /FL:LOE IS SET
          EMT      C$CLP1

          JSR      R5, CHERR    ;CHECK CONTROLLER FOR ERRORS
          CKLOOP   ;CHECK IF /FL:LOE IS SET
          EMT      C$CLP1

MOV      #GSBIT!MK, @RLDA ;SET UP FOR GET STATUS IN DA
          JSR      R5, LDFUNC   ;ISSUE FUNCTION OF FOLLOWING WORD
          GSTAT     ;GET STATUS
          JSR      R5, WTCRDY   ;WAIT FOR CONTROLLER READY HIGH
          CKLOOP   ;CHECK IF /FL:LOE IS SET
          EMT      C$CLP1

          JSR      R5, CHERR    ;CHECK CONTROLLER FOR ERRORS
          ESCAPE   TST        ;IF /FL:LOE SET LOOP, ELSE EXIT TST
          EMT      C$ESCAPE
          .WORD    L10066-

MOV      E.MP, BDDAT      ;READ STATUS FOR HEAD SELECT BIT
BIC      #177677, BDDAT   ;LEAVE ONLY H.S. BIT
CMP      GDDAT, BDDAT     ;IS HEAD SELECT CORRECT?
BEQ      6$              ;YES, CONTINUE

ERRDF    47, EMS6, ERR4
TRAP    T$ERRCODE

```

021452 000057  
021454 006115  
021456 007732  
021460

.WORD 47  
.WORD EM56  
.WORD ERR4

65:

021460  
021460  
021460 104001

ENDTST :\*\*\*\*END OF TEST\*\*\*\*  
L10066: EMT CSETST

.SBTTL \*\*TEST 41\*\* - VERIFY HEAD SELECT 1 VIA GET STATUS

021462  
021462

BGNTST :\*\*\*\*START OF TEST\*\*\*\*

STARS  
:\*\*\*\*\*  
:CHECK THAT WE CAN READ BACK HEAD SELECT 1 WITH A GET  
:STATUS FUNCTION. SELECT H.S. 1 WITH A SEEK AND VERIFY WITH  
:GET STATUS  
STARS  
:\*\*\*\*\*

021462 012777 000001 160434  
021470 052777 000020 160426  
021476 012737 000100 002224  
021504 004537 012552  
021510 000006  
021512 004537 013362  
021516 104006

25: MOV #MK, @RLDA ;SET MARKER IN RLDA  
35: BIS #DAHS, @RLDA ;LOAD HS=1 INTO RLDA  
MOV #STHS, @DDAT ;SET UP EXP'D  
JSR R5, LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD  
SEEK ;SEEK  
JSR R5, WTCRDY ;WAIT FOR CONTROLLER READY HIGH  
CKLOOP ;CHECK IF /FL:LOE IS SET  
EMT C\$CLP1

021520 004537 012252  
021524 104006

JSR R5, CHERR ;CHECK CONTROLLER FOR ERRORS  
CKLOOP ;CHECK IF /FL:LOE IS SET  
EMT C\$CLP1

021526 004537 013316  
021532 104006

JSR R5, WTCRDY ;WAIT FOR DRIVE READY  
CKLOOP ;CHECK IF /FL:LOE IS SET  
EMT C\$CLP1

021534 004537 012252  
021540 104006

JSR R5, CHERR ;CHECK CONTROLLER FOR ERRORS  
CKLOOP ;CHECK IF /FL:LOE IS SET  
EMT C\$CLP1

021542 012777 000003 160354  
021550 004537 012552  
021554 000004  
021556 004537 013362  
021562 104010  
021564 000046

MOV #GSBIT!MK, @RLDA ;SET UP FOR GET STATUS IN DA  
JSR R5, LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD  
GSTAT ;GET STATUS  
JSR R5, WTCRDY ;WAIT FOR CONTROLLER READY HIGH  
ESCAPE TST ;IF /FL:LOE SET LOOP, ELSE EXIT TST  
EMT C\$ESCAPE  
.WORD L10067-

021566 004537 012252  
021572 104010  
021574 000036

JSR R5, CHERR ;CHECK CONTROLLER FOR ERRORS  
ESCAPE TST ;IF /FL:LOE SET LOOP, ELSE EXIT TST  
EMT C\$ESCAPE  
.WORD L10067-



```

5356 021576 013737 002160 002226      MOV      E.MP,BDDAT      ;READ STATUS FOR HEAD SELECT BIT
5357 021604 042737 177677 002226      BIC      #177677,BDDAT   ;LEAVE ONLY H.S. BIT
5358 021612 023737 002224 002226      CMP      GDDAT,BDDAT    ;IS HEAD SELECT CORRECT?
5359 021620 001404                      BEQ      6$              ;YES, CONTINUE
5360
5361 021622                      ERROF   48,EM56,ERR4
5362 (3) 021622 104462      TRAP   T$ERRCODE
5363 (5) 021624 000060      .WORD 48
5364 (5) 021626 006115      .WORD EM56
5365 (5) 021630 007722      .WORD ERR4
5366 021632                      6$:
5367 021632                      ENDTST                      ;****END OF TEST****
5368 (3) 021632                      L10067:
5369 (3) 021632 104001      EMT      C$ETST
5370
5371 .SBTTL **TEST 42** - TEST TIME AT WHICH DIF WD GETS TRANSMITTED
5372 BGNTST                      ;****START OF TEST****
5373
5374 STARS
5375 :*****
5376 :VERIFY THAT THE DIFFERENCE WORD ON A SEEK IS
5377 :TRANSMITTED PRIOR TO CONTROLLER READY SETTING. THIS
5378 :IS DONE BY SETTING A KNOWN DIFFERENCE WORD IN
5379 :THE RLDA ISSUING A A SEEK, WAITING FOR CONTROLLER READY
5380 :(BUT NOT DRIVE READY), WRITING A DIFFERENT RLDA AND WAITING
5381 :FOR DRIVE READY. THE RESULTANT POSITION SHOULD BE THAT
5382 :OF THE FIRST RLDA ONLY.
5383 STARS
5384 :*****
5385 021634 004537 012552      JSR      R5,LDFUNC      ;ISSUE FUNCTION OF FOLLOWING WORD
5386 021640 000010                      RDHDR                      ;READ HEADER
5387 021642 004537 013362      JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY HIGH
5388 (3) 021646 104006      EMT      C$CLP1          ;CHECK IF /FL:LOE IS SET
5389
5390 021650 004537 012252      JSR      R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
5391 (3) 021654 104006      CKLOOP  EMT      C$CLP1  ;CHECK IF /FL:LOE IS SET
5392
5393 021656 013737 002160 002224      MOV      E.MP,GDDAT     ;READ HEADER
5394 021664 043737 002176 002224      BIC      SECMSK,GDDAT   ;CLEAR SECTOR BITS
5395 021672 012777 000001 160224      MOV      #MK,@RLDA     ;SET MARKER IN RLDA
5396 021700 032737 000100 002224      BIT      #RHHS,GDDAT   ;TEST H.S.
5397 021706 001403                      BEQ      2$              ;IF ZERO, CONTINUE
5398 021710 052777 000020 160206      BIS      #DAHS,@RLDA   ;ONE SET SO WE WILL REMAIN THERE
5399 021716 013737 002224 002216      MOV      GDDAT,TMPO     ;STORE HEADER
5400 021724 042737 000100 002216      BIC      #RHHS,TMPO    ;CLEAR H.S. FROM STORED WORD
5401 021732 023737 002216 002544      CMP      TMPO,HALMAX   ;WHERE ARE WE?
5402 021740 101007                      BHI      3$              ;BRANCH IF ON INNER HALF

```

# K07

5402	021742	052777	000004	160154		BIS	#SIGN,RLDA	:ON OUTER HALF, SET SEEK TO GO IN
5403	021750	063737	002542	002224		ADD	QUAMAX,GDDAT	:SET UP EXPECTED HEADER
5404	021756	000403				BR	4\$	:CONTINUE
5405	021760	163737	002542	002224	3\$:	SUB	QUAMAX,GDDAT	:SET UP EXPECTED HEADER
5406	021766	053777	002542	160130	4\$:	BIS	QUAMAX,RLDA	:SET DIFFERENCE WORD IN RLDA
5407	021774	012737	000001	002220		MOV	#MK,TMP1	:SET UP ANOTHER "RLDA" FOR LOADING
5408	022002	032777	000020	160114		BIT	#DAHS,RLDA	:AFTER SEEK, TO CHANGE ONLY
5409	022010	001003				BNE	5\$	:HEAD
5410	022012	052737	000020	002220		BIS	#DAHS,TMP1	
5411	022020	004537	012552		5\$:	JSR	R5,LDFUNC	:ISSUE FUNCTION OF FOLLOWING WORD
5412	022024	000006				SEEK		:SEEK
5413	022026	004537	013362			JSR	R5,WTCRDY	:WAIT FOR CONTROLLER READY HIGH
5414	022032					CKLOOP		:CHECK IF /FL:LOE IS SET
(3)	022032	104006				EMT	C\$CLP1	
5415								
5416								
5417	022034	004537	012252			JSR	R5,CHERR	:CHECK CONTROLLER FOR ERRORS
5418	022040					CKLOOP		:CHECK IF /FL:LOE IS SET
(3)	022040	104006				EMT	L\$CLP1	
5419								
5420	022042	013777	002220	160054		MOV	TMP1,RLDA	:SEND IN NEW DIFFERENCE WORD
5421	022050	004537	013362			JSR	R5,WTCRDY	:WAIT FOR CONTROLLER READY HIGH
5422	022054					CKLOOP		:CHECK IF /FL:LOE IS SET
(3)	022054	104006				EMT	C\$CLP1	
5423								
5424	022056	004537	012252			JSR	R5,CHERR	:CHECK CONTROLLER FOR ERRORS
5425	022062					CKLOOP		:CHECK IF /FL:LOE IS SET
(3)	022062	104006				EMT	C\$CLP1	
5426								
5427	022064	004537	013316			JSR	R5,WTCRDY	:WAIT FOR DRIVE READY
5428	022070				8\$:	CKLOOP		:CHECK IF /FL:LOE IS SET
(3)	022070	104006				EMT	C\$CLP1	
5429								
5430								
5431	022072	004537	012252			JSR	R5,CHERR	:CHECK CONTROLLER FOR ERRORS
5432	022076					CKLOOP		:CHECK IF /FL:LOE IS SET
(3)	022076	104006				EMT	C\$CLP1	
5433								
5434	022100	004537	012552			JSR	R5,LDFUNC	:ISSUE FUNCTION OF FOLLOWING WORD
5435	022104	000010				RDHDR		:READ HEADER
5436	022106	004537	013362			JSR	R5,WTCRDY	:WAIT FOR CONTROLLER READY HIGH
5437	022112					CKLOOP		:CHECK IF /FL:LOE IS SET
(3)	022112	104006				EMT	C\$CLP1	
5438								
5439	022114	004537	012252			JSR	R5,CHERR	:CHECK CONTROLLER FOR ERRORS
5440	022120					ESCAPE	T\$T	:IF /FL:LOE SET LOOP, ELSE EXIT T\$T
(3)	022120	104010				EMT	C\$ESCAPE	
(3)	022122	000036				.WORD	L10070-	
5441								
5442	022124	013737	002160	002226		MOV	E.MP,BDDAT	:READ HEADER
5443	022132	043737	002176	002226		BIC	SECMSK,BDDAT	:CLEAR SECTOR ADDRESS
5444	022140	023737	002224	002226		CMP	GDDAT,BDDAT	:IS HEADER CORRECT?
5445	022146	001404				BEQ	10\$	:IF SO BRANCH
5446								
5447	022150					ERRDF	50,EM57,ERR4	
(3)	022150	104462				TRAP	T\$ERRCODE	

(S) 022152 000062  
 (S) 022154 006154  
 (S) 022156 007722  
 (S) 022160  
 (S) 022160  
 (S) 022160 104001  
 (S) 022162  
 (S) 022162  
 (S) 022162  
 (S) 022162  
 (S) 022162 012703 002466  
 (S) 022166  
 (S) 022166 104004  
 (S) 022170  
 (S) 022170 004537 012552  
 (S) 022174 000010  
 (S) 022176 004537 013362  
 (S) 022202  
 (S) 022202 104006  
 (S) 022204 004537 012252  
 (S) 022210  
 (S) 022210 104006  
 (S) 022212 013737 002160 002226  
 (S) 022220 043737 002176 002226  
 (S) 022226 001461  
 (S) 022230 042737 000100 002226  
 (S) 022236 013777 002226 157660  
 (S) 022244 052777 000001 157652  
 (S) 022252 004537 012552  
 (S) 022256 000006  
 (S) 022260 004537 013362  
 (S) 022264  
 (S) 022264 104006  
 (S) 022266 004537 012252  
 (S) 022272  
 (S) 022272 104006

.WORD 50  
 .WORD EM57  
 .WORD ERR4  
 10\$:  
 ENDTST ;\*\*\*\*END OF TEST\*\*\*\*  
 L10070: EMT C\$ETST  
 .SBTTL \*\*TEST 43\*\* - EXTENSIVE CHECK OF HEADER CRC  
 BGNST ;\*\*\*\*START OF TEST\*\*\*\*  
 STARS  
 ;\*\*\*\*\*  
 ;MORE EXTENSIVE CHECK OF HEADER CRC. WE WILL SEEK  
 ;AND READ HEADERS VERIFYING HDR CRC ACROSS THE  
 ;PLATTER USING THE GROWING 0, GROWING 1, SHIFTING 0 AND  
 ;GROWING 0 PATTERNS FOR TRACK ADDRESSES.  
 STARS  
 ;\*\*\*\*\*  
 BGNSEG MOV #SKLST,R3 ;GET LIST OF DIFFERENCE WORDS  
 ;\*\*\*\*START OF SEGMENT\*\*\*\*  
 EMT C\$BSEG  
 1\$: JSR R5,LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD  
 RDHDR ;READ HEADER  
 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY HIGH  
 98\$: CKLOOP ;CHECK IF /FL:LOE IS SET  
 EMT C\$CLP1  
 JSR R5,CHERR ;CHECK CONTROLLER FOR ERRORS  
 CKLOOP ;CHECK IF /FL:LOE IS SET  
 EMT C\$CLP1  
 MOV E.MP,BDDAT ;READ HEADER  
 BIC SECMSK,BDDAT ;CLEAR OUT SECTOR  
 BEQ 5\$ ;IF ON TRACK ZERO, H.S. ZERO, OK  
 ;NOT ON TRACK ZERO CALCULATE DIFFERENCE WORD AND PUT IT BACK  
 ;ON ZERO.  
 BIC #RHHS,BDDAT ;CLEAR OUT HEAD SELECT  
 MOV BDDAT,ARLDA ;PUT CYLINDER AS DIFFERENCE WORD  
 BIS #MK,ARLDA ;SET MARKER BIT  
 JSR R5,LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD  
 SEEK ;SEEK  
 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY HIGH  
 CKLOOP ;CHECK IF /FL:LOE IS SET  
 EMT C\$CLP1  
 JSR R5,CHERR ;CHECK CONTROLLER FOR ERRORS  
 CKLOOP ;CHECK IF /FL:LOE IS SET  
 EMT C\$CLP1

# M07

007ERR MACY11 30.1046) 30-OCT-77 16:51 PAGE 88-16  
 22RLAA.P11 05-OCT-77 10:41

\*\*TEST 43\*\* - EXTENSIVE CHECK OF HEADER CRC

SEQ 0090

5492	022274	004537	013316		JSR	R5,WTDROY	;WAIT FOR DRIVE READY
5493	022300			89\$:	CKLOOP		;CHECK IF /FL:LOE IS SET
(3)	022300	104006			EMT	C\$CLP1	
5495							
5496	022302	004537	012252		JSR	R5,CHERR	;CHECK CONTROLLER FOR ERRORS
5497	022306				CKLOOP		;CHECK IF /FL:LOE IS SET
(3)	022306	104006			EMT	C\$CLP1	
5498							
5499							
5500	022310	004537	012552		JSR	R5,LDFUNC	;ISSUE FUNCTION OF FOLLOWING WORD
5501	022314	000010			RDHDR		;READ HEADER
5502	022316	004537	013362		JSR	R5,WTCRDY	;WAIT FOR CONTROLLER READY HIGH
5503	022322			96\$:	CKLOOP		;CHECK IF /FL:LOE IS SET
(3)	022322	104006			EMT	C\$CLP1	
5504							
5505	022324	004537	012252		JSR	R5,CHERR	;CHECK CONTROLLER FOR ERRORS
5506	022330				CKLOOP		;CHECK IF /FL:LOE IS SET
(3)	022330	104006			EMT	C\$CLP1	
5507							
5508	022332	005037	002224		CLR	GDDAT	;CLEAR EXPECTED
5509	022336	013737	002226	002240	MOV	BDDAT,DWORD	;SAVE DIFFERENCE WORD
5510	022344	013737	002160	002226	MOV	E.MP,BDDAT	;READ HEADER
5511	022352	043737	002176	002226	BIC	SECMSK,BDDAT	;MASK OUT SECTOR BITS
5512	022360	001404			BEQ	\$S	;BRANCH IF ON ZERO TRACK
5513							
5514	022362				ERRDF	\$1,EMS4,ERR3	
(3)	022362	104462			TRAP	T\$ERCODE	
(5)	022364	000063			.WORD	\$1	
(5)	022366	006023			.WORD	EMS4	
(5)	022370	007650			.WORD	ERR3	
5515	022372				CKLOOP		;CHECK IF /FL:LOE IS SET
(3)	022372	104006		5\$:	EMT	C\$CLP1	
5516							
5517	022374	011377	157524		MOV	(R3),@RLDA	;GET DIFFERENCE WORD
5518	022400	052777	000005	157516	BIS	#SIGN!MK,@RLDA	;SET SIGN (TOWARDS SPINDLE) AND MARKER
5519	022406	004537	012552		JSR	R5,LDFUNC	;ISSUE FUNCTION OF FOLLOWING WORD
5520	022412	000006			SEEK		;SEEK
5521	022414	004537	013362		JSR	R5,WTCRDY	;WAIT FOR CONTROLLER READY HIGH
5522	022420				CKLOOP		;CHECK IF /FL:LOE IS SET
(3)	022420	104006			EMT	C\$CLP1	
5523							
5524	022422	004537	012252		JSR	R5,CHERR	;CHECK CONTROLLER FOR ERRORS
5525	022426				CKLOOP		;CHECK IF /FL:LOE IS SET
(3)	022426	104006			EMT	C\$CLP1	
5526							
5527	022430	004537	013316		JSR	R5,WTDROY	;WAIT FOR DRIVE READY
5528	022434				CKLOOP		;CHECK IF /FL:LOE IS SET
(3)	022434	104006			EMT	C\$CLP1	
5529							
5530							
5531	022436	004537	012252		JSR	R5,CHERR	;CHECK CONTROLLER FOR ERRORS
5532	022442				CKLOOP		;CHECK IF /FL:LOE IS SET
(3)	022442	104006			EMT	C\$CLP1	
5533							
5534	022444	004537	012552		JSR	R5,LDFUNC	;ISSUE FUNCTION OF FOLLOWING WORD



```

5579 022654          ENDSEG          ;****END OF SEGMENT****
      (3) 022654          10000$:
      (3) 022654 104005  EMT      C$ESEG
5580 022656          ENDTST          ;****END OF TEST****
      (3) 022656          L10071:
      (3) 022656 104001  EMT      C$ETST
5581
5582
5583 .SBTTL **TEST 44** - VERIFY GET STATUS WHILE DRDY IS LOW
5584
5585 022660          BGNTST          ;****START OF TEST****
5586
5587 022660          STARS
      (2) ;*****
5589 ;VERIFY THAT WE CAN ISSUE GET STATUS AND RECIEVE
5589 ;THE STATUS WORD WHILE THE DRIVE IS IN NOTION SEEKING
5590 022660          STARS
      (2) ;*****
5591
5592
5593 022660          IS:
5594 022660 004537 012552  JSR      R5,LDFUNC      ;ISSUE FUNCTION OF FOLLOWING WORD
5595 022664 000010  RDHDR          ;READ HEADER
5596 022666 004537 013362  JSR      R5,WTCRDY     ;WAIT FOR CONTROLLER READY HIGH
5597 022672          CKLOOP          ;CHECK IF /FL:LOE IS SET
      (3) 022672 104006  EMT      C$CLP1
5598
5599 022674 004537 012252  JSR      R5,CHERR      ;CHECK CONTROLLER FOR ERRORS
5600 022700          CKLOOP          ;CHECK IF /FL:LOE IS SET
      (3) 022700 104006  EMT      C$CLP1
5601
5602 022702 013737 002160 002226  MOV      E,MP,BDDAT    ;READ HEADER
5603 022710 043737 002176 002226  BIC      SECMSK,BDDAT  ;CLEAR OUT SECTOR
5604 022716 001461          BEQ      SS          ;IF ON TRACK ZERO, H.S. ZERO, OK
5605
5606 ;NOT ON TRACK ZERO CALCULATE DIFFERENCE WORD AND PUT IT BACK
5607 ;ON ZERO.
5608
5609 022720 042737 000100 002226  BIC      #RHHS,BDDAT   ;CLEAR OUT HEAD SELECT
5610 022726 013777 002226 157170  MOV      BDDAT,ARLDA   ;PUT CYLINDER AS DIFFERENCE WORD
5611 022734 052777 000001 157162  BIS      #MK,ARLDA    ;SET MARKER BIT
5612 022742 004537 012552  JSR      R5,LDFUNC     ;ISSUE FUNCTION OF FOLLOWING WORD
5613 022746 000006          SEEK          ;SEEK
5614 022750 004537 013362  JSR      R5,WTCRDY     ;WAIT FOR CONTROLLER READY HIGH
5615 022754          CKLOOP          ;CHECK IF /FL:LOE IS SET
      (3) 022754 104006  EMT      C$CLP1
5616
5617 022756 004537 012252  JSR      R5,CHERR      ;CHECK CONTROLLER FOR ERRORS
5618 022762          CKLOOP          ;CHECK IF /FL:LOE IS SET
      (3) 022762 104006  EMT      C$CLP1
5619
5620 022764 004537 013316  JSR      R5,WTCRDY     ;WAIT FOR DRIVE READY
5621 022770          CKLOOP          ;CHECK IF /FL:LOE IS SET
      (3) 022770 104006  EMT      C$CLP1
5622
5623 022772 004537 012252  JSR      R5,CHERR      ;CHECK CONTROLLER FOR ERRORS

```

Address	Op Code	Operand 1	Operand 2	Operand 3	Operand 4	Comment
5624	CKLOOP					;CHECK IF /FL:LOE IS SET
(3)	EMT	104006				C\$CLP1
5625						
5626						
5627	JSR	023000	004537	012552		;ISSUE FUNCTION OF FOLLOWING WORD
5628	RDHDR	023004	000010			;READ HEADER
5629	JSR	023006	004537	013362		;WAIT FOR CONTROLLER READY HIGH
5630	CKLOOP					;CHECK IF /FL:LOE IS SET
(3)	EMT	104006				C\$CLP1
5631						
5632	JSR	023014	004537	012252		;CHECK CONTROLLER FOR ERRORS
5633	CKLOOP					;CHECK IF /FL:LOE IS SET
(3)	EMT	104006				C\$CLP1
5634						
5635	CLR	023022	005037	002224		;CLEAR EXPECTED
5636	MOV	023026	013737	002226	002240	;SAVE DIFFERENCE WORD
5637	MOV	023034	013737	002160	002226	;READ HEADER
5638	BIC	023042	043737	002176	002226	;MASK OUT SECTOR BITS
5639	BEQ	023050	001404			;BRANCH IF ON ZERO TRACK
5640						
5641	ERRDF	023052				S4, EMS4, ERR3
(3)	TRAP	023052	104462			T\$ERRCODE
(5)	.WORD	023054	000066			S4
(5)	.WORD	023056	006023			EMS4
(5)	.WORD	023060	007650			ERR3
5642	SS:	023062				;CHECK IF /FL:LOE IS SET
(3)	CKLOOP	023062	104006			C\$CLP1
(3)	EMT					
5643						
5644	MOV	023064	012777	077601	157032	;GET DIFFERENCE WORD
5645	BIS	023072	052777	000005	157024	;SET SIGN (TOWARDS SPINDLE) AND MARKER
5646	JSR	023100	004537	012552		;ISSUE FUNCTION OF FOLLOWING WORD
5647	SEEK	023104	000006			;SEEK
5648	JSR	023106	004537	013362		;WAIT FOR CONTROLLER READY HIGH
5649	CKLOOP					;CHECK IF /FL:LOE IS SET
(3)	EMT	023112	104006			C\$CLP1
5650						
5651						
5652	JSR	023114	004537	012252		;CHECK CONTROLLER FOR ERRORS
5653	CKLOOP					;CHECK IF /FL:LOE IS SET
(3)	EMT	023120	104006			C\$CLP1
5654	MOV	023122	012777	000003	156774	;ISSUE FUNCTION OF FOLLOWING WORD
5655	JSR	023130	004537	012552		;ISSUE FUNCTION OF FOLLOWING WORD
5656	GSTAT	023134	000004			
5657	JSR	023136	004537	013362		;WAIT FOR CONTROLLER READY HIGH
5658	CKLOOP					;CHECK IF /FL:LOE IS SET
(3)	EMT	023142	104006			C\$CLP1
5659	JSR	023144	004537	012252		;CHECK CONTROLLER FOR ERRORS
5660						
5661	ENDTST	023150				;****END OF TEST****
(3)	L10072:	023150				
(3)	EMT	023150	104001			C\$SETST
5662						
5663	BGNMOD	023152				HRDPRM
5664						
5665	BGNHRD	023152				WORD L10073-L\$HARD/2
(3)	.WORD	023152	000025			

5666					GPRML	CNTMSG,CNT,1,YES
5667	023154				.WORD	TSCODE
(4)	023154	004130			.WORD	CNTMSG
(4)	023156	023242			.WORD	1
(4)	023160	000001			.WORD	CSRMSG,CSR,0.160000,177776,YES
5668	023162				GPRMA	TSCODE
(4)	023162	000031			.WORD	CSRMSG
(4)	023164	023226			.WORD	TSLOLIM
(4)	023166	160000			.WORD	TSHILIM
(4)	023170	177776			.WORD	VECMG,VECT,0.0,776,YES
5669	023172				GPRMA	TSCODE
(4)	023172	001031			.WORD	VECMG
(4)	023174	023260			.WORD	TSLOLIM
(4)	023176	000000			.WORD	TSHILIM
(4)	023200	000776			.WORD	BRMSG,PRIOR,0.340,0.7,YES
5670	023202				GPRMD	TSCODE
(4)	023202	002032			.WORD	BRMSG
(4)	023204	023247			.WORD	340
(4)	023206	000340			.WORD	TSLOLIM
(4)	023210	000000			.WORD	TSHILIM
(4)	023212	000007			.WORD	DRMSG,DRBT,0.03400.0,7,YES
5671	023214				GPRMD	TSCODE
(4)	023214	003032			.WORD	DRMSG
(4)	023216	023267			.WORD	03400
(4)	023220	003400			.WORD	TSLOLIM
(4)	023222	000000			.WORD	TSHILIM
(4)	023224	000007			.WORD	
5672						
5673	023226				ENDHRD	
(2)					.EVEN	
5674	023226		L10073:			
5675	023226	052502	020123	042101	CSRMSG:	.ASCIZ /BUS ADDRESS/
	023234	051104	051505	000123		
5676	023242	046122	030461	000	CNTMSG:	.ASCIZ /RL11/
5677	023247	102	020122	042514	BRMSG:	.ASCIZ /BR LEVEL/
	023254	042526	000114			
5678	023260	042526	052103	051117	VECMG:	.ASCIZ /VECTOR/
	023266	000				
5679	023267	104	044522	042526	DRMSG:	.ASCIZ /DRIVE/
	023274	000				
5680		023276			.EVEN	
5681						
5682	023276				ENDMOD	
5683						
5684						
5685					BGNMOD	SFTPRM
5686	023276					
5687					BGNSFT	
5688	023276	000014			.WORD	L10074-LSSOFT/2
5689	023300				GPRML	OMSG,DLT,1,YES
(4)	023300	000130			.WORD	TSCODE
(4)	023302	023330			.WORD	OMSG
(4)	023304	000001			.WORD	1
5690	023306				XFERF	1\$



```

(5) 023306 006044 .WORD TSCODE
5691 023310 GPRMD EMSG,ELT.0,177777,0,177777,YES
(4) 023310 .WORD TSCODE
(4) 023312 001032 .WORD EMSG
(4) 023314 177777 .WORD 177777
(4) 023316 000000 .WORD T$LOLIM
(4) 023320 177777 .WORD T$HILIM
5692 023322 1S: GPRML SMSG,SIZE.1,YES
(4) 023322 002130 .WORD TSCODE
(4) 023324 023354 .WORD SMSG
(4) 023326 000001 .WORD 1
5693 023330 ENDSFT
(2) .EVEN
(3) 023330 L10074:
5694
5698
5699 023330 051104 050117 047440 DMSG: .ASCIZ /DROP ON ERROR LIMIT/
5700 023354 052501 047524 044523 SMSG: .ASCIZ /AUTOSIZE/
5701 023365 105 051122 051117 EMSG: .ASCIZ /ERROR LIMIT/
5702
5706 023402 .EVEN
5708
5709 023402 ENDMOD
5710
5711 023402 LASTAD
(2) .EVEN
5712

```

PDP-11 DIAGNOSTIC SUPERVISOR  
SERLAA.SJP 11-OCT-77 15:38

MACY11 30(1046) 30-OCT-77 16:51 PAGE 89  
\*\*TEST 44\*\* - VERIFY GET STATUS WHILE DRDY IS LOW

SEQ 0096

17013	054302
17009	071776
17011	000000
17012	072000
17014	000200
17015	

071776

```

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

```

ABOFLA 023654 G  
 ABOPAS 023566 G  
 ABO.FM 026530  
 ABOCOD 012162 G  
 AFREG 004301  
 AFTER 013062  
 ALLOC 044320  
 ARLBA 004236  
 ARLCS 004231  
 ARLDA 004244  
 ARLMP 004252  
 ASAAW 030440  
 ASAAZ 030454  
 ASAAZ 030462  
 ASAAZ 030476  
 ASABA 030506  
 BATEST 014146  
 BA16 = 000020  
 BA17 = 000040  
 BCCFBK 002204  
 BCSR 002130  
 BDDAT 002226  
 BEFORE 013030  
 BEGPAT 002256  
 BEREG 004260  
 BGN.SU = 023402  
 BINMSG 042630  
 BIT0 = 000001 G  
 BIT00 = 000001 G  
 BIT01 = 000002 G  
 BIT02 = 000004 G  
 BIT03 = 000010 G  
 BIT04 = 000020 G  
 BIT05 = 000040 G  
 BIT06 = 000100 G  
 BIT07 = 000200 G  
 BIT08 = 000400 G  
 BIT09 = 001000 G  
 BIT1 = 000002 G  
 BIT10 = 002000 G  
 BIT11 = 004000 G  
 BIT12 = 010000 G  
 BIT13 = 020000 G  
 BIT14 = 040000 G  
 BIT15 = 100000 G  
 BIT2 = 000004 G  
 BIT3 = 000010 G  
 BIT4 = 000020 G  
 BIT5 = 000040 G  
 BIT6 = 000100 G  
 BIT7 = 000200 G  
 BIT8 = 000400 G  
 BIT9 = 001000 G

BLO.MW 031320  
 BLOCK 046454  
 BPRIOR 002132  
 BRMSG 023247  
 BVEC 002134  
 BSAB 032722  
 BSAAF 032634  
 B.BA 002142  
 B.CS 002140  
 B.DA 002144  
 B.MP 002146  
 CALBCC 002206  
 CALLPC = 000022  
 CALLPS = 000024  
 CALLSP = 000026  
 CALLTC = 000030  
 CAL.CL 051036  
 CAL.TI 051074 G  
 CHERR 012252 G  
 CHKLRP 032736  
 CHKSTR 044662  
 CHKTTY 042750  
 CHK.FO 025146  
 CHK.MA 031076  
 CHK.PC 036164  
 CHK.SW 024670  
 CHRCNT 044202  
 CH.FLA 030600  
 CH.PAS 030622  
 CKERLT 012166  
 CLEAR. 032220  
 CLKACC 023564 G  
 CLKBFR 051040  
 CLKCNT 023562 G  
 CLKRES 052436 G  
 CLKSER 052736 G  
 CLKSON 023626 G  
 CLK.SE 030700 G  
 CLNCOD 012110 G  
 CLR.MA 031154  
 CNT = 000010  
 CNTMSG 023242  
 CNVT 047114  
 COMMTA 046734  
 COMP 003502  
 CONT 011462  
 CONTCL 052516 G  
 CCNTIN 011344  
 CROY = 000200  
 CRLF 043032  
 CRTIM 004322  
 CSEND 002666  
 CSPAT 002570

CSR = 000000  
 CSRMSG 023226  
 CSTEST 014026  
 CURR.T 023602 G  
 CYLSK 002232  
 CSAAD 036136  
 CSAAE 036150  
 CSAAK 036766  
 CSAL 037076  
 CSABRT = 000021  
 CSADR = 000020  
 CSAU = 000054  
 CSBRK = 000022  
 CSBSEG = 000004  
 CSBSUB = 000002  
 CSBUFF = 000030  
 CSCEFG = 000046  
 CSCLEA = 000012  
 CSCLP1 = 000006  
 CSCVEC = 000036  
 CSODLN = 000044  
 CSODDU = 000053  
 CSORPT = 000024  
 CSOU = 000055  
 CSEDIT = 000006  
 CSERDF = 000002  
 CSERHR = 000003  
 CSERSF = 000001  
 CSERSO = 000004  
 CSERSCA = 000010  
 CSSESEG = 000005  
 CSSESUB = 000003  
 CSETST = 000001  
 CSEXIT = 000032  
 CSGMAN = 000043  
 CSGPHR = 000042  
 CSGPRI = 000040  
 CSGTIM = 000052  
 CSINIT = 000011  
 CSINLP = 000020  
 CSKWOF = 000035  
 CSKWON = 000034  
 CSLOOP = 000100  
 CSMANI = 000051  
 CSMSG = 000023  
 CSPNTB = 000014  
 CSPNTF = 000017  
 CSPNTS = 000016  
 CSPNTX = 000015  
 CSPOIN = 000040  
 CSQIO = 000377  
 CSROBU = 000007  
 CSREFG = 000050

CSREQT = 000045  
 CSRESE = 000033  
 CSREVI = 000001  
 CSRPT = 000025  
 CSSEFG = 000047  
 CSSPRI = 000041  
 CSSVEC = 000037  
 CSTPRI = 000013  
 CSUNBU = 000031  
 CSWTM = 000026  
 CSWTU = 000027  
 DAHS = 000020  
 DATEST 014250  
 DCKMES 003463  
 DECMSG 042644  
 DEMES 003431  
 DERFLG 002150  
 DERR = 040000  
 DIAG.T 023662 G  
 DLT = 000000  
 DLTMS 003470  
 DMSG 023330  
 DPOVD 053446 G  
 DPMUL 053334 G  
 DRBT = 000006  
 DRDY = 000001  
 DRIVE 002136  
 DRMSG 023267  
 DROP 011114  
 DRPCOD 012156 G  
 DRST = 000010  
 DRTIM 004347  
 DSPCOD 011122 G  
 DSO = 000000  
 DS1 = 000400  
 DS2 = 001000  
 DS3 = 001400  
 DUNIT. 023572 G  
 DVC.FT 036736  
 DWORD 002240  
 DSAAG 037606  
 DSAAH 037624  
 DSAAI 042376  
 DSAAJ 042402  
 DSAAK 042420  
 DSAL 042436  
 DSAM 042446  
 EF.CON = 000036 G  
 EF.NEW = 000035 G  
 EF.PWR = 000034 G  
 EF.RES = 000037 G  
 EF.STA = 000040 G  
 EF01 = 000001 G

EF02 = 000002 G  
 EF03 = 000003 G  
 EF04 = 000004 G  
 EF05 = 000005 G  
 EF06 = 000006 G  
 EF07 = 000007 G  
 EF08 = 000010 G  
 EF09 = 000011 G  
 EF10 = 000012 G  
 EF11 = 000013 G  
 EF12 = 000014 G  
 EF13 = 000015 G  
 EF14 = 000016 G  
 EF15 = 000017 G  
 EF16 = 000020 G  
 EL = 000002  
 EMSG 023365  
 EMT.TR 023660 G  
 EM1 004375  
 EM101 007320  
 EM102 007365  
 EM11 004646  
 EM12 004707  
 EM14 004741  
 EM15 004767  
 EM16 005015  
 EM17 005043  
 EM2 004422  
 EM3 004447  
 EM30 005076  
 EM32 005135  
 EM33 005171  
 EM34 005236  
 EM37 005313  
 EM4 004474  
 EM41 005353  
 EM42 005444  
 EM43 005502  
 EM44 005601  
 EM45 005634  
 EM46 005667  
 EM47 005722  
 EM5 004521  
 EM52 005753  
 EM54 006023  
 EM55 006062  
 EM56 006115  
 EM57 006154  
 EM6 004572  
 EM61 006255  
 EM62 006336  
 EM63 006421  
 EM64 006502

FREE	044556	G\$OFFS=	000400	ISSUB	=	000041	L\$SPC	002030	G
FRMT1	010432	G\$OFFSI=	000376	ISTST	=	000041	L\$SPCP	002050	G
FRMT11	010724	G\$PRMA=	000001	J\$JMP	=	000167	L\$SPTP	002060	G
FRMT12	010765	G\$PRMD=	000002	KBPTR		023434	L\$STA	002066	G
FRMT13	011044	G\$PRML=	000000	KBUF		023436	L\$SW	011114	G
FRMT2	010472	G\$RADA=	000140	LDCSR		002174	L\$TJML	002022	G
FRMT2A	010511	G\$RADB=	000000	LOFUNC		012552	L\$TIMU	002020	G
FRMT2B	010524	G\$RADD=	000040	LF		003500	L\$TIMI	002016	G
FRMT3	010553	G\$RADF=	000200	LINE.F		023656	L\$TSTI	002074	G
FRMT4	010560	G\$RADL=	000120	LINE1		010102	L\$UNIT	002014	G
FRMT5	010616	G\$RADO=	000020	LINE2		010136	L.CLK.	030166	G
FRMT6	010667	G\$RADT=	000100	LINE3		010360	L10000	007572	G
FRMT99	010613	G\$XFER=	000004	LOAD.F		030620	L10001	007604	G
F\$AU =	000015	G\$YES =	000010	LOGMSG		043652	L10002	007646	G
F\$BGN =	000040	HALMAX	002544	LPBFR		023432	L10003	007720	G
F\$CLER =	000007	HCOREQ	030362	LPCNTR		023430	L10004	007766	G
F\$DU =	000016	HCORET	023614	LPT.AD		030220	L10005	010000	G
F\$END =	000041	HCRME	003450	LPT.RE		030214	L10006	010042	G
F\$HARD =	000004	HDRBUF	002670	LSI.RE		030210	L10007	010100	G
F\$HW =	000013	HDRLST	012772	LUP		050742	L10010	011112	G
F\$INIT =	000006	HERTZ	030202	LUP.AD		036166	L10011	011122	G
F\$JMP	000050	HNFMES	003456	L\$AU		012162	L10012	012106	G
F\$MOD	000000	HOLDSP=	000020	L\$AUT		002070	L10013	012154	G
F\$MSG	000011	HPTCOD	011076	L\$CCP		002044	L10014	012160	G
F\$PWR	000017	HRDPRM	023152	L\$CLEA		012110	L10015	012164	G
F\$RPT	000012	HSAB	047442	L\$DEPO		002011	L10016	013314	G
F\$SEG	000003	INIT	023604	L\$DEVP		002052	L10017	013532	G
F\$SOFT	000005	INITCO	011254	L\$DISP		011124	L10020	013626	G
F\$SRV	000010	INITIA	042660	L\$DR		002102	L10021	013722	G
F\$SUB	000002	INIT.M	031222	L\$DRCT		002062	L10022	014016	G
F\$SW	000014	INIT.R	023416	L\$DRS		002064	L10023	014136	G
F\$TEST =	000001	INPUTA	043606	L\$DRST		002102	L10024	014240	G
GARBAG	044204	INTEN =	000100	L\$DTP		002040	L10025	014326	G
GDDAT	002224	INTFLG	002172	L\$DU		012156	L10026	014452	G
GETCHR	042710	INTFOR	037104	L\$OUT		002072	L10027	014576	G
GETCMN	046274	INTSRV	013310	L\$DVTY		002104	L10030	014702	G
GETPAR	037766	INVAL.	030330	L\$EF		002024	L10031	015002	G
GETSWI	045270	INVINT	036776	L\$EXP1		002032	L10032	015072	G
GET.TW	045040	INV.SW	024624	L\$EXP2		002034	L10033	015172	G
GLBDAT	002112	IN.SUF	032172	L\$EXP3		002036	L10034	015302	G
GLBEQA	002112	ISAU =	000041	L\$HARD		023154	L10035	015354	G
GLBERR	007556	ISCLN =	000041	L\$HPCP		002046	L10036	015412	G
GLBSUB	012166	ISDU =	000041	L\$HPTP		002056	L10037	015536	G
GLBTXT	003370	ISHR =	000041	L\$HW		011100	L10040	015676	G
GODRVR=	000202	ISINIT=	000041	L\$ICP		002042	L10041	016036	G
G\$BIT =	000002	ISMOD =	000041	L\$INIT		011254	L10042	016242	G
G\$TAT =	000004	ISMSG =	000041	L\$LADP		002076	L10043	016272	G
G\$TINT	004172	ISPR =	000041	L\$LAST		023402	L10044	016476	G
G\$TMES	004133	ISRPT =	000041	L\$MREV		002012	L10045	016562	G
G\$EXCP=	000400	ISSEG =	000041	L\$NAME		002000	L10046	016736	G
G\$HILI=	000002	ISSFT =	000041	L\$REPP		002054	L10047	016756	G
G\$LOLI=	000001	ISSRV =	000041	L\$REV		002010	L10050	017044	G
G\$NO =	000000			L\$SOFT		023300	L10051	017132	G

L10052	017260	NUM.LA	037442	RESTMS	012534	SVHD	002244	T\$TEST=	000054
L10053	017302	NUM.JN	024000	RE.SET	024772	SWCHAN	030422	T\$TSM=	177777
L10054	017362	NUNITS	032710	RHDINT	004007	SWITCH	047012	T\$TSTS=	000001
L10055	017526	NXM =	020000	RHCMS	003747	SW.PTA	030406	T\$SAU =	010015
L10056	017664	NXMMES	003436	RHHS =	000100	SYS.FT	036726	T\$SCLE=	010013
L10057	020202	NXT	011354	RLBA	002122	S\$LSYM=	010000	T\$SDU =	010014
L10060	020240	NXTFOR	047106	RLCS	002120	TEMP2	002210	T\$SHAR=	010073
L10061	020304	OCTMSG	042636	RLDA	002124	TEMP3	002212	T\$SHW =	010010
L10062	020430	OKHDR	013002	RLMP	002126	TEMP4	002214	T\$SINI=	010012
L10063	021024	OPI =	002000	RSTACK	053130	TERMI	051032	T\$SMG=	010007
L10064	021156	OPIERR	003510	RSX.FL	030614	TERMLI	046640	T\$SSEG=	010000
L10065	021320	OPIMES	003443	SEARCH	045006	TERMTA	042622	T\$SOF=	010074
L10066	021460	OSAPTS=	000000	SECMK	002176	TEST.M	030532	T\$SPV=	010016
L10067	021632	OSAU =	000001	SEEK =	000006	TIMFLG	023560	T\$SSW =	010011
L10070	022160	OSBGNR	000000	SEGSTA	023630	TIM.CO	023412	T\$STES=	010072
L10071	022656	OSBGNS=	000001	SEKINT	004101	TIM.OP	037246	T.CNTL	002252
L10072	023150	OSDU =	000001	SEKMES	004050	TMPFNC	002254	T.SIZE	011120
L10073	023226	OSGNSW=	000001	SET.MA	031010	TMPD	002216	T1	013440
L10074	023330	OSPOIN=	000001	SFTPRM	023276	TMP1	002220	T10	014600
MAY.IN	023406	PARSES	046346	SHIFT	053766	TMP2	002222	T11	014704
MAY.LO	051042	PAR.LA	042340	SIGN =	000004	TOO.MA	042602	T12	015004
MAY.US	023410	PFLG	002166	SIMBCC	013130	TRPFLG	002170	T13	015074
MAN.TI	025710	PRINTC	044160	SIZE =	000004	TRPHAN	013302	T14	015174
MAP16	053704	PRINTF	047462	SIZE.C	052644	TST.AB	033046	T15	015304
MASK.B	032734	PRIOR =	000004	SIZE.M	052562	TST.TO	024652	T16	015356
MASK.W	032732	PRIO0 =	000000	SIZ.TR	052722	TYPEC	043176	T17	015414
MAXCYL	002242	PRIO1 =	000040	SKEND	002566	TYPEPC	037072	T18	015540
MAXSEC	002236	PRIO2 =	000100	SKLST	002466	TYPFLA	046514	T19	015700
MCHEDR	002000	PRIO3 =	000140	SMSG	023354	TYPLIN	043074	T2	013534
MEM.SI	030230	PRIO4 =	000200	SPEC.U	030520	TYPNUM	042462	T20	016040
MERLMT	011116	PRIO5 =	000240	SPTCOO	011112	TYPSTR	043114	T21	016244
MIN.IN	023402	PRIO6 =	000300	SPV.SE	025222	TYP.ER	036756	T22	016274
MIN.LS	023404	PRIO7 =	000340	START	011326	TY.UNI	032212	T23	016500
MK =	000001	PRNTST	044050	STARTC	052512	T\$ARGC=	000001	T24	016564
MOOR	053246	PRO.CM	030572	STHS =	000100	T\$CODE=	002130	T25	016730
MSCRLF	003475	PTAB.S	023612	STRCHR	043540	T\$ERCO=	000062	T26	016760
MUL	053202	PUTCHR	042664	STREQ.	030342	T\$ERRN=	000066	T27	017046
MXSEC1	002234	PWRFLG	002112	STR.T	030576	T\$EXCP=	000000	T28	017134
NEWPRI	052726	PWR.FA	054140	ST.REQ	030516	T\$FLAG=	000040	T29	017262
NEXTAR	047036	PWR.FL	023414	ST.SET	025042	T\$HILI=	177777	T3	013630
NOORY	003406	PWR.MS	054266	SUNIT.	030604	T\$LOLI=	000000	T30	017304
NOOP0 =	000000	PWR.SA	054262	SUPERV	026562	T\$LSYM=	010000	T31	017364
NOOP7 =	000016	PWR.UP	054264	SUPFLA	023570	T\$MCAL=	177777	T32	017530
NOPINT	003614	P.CLK.	030174	SUPV.T	023746	T\$NEST=	177777	T33	017666
NOPMES	003563	QUAMAX	002542	SUP.PR	024614	T\$NSKO=	000000	T34	020204
NOPWR	011276	ROHDR =	000010	SVCCNT=	177777	T\$NSKI=	000005	T35	020242
NORES	003370	READ =	000014	SVCGBL=	177777	T\$SAVL=	177777	T36	020306
NC.CLK	025740	READ.P	051044	SVCHAN	033114	T\$SEGL=	177777	T37	020432
NC.FLA	046632	REGBAC	053670	SVCINS=	000000	T\$SEKO=	010000	T39	021026
NC.LPT	044150	REGSAV	053654	SVCSTK=	177777	T\$SUBN=	000000	T39	021160
NC.PTA	030430	REGN.P	030602	SVCSUB=	177777	T\$TAGL=	177777	T4	013724
NR =	000000	REGN.T	030574	SVCTAG=	000000	T\$TAGN=	010075	T40	021322
NUMBIN	037274	REST	011406	SVCTST=	177777	T\$TEMP=	000000	T41	021462

T42	021634	G	USER.T	023610	G	WRCHK	=	000002	XEQ.PR	025750	X1X1	=	072000	
T43	022162	G	UUT	002114		WRITE	=	000012	XEQ.TE	032472	\$BREG	=	030676	
T44	022660	G	VALID.	024050		WTCROY	=	013362	XPOLY	002200	\$ENDAD	=	052766	G
T5	014020	G	VAL.LA	024600		WTDROY	=	013316	XTIME	051522	\$SAV2	=	054032	G
T6	014140	G	VAL.SW	030636		XEQDIA	=	053014	XTIMEN	052346	\$SAV3	=	054046	G
T7	014242	G	VECMG	023260		XEQSUB	=	053002	XTIMST	051544	\$SAV4	=	054064	G
T8	014330	G	VECT	=	000002	XEQ.CL	=	032652	XXDP.D	030370	\$SAV5	=	054104	G
T9	014454	G	WCKINT	003706		XEQ.CM	=	030160	X\$ALWA	=	000000		=	072000
UNITST	002116		WCKMES	003646		XEQ.IN	=	032634	X\$FALS	=	000040			
UNI.MA	030522		WHY	002250		XEQ.LA	=	026516	X\$OFFS	=	000400			
USER.P	023606	G	WIDTH	037642		XEQ.OP	=	032426	X\$TRUE	=	000020			

. ABS. 072000 300

ERRORS DETECTED: 0

DSKZ:DZRLAA, DSKZ:DZRLAA=DZRLAA.SML, DZRLAA.P11, DZRLAA.SUP  
RUN-TIME: 36 40 .9 SECONDS  
RUN-TIME RATIO: 342 77=4.3  
CORE USED: 18K (35 PAGES)

K08