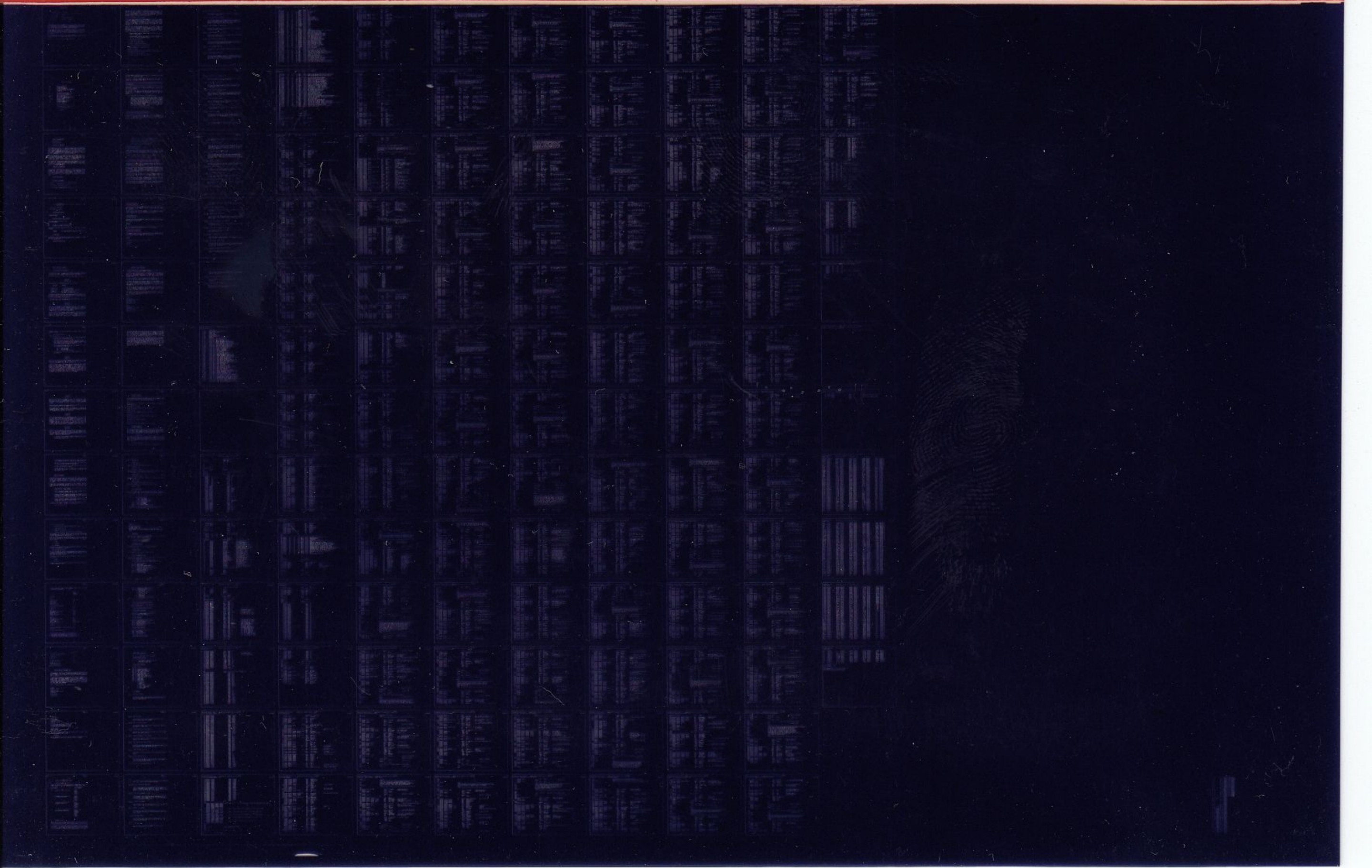


RL11,RLV11

CTLR 2
CZRLHA0

AH-F114A-MC
COPYRIGHT © 1979
FICHE 1 OF 1

MAY 1979
digital
MADE IN USA



IDENTIFICATION

B 1

SEQ 0001

PRODUCT CODE: AC-F115A-MC
PRODUCT NAME: CZRLH40 RL11/RLV11 CONTROLLER TEST PART 2
DATE CREATED: 5-JAN-79
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: D. DEKNIS, C. CAMPBELL

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

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

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

COPYRIGHT (C) 1979, DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	HOW TO RUN THIS DIAGNOSTIC
2.1.1	THE SIX STEPS OF EXECUTION
2.1.2	SAMPLE RUN-THROUGH
2.2	HOW TO CREATE A CHAINABLE FILE
2.3	DETAILS OF COMMANDS AND SYNTAX
2.3.1	TABLE OF COMMAND VALIDITY
2.3.2	COMMAND SYNTAX
2.4	EXTENDED P-TABLE DIALOGUE
2.5	HARDWARE PARAMETERS
2.6	SOFTWARE PARAMETERS
3.0	ERROR INFORMATION
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES

1.0 GENERAL INFORMATION
-----1.1 PROGRAM ABSTRACT
-----1.1.1 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC OCCUPIES 14.5K WORDS OF MEMORY AND IS COMPATIBLE WITH BOTH XXDP AND ACT. IT 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 B>). AT COMMAND MODE THE OPERATOR MAY ENTER ANY OF SEVERAL COMMANDS AS DESCRIBED BELOW.

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

1.1.2 DIAGNOSTIC INFORMATION

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

1.2 SYSTEM REQUIREMENTS
-----1.2.1 HARDWARE REQUIREMENTS

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

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

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

1.2.2 SOFTWARE REQUIREMENTS

CZRLHA RL11/RLV11 CTLR 2
(FORMERLY CZRLBB)

1.3 RELATED DOCUMENTS AND STANDARDS

RL01 USERS MANUAL (EK-RL01-UG-PRE)
XXDP USERS MANUAL

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

THE RL01/02 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING
PROGRAMS:

CVRLAAO	RLV11 RL01 DISKLESS TEST (RLV11 ONLY)
CZRLGAO	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 1)

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RL01/02 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 HOW TO RUN THIS DIAGNOSTIC

2.1.1 THE SIX STEPS OF EXECUTION

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

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

* STEP 1 *

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

<u>QUESTION</u>	<u>MEANING</u>
L-CLK (L) N ?	IS THERE AN L-CLOCK?
P-CLK (L) N ?	" " " P-CLOCK?
50HZ (L) N ?	IS THE POWER 50 CYCLES (EUROPE)?
LSI (L) N ?	IS MACHINE AN LSI?
LPT (L) N ?	IS THERE A LINE PRINTER?
MEM (K) (D) 16 ?	HOW MANY K OF MEMORY ARE THERE?

THE DEFAULTS (SHOWN AFTER EACH QUESTION) CAN BE SELECTED BY HITTING CARRIAGE RETURN. IT IS POSSIBLE THAT NOT ALL OF THE QUESTIONS WILL BE ASKED: FOR EXAMPLE, IF YOU SAY "YES" TO THE L-CLOCK QUESTION, THE P-CLOCK QUESTION WILL NOT BE ASKED.

IF NEITHER P OR L CLOCK ARE ANSWERED YES THE OPERATOR WILL BE ASKED TO TYPE TWO CHARACTERS 4 SECONDS APART.

* STEP 2 *

WHEN YOU HAVE ANSWERED ALL THE HARDCORE QUESTIONS, THE DIAGNOSTIC WILL ISSUE THE PROMPT "DS-B>". FROM THIS POINT UNTIL THE TIME WHEN YOU RESTART XXDP, YOU WILL BE TALKING TO THE DIAGNOSTIC, NOT XXDP. WE WILL REFER TO THE PRESENCE OF THIS PROMPT AS BEING IN DIAGNOSTIC COMMAND MODE, AS OPPOSED TO XXDP COMMAND MODE.

AT THIS POINT YOU WILL ENTER A "START" COMMAND. THIS IS NOT THE SAME AS THE XXDP "START" COMMAND, WHICH YOU ALREADY ISSUED IN RESPONSE TO THE XXDP DOT PROMPT. THIS "START" COMMAND CAN TAKE A NUMBER OF SWITCHES AND FLAGS (ALL OPTIONAL) AND THE DETAILS OF THESE ARE SET FORTH IN "2.3 DETAILS OF COMMANDS AND SYNTAX".

HOWEVER, IN ORDER TO USE THE PROGRAM, ALL YOU NEED TO SAY IS SOMETHING LIKE THIS:

```
STA/PASS:1/FLAGS:HOE
```

THINGS TO NOTE HERE:

1. ONLY THE FIRST THREE CHARACTERS OF THIS OR ANY COMMAND AT THE "DS-B>" LEVEL NEED TO BE TYPED.
2. THE "PASS" SWITCH SPECIFIES HOW MANY PASSES YOU DESIRE. A PASS CONSISTS OF RUNNING THE FULL DIAGNOSTIC AGAINST ALL UNITS BEING TESTED (THIS WILL BE EXPLAINED SHORTLY). ONE PASS IS SPECIFIED IN THE ABOVE EXAMPLE.
3. THE "FLAGS" SWITCH MAY SPECIFY ANY OF A NUMBER OF FLAGS, BUT THE MAIN USEFUL ONES ARE:

```
LOE    LOOP ON ERROR  
HOE    HALT ON ERROR  
IER    INHIBIT ERROR PRINTOUT
```

THE HOE FLAG IS SPECIFIED IN THE ABOVE EXAMPLE (WE'LL SEE WHY SHORTLY).

```
*****  
* STEP 3 *  
*****
```

WHEN YOU HAVE TYPED IN A "START" COMMAND, THE DIAGNOSTIC WILL COME BACK WITH THE QUESTION "# UNITS?" TO WHICH YOU SHOULD RESPOND BY TYPING IN THE NUMBER OF DEVICES YOU WISH TO TEST.

A WORD OF WARNING HERE: THE NUMBER OF UNITS DEPENDS ON THE TARGET DEVICE OF THE DIAGNOSTIC. FOR EXAMPLE, IF THE DIAGNOSTIC IS DIRECTED AT A DISK DRIVE, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF DRIVES TO BE TESTED. WHEREAS IF THE DIAGNOSTIC WAS DIRECTED AT THE DISK CONTROLLER, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF CONTROLLERS. THE TARGET DEVICE OF A DIAGNOSTIC CAN ALWAYS BE DETERMINED BY INSPECTING THE "HEADER" STATEMENT NEAR THE BEGINNING OF THE SOURCE CODE. ONE OF THE OPERANDS OF THIS "HEADER" STATEMENT SHOULD BE THE DEVICE TYPE OF THE DIAGNOSTIC.

* STEP 4 *

WHEN YOU HAVE TYPED IN THE NUMBER OF UNITS TO BE TESTED, THE DIAGNOSTIC WILL ASK YOU THE "HARDWARE QUESTIONS". THE ANSWERS TO THESE QUESTIONS ARE USED TO BUILD TABLES IN CORE, CALLED "HARDWARE P-TABLES". ONE HARDWARE P-TABLE WILL BE BUILT FOR EACH UNIT TO BE TESTED.

THERE ARE SEVERAL HARDWARE QUESTIONS AND THE ENTIRE SERIES WILL BE POSED N TIMES, WHERE N IS THE NUMBER OF UNITS.

THIS REPRESENTS A NEW PHILOSOPHY IN DIAGNOSTIC ENGINEERING. DIAGNOSTICS IN THE FUTURE WILL NOT BE WRITTEN TO AUTOSIZE OR ASSUME STANDARD ADDRESSES: INSTEAD, THEY WILL ASK THE OPERATOR FOR ALL THE INFORMATION THEY NEED TO TEST THE DEVICE.

* STEP 5 *

AFTER YOU HAVE ANSWERED ALL THE HARDWARE QUESTIONS (SEC 2.5) FOR ALL THE UNITS, YOU WILL BE ASKED "CHANGE SW?" IF YOU WANT TO BE ASKED THE SOFTWARE QUESTIONS THAT DETERMINE THE BEHAVIOR OF THIS PROGRAM, TYPE "Y". IF YOU WANT TO TAKE ALL THE DEFAULTS TO THESE QUESTIONS, TYPE "N". IF YOU TYPE "Y" YOU WILL BE ASKED THE SOFTWARE QUESTIONS (SEC 2.6), AND THE ANSWERS WILL BE PUT INTO THE SOFTWARE P-TABLE IN THE PROGRAM. THE SERIES OF QUESTIONS WILL BE ASKED JUST ONCE, REGARDLESS OF THE NUMBER OF UNITS TO BE TESTED.

* STEP 6 *

AFTER YOU HAVE ANSWERED THE SOFTWARE QUESTIONS, THE DIAGNOSTIC WILL BEGIN TO EXECUTE THE HARDWARE TEST CODE. THERE ARE SEVERAL THINGS THAT CAN HAPPEN NEXT, DEPENDING ON WHETHER A HARDWARE ERROR IS ENCOUNTERED AND ALSO ON WHAT SWITCH VALUES YOU SELECTED ON THE START COMMAND. CONSIDER THE POSSIBILITIES:

1. IF NO ERROR IS ENCOUNTERED, THEN THE DIAGNOSTIC WILL SIMPLY EXECUTE THE DESIRED NUMBER OF PASSES AND RETURN TO COMMAND MODE (PROMPT DS-B>).

2. IF AN ERROR IS ENCOUNTERED, THEN ONE OF THREE THINGS HAPPENS, DEPENDING ON THE SETTINGS OF THE HOE AND LOE FLAGS.

HOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND THE DIAGNOSTIC WILL RETURN TO COMMAND MODE.

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

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

2.1.2 SAMPLE RUN-THROUGH

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

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

1. ISSUE ANOTHER "START" COMMAND (THUS GOING THRU ALL OF STEPS 2, 3, 4, 5, AND 6 AGAIN)
2. ISSUE A "RESTART" COMMAND (SAME AS START COMMAND EXCEPT THAT THE HARDWARE QUESTIONS ARE NOT ASKED)
3. ISSUE A "CONTINUE" COMMAND (EXECUTION WILL RESUME AT THE BEGINNING OF THE PARTICULAR HARDWARE TEST (MOST DIAGNOSTICS CONSIST OF A NUMBER OF THESE) THAT IT WAS IN WHEN THE ERROR HALT OCCURED. NO QUESTIONS ASKED.
4. ISSUE A "PROCEED" COMMAND: EXECUTION WILL RESUME AT THE INSTRUCTION FOLLOWING THE ERROR REPORT (THIS IS A SPECIAL COMMAND AND CAN BE ISSUED ONLY AT A HALT)

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

PRO/FLAGS:IER:LOE:HOE=0

THIS WILL DO THE FOLLOWING:

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

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

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

1. START
2. RESTART
3. CONTINUE

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

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

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

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

1. START
2. RESTART

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

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

	BY WHOM ENTERED: -----
.R CZRLHA	O
CZRLH	D
L-CLK (L) N ? Y	D,O
50HZ (L) N ?	D
LSI (L) N ?	D
LPT (L) N ?	D
MEM (K) (D) 16 ?	D
DS-B>STA/PASS:1/FLAGS:HOE	D,O
# UNITS (D) ? 2	D,O
UNIT 1	D
RL11 (L) Y ?	D,O
BUS ADDRESS (O) 174400 ?	D,O
VECTOR (O) 160 ?	D,O
DRIVE (O) 0 ?	D,O
DRIVE TYPE = RL01 (L) Y ?	D,O
BR LEVEL (O) 5 ?	D,O
UNIT 2	D
RL11 (L) Y ?	D,O
BUS ADDRESS (O) 174400 ?	D,O
VECTOR (O) 160 ?	D,O
DRIVE (O) 0 ? 1	D,O
DRIVE TYPE = RL01 (L) ? N	D,O (N=RL02)
BR LEVEL (O) 5 ?	D,O
CHANGE SW (L) ? N	D,O
CZRLH HRD ERR 00004 TST 003 SUB 002 PC:004130 ERR HLT	
DS-B>PRO/FLAGS:IER:LOE:HOE=0	D,O

 AT THIS POINT THE DIAGNOSTIC IS LOOPING ON THE
 ERROR WITHOUT PRINTING ANYTHING. YOU CAN SCOPE
 THE ERROR UNTIL YOU HAVE LOCATED IT, THEN ^C OUT

```
^C                                0
DS-B>CON/FLAGS:HOE:IER:LOE=0      D,0
CHANGE SW (L) ? N                  D,0
CZRLH EOP 1                          D
^C
DS-B>RESTART/PASS:1                 D,0
CHANGE SW (L) ? N                    D,0
-----
-----
-----
-----
```

2.2 HOW TO CREATE A CHAINABLE FILE

THE DIAGNOSTIC AS RECEIVED FROM RELEASE ENGINEERING CANNOT BE RUN IN CHAIN MODE. THAT IS WHY IT BEARS THE EXTENSION "BIN" INSTEAD OF "BIC". THERE IS A WAY, HOWEVER, TO CREATE A CHAINABLE PROGRAM FROM WHAT YOU'VE GOT.

IT CONSISTS OF RUNNING THE PROGRAM WITH THE SPECIAL COMMAND "CCI" ISSUED WHERE YOU WOULD NORMALLY ISSUE A START COMMAND (TO THE PROMPT DS-B>). THIS COMMAND CAUSES THE DIAGNOSTIC TO GO THRU ALL THE QUESTIONS AND ANSWERS AND THEN TO HALT, JUST WHERE IT WOULD ORDINARILY BEGIN EXECUTION OF THE HARDWARE TEST CODE. AT THIS POINT YOU CAN DUMP THE PROGRAM AS IT SITS IN CORE TO THE LOAD MEDIUM, WITH THE NEW EXTENSION "BIC".

HERE IS A SAMPLE DIALOGUE TO ACCOMPLISH THIS:

```
.R UPD2
RESTART: XXXXXX
*CLR
*LOAD DIAG.BIN
XFER:200 CORE:0,60602
*START 200
L-CLK (L) N ?
-----
-----
-----
```

DS-B>CCI
UNITS (D) ? 4

CHANGE SW (L) ? N
PTAB END: 60632

AT THIS POINT THE MACHINE HALTS AND
YOU MUST RESTART AT ADDRESS XXXXXX

*MICORE 60632
CORE: 0,60632
*DUMP DK0: DIAG.BIC

THE RESULT OF DOING THIS IS THAT YOU CAN NOW BUILD AN XXDP CHAIN
FILE CONTAINING THE XXDP COMMAND

.R DIAG.BIC

AND THE DIAGNOSTIC WILL EXECUTE WITHOUT MANUAL INTERVENTION, USING
THE ANSWERS THAT YOU GAVE IT WHEN YOU DID THE CCI COMMAND.

2.3 DETAILS OF COMMANDS AND SYNTAX

2.3.1 TABLE OF COMMAND VALIDITY

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

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

2.3.2 COMMAND SYNTAX

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

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. THE MESSAGE "# UNITS?" IS PRINTED. THE START COMMAND MAY BE ISSUED WHEN DIAGNOSTIC COMMAND MODE HAS BEEN ENTERED VIA ONE OF THE FOLLOWING: A) OPERATOR TYPED "RUN DIAGNOSTIC" B) DIAGNOSTIC

FINISHED EXECUTING () ERROR WAS ENCOUNTERED WITH HOE FLAG SET D) OPERATOR ENTERED CONTROL/C. AFTER THE OPERATOR RESPONDS TO "UNITS?", THE HARDWARE DIALOGUE IS INITIATED. WHEN IT IS COMPLETED, THE QUESTIONS "CHANGE SW?" IS ISSUED, AND THE ANSWERS, IF GIVEN, BECOME THE NEW DEFAULTS. THEREFORE IT IS NECESSARY TO RELOAD THE PROGRAM IN ORDER TO RETURN TO THE LOAD DEFAULTS.

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

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

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

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

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

IER INHIBIT ERROR REPORTING

IBE INHIBIT BASIC ERROR REPORTS

IXE INHIBIT EXTENDED ERROR REPORTS

PRI DIRECT ALL MESSAGES TO A LINE PRINTER

PNT PRINT NUMBER OF TEST BEING EXECUTED

BOE BELL ON ERROR

UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS

ISR INHIBIT STATISTICAL REPORTS

IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

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

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

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

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

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

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

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

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

F SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

PROCEED//FLAGS:<FLAG-LIST>

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

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

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

CCI/TEST:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR

THE DIAGNOSTIC EXECUTES THRU ALL OPERATOR DIALOGUE AND HALTS AT THE HARDWARE TEST CODE. NOW THE OPERATOR CAN DUMP THE CORE IMAGE TO THE MEDIUM WITH A BIC EXTENSION.

THE BIC FILE MUST BE HANDLED DIFFERENTLY DEPENDING ON WHETHER IT IS RUN MANUALLY OR IN CHAIN MODE. IF RUN MANUALLY IT CAN BE INVOKED EITHER WITH A "START" (IN WHICH CASE IT WILL BEHAVE LIKE THE BIN FILE: THE PRE-GENERATED ANSWERS TO OPERATOR QUESTIONS WILL BE IGNORED) OR WITH A "RESTART" (IN WHICH CASE THE PRE-GENERATED OPERATOR ANSWERS WILL BE USED).

IF RUN IN CHAIN MODE, AUTOMATIC EXECUTION WILL COMMENCE IMMEDIATELY FROM THE XXDP COMMAND ".R DIAG". THE COMMAND PROMPT "DS-B>" WILL NOT BE ISSUED.

ANY SWITCHES SPECIFIED ON THE CCI COMMAND WILL CARRY OVER WHEN THE BIC FILE IS RUN IN CHAIN MODE (EXCEPT THAT UAM IS ALWAYS SET THERE) BUT WILL NOT CARRY OVER WHEN IT IS RUN MANUALLY.

TO DO A CCI ON A FULL SIZED DIAGNOSTIC (14.5K WORDS), A MACHINE SIZE LARGER THAN 16K IS REQUIRED. THE EXACT SIZE NEEDED DEPENDS ON WHICH UTILITY IS USED TO EXECUTE THE DIAGNOSTIC AT CCI TIME.

DRO(P)/UNITS:UNIT-LIST

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

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

ADD/UNITS:UNIT-LIST

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

PRI(NT)

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

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

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

FLA(GS)

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

ZFL(AGS)

ALL FLAGS ARE CLEARED.

2.4 EXTENDED 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), 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.

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 8 RL UNITS, AND THAT THERE ARE FIVE (5) HARDWARE PARAMETERS FOR EACH (5 SLOTS IN THE P-TABLE, 5 HARDWARE QUESTIONS IN THE DIALOGUE).

FOLLOWING IS THE DIALOGUE FOR THIS 8 RLOX DRIVE SYSTEM. THIS SYSTEM HAS TWO (2) RL11 TYPE CONTROLLERS ALL TO BE SET AT "BR LEVEL" 5. THE FIRST 4 DRIVES ARE RL01'S AND THE LAST 4 DRIVES ARE RL02'S (ON THE SECOND CONTROLLER):

UNITS (D) ? 8

UNIT 1

RL11 (L) Y ?

BUS ADDRESS (O) 174400 ?

VECTOR (O) 160 ?

DRIVE (O) 0 ? 0-3

DRIVE TYPE = RL01 (L) Y ?

BR LEVEL (O) 5 ?

UNIT 5

RL11 (L) Y ?

BUS ADDRESS (O) 174400 ? 175400

VECTOR (O) 160 ? 164

DRIVE (O) 0 ? 0-3

DRIVE TYPE = RL01 (L) Y ? N

BR LEVEL (O) 5 ?

THE FIRST TIME THRU THE P-TABLE QUESTIONS THE DEFAULT VALUES ARE USED FOR THE CONTROLLER TYPE (QUESTION #1), CSR ADDRESS OF THE CONTROLLER (QUESTION #2), THE CONTROLLER VECTOR ASSIGNMENT (QUESTION #3), THE DRIVE TYPE (QUESTION #5), AND THE "BR LEVEL" (QUESTION #6). THE ACTUAL UNIT NUMBERS OF THE RL01'S FOR QUESTION #4 WAS ASSIGNED 0 THRU 3 FOR THE FIRST 4 P-TABLE SLOTS.

THE SECOND TIME THRU THE P-TABLE QUESTIONS (FOR THE RL02 ASSIGNMENT ON THE SECOND CONTROLLER), THE FIRST QUESTION DEFAULTED TO "RL11" TYPE CONTROLLER. THE SECOND QUESTION WAS ANSWERED TO REFLECT THE CHANGE IN CSR ADDRESS FOR THE RL02 CONTROLLER (175400). THE SECOND CONTROLLER'S VECTOR WAS ALSO CHANGED TO 164 IN QUESTION #3. THE RL02 TEST UNIT NUMBERS WERE ASSIGNED VALUES 0 TO 3 IN QUESTION #4 AND THE DRIVE TYPE WAS SET FOR RL02'S FOR THE REMAINING 4 UNITS IN QUESTION #5. THE LAST QUESTION WAS DEFAULTED USING THE "BR LEVEL" FROM THE FIRST PASS.

2.5 HARDWARE PARAMETERS

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

RL11 (L) Y?

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

BUS ADDRESS (O) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (O) 160?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

DRIVE (O) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

DRIVE TYPE = RL01 (L) ?

ANSWER NO (N) IF DRIVE IS AN RL02

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

2.6 SOFTWARE PARAMETERS

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

"CHANGE S.W. ?"

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

"DROP ON ERROR LIMIT (L) Y?"

TO ALLOW THE UNIT TO BE DROPPED ONCE A PREDETERMINED NUMBER OF ER-

RORS ARE ENCOUNTERED.

ANSWER Y OR N

"ERROR LIMIT (D) 10?"

NUMBER OF ERRORS ALLOWED BEFORE DROPPING UNIT.

ANSWER 1 TO 65K

"AUTOSIZE (L) N? "

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

ANSWER Y OR N

"COMPARE DATA ON DCK (L) N?"

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

ANSWER Y OR N

"# OF WORDS IN ERROR REPORTED (D) 3? "

NUMBER OF MISCOMPARES TO BE PRINTED ON CONSOLE DEVICE.

ANSWER 0 - 128

3.0 ERROR INFORMATION

ALL ERROR INFORMATION IS PRINTED ON THE CONSOLE DECIVE. 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

3.2 ERROR HALTS

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

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

4.2 PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

5.0 DEVICE INFORMATION TABLES

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

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

- BIT 15 - COMPOSITE ERROR
- BIT 14 - DRIVE ERROR
- BIT 13 - NON EXISTANT MEMORY ERROR
- BIT 12 - HEADER NOT FOUND (WITH BIT 10 SET)
- DATA LATE (WITH BIT 10 CLEAR)
- BIT 11 - HEADER CRC (WITH BIT 10 SET)
- DATA CRC (WITH BIT 10 CLEAR)
- BIT 10 - OPERATION INCOMPLETE
- BIT 9/8 - DRIVE SELECT (0-3)
- BIT 7 - CONTROLLER READY
- BIT 6 - INTERRUPT ENABLE

BIT 5 - EXTENDED BUS ADDRESS (BIT 17)
BIT 4 - EXTENDED BUS ADDRESS (BIT 16)
BIT 3-1 - FUNCTION CODE
 0 - NOP (PDP-11) MAINT (LSI-11)
 1 - WRITE CHECK
 2 - GET DRIVE STATUS
 3 - SEEK
 4 - READ HEADER
 5 - WRITE DATA
 6 - READ DATA
 7 - READ WITHOUT HEADER COMPARE

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

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

RLDA - DISK ADDRESS REGISTER (XXXXX4)

FOR READ/WRITE FUNCTIONS

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

FOR SEEK FUNCTION

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

FOR GET STATUS FUNCTION

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

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

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

FOR READ HEADER FUNCTION

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

FOR GET STATUS FUNCTION

HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR
BIT 14 - CURRENT HEAD ERROR (CHE)
BIT 13 - WRITE LOCK STATUS (WL)
BIT 12 - SEEK TIME OUT (SKTO)
BIT 11 - SPIN ERROR (SPE)
BIT 10 - WRITE GATE ERROR (WGE)
BIT 9 - VOLUME CHECK (VC)
BIT 8 - DRIVE SELECT ERROR (DSE)
BIT 7 - DRIVE TYPE IS RL02 IF SET
BIT 6 - SURFACE (0=UPPPER, 1=LOWER)
BIT 5 - COVER OPEN
BIT 4 - HEADS HOME
BIT 3 - BRUSHES HOME
BIT 2-0 - STATE BITS
 0 - LOAD STATE
 1 - SPIN UP
 2 - BRUSH CYCLE
 3 - LOAD HEADS
 4 - SEEK - TRACK COUNTING
 5 - SEEK - LINEAR MODE
 6 - UNLOAD HEADS
 7 - SPIN DOWN

6.0

TEST SUMMARIES

TEST 1 - WRITE NPR INTEGRITY

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

TEST 2 - WRITE FUNCTION

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

TEST 3 - WRITE FUNCTION INTERRUPT

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

TEST 4 - PROPER INCREMENT OF RLBA ON WRITE

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

TEST 5 - PROPER INCREMENT OF RLDA ON WRITE

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

TEST 6 - FORCE HEADER NOT FOUND WITH WRITE

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

TEST 7 - FORCE INTERRUPT WITH HNF

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

TEST 8 - CHECK OPI TIME WITH HNF

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

TEST 9 - MULTIPLE SECTOR TRANSFER ON WRITE

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

TEST 10 - CHECK DIRECTION OF WRITE NPR

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

TEST 11 - CHECK FULL INCREMENT OF RLBA

THIS TEST WILL CHECK THAT THE RLBA CAN INCREMENT OF THE FULL 16 BIT RANGE. THIS IS DONE BY ISSUING A ONE WORD WRITE TO CHECK EACH BIT TOGGLE FROM 1-0 AND 0-1. THIS IS DONE FROM 0 TO 17776 REGARDLESS OF MEMORY SIZE.

TEST 12 - BA BIT 16 INCREMENT

THIS TEST WILL CHECK THAT BUS ADDRESS BIT 16 WILL SET WHEN THE RLBA IS 17776. AND THAT THE RLBA GOES TO 0.

TEST 13 - BA BIT 17 INCREMENT

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

TEST 14 - READ NPR INTEGRITY

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

TEST 15 - READ FUNCTION

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

TEST 16 - READ FUNCTION INTERRUPT

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

TEST 17 - CHECK DIRECTION OF READ NPR

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

TEST 18 - PROPER INCREMENT OF RLBA ON READ

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

TEST 19 - PROPER INCREMENT OF RLDA ON READ

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

TEST 20 - FORCE HEADER NOT FOUND WITH READ

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

TEST 21 - FORCE INTERRUPT WITH HNF

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

TEST 22 - CHECK HEADER COMPARE LOGIC

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

TEST 23 - MULTIPLE SECTOR TRANSFER ON READ

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

TEST 24 - FORCE HNF AT END OF TRACK

THIS TEST WILL CHECK THE ABILITY TO DETECT HEADER NOT FOUND AT THE END OF A TRACK. THIS DONE BY SETTING UP FOR A TWO SECTOR READ AT SECTOR 39.

TEST 25 - FORCE NON-EXISTANT MEMORY ERROR

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

TEST 26 - FORCE NXM UNDER INTERRUPT

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

TEST 27 - CHECK READ WRITE LOOP

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

TEST 28 - CHECK OF SILO LINES

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

TEST 29 - CHECK THROUGHPUT OF SILO

THIS TEST WILL ATTEMPT TO CHECK THAT THE FALL THROUGH OF THE SILO IS WORKING CORRECTLY. WE WRITE A SECTOR OF 128 UNIQUE

PATTERNS AND READ IT BACK CHECKING THAT EACH LOCATION IS UNIQUE AND CORRECT.

TEST 30 - CHECK ZERO FILL ON WRITE

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

TEST 31 - CHECK SECTOR BITS ON HEADER COMPARE

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

TEST 32 - WRITE CHECK NPR INTEGRITY

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

TEST 33 - WRITE CHECK FUNCTION

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

TEST 34 - WRITE CHECK FUNCTION INTERRUPT

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

TEST 35 - PROPER INCREMENT OF RLBA ON WRITE CHECK

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

TEST 36 - PROPER INCREMENT OF RLDA ON WRITE CHECK

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

TEST 37 - MULTIPLE SECTOR WRITE CHECK

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

TEST 38 - FORCE DCK WITH WRITE CHECK

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

TEST 39 - FORCE DCK WITH WRITE CHECK INTERRUPT

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

TEST 40 - CHECK ZERO FILL ON WRITE WITH WRITE CHECK

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

TEST 41 - 42 - EXTENDED CHECK OF WRITE CHECK

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

TEST 43 - READ WITHOUT HEADER COMPARE

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

TEST 44 - READ WITHOUT HEADER COMPARE INTERRUPT

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

TEST 45 - CHECK RD W/O HDR CMP READS

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

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

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

TEST 47 - CHECK RLDA DOES INCREMENT

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

1- 88	GLOBAL DATA
1- 155	LIST TO CHECK HEADER COMPARE LOGIC
1- 282	BUFFER FOR READ/WRITE
1- 288	GLOBAL TEXT
1- 395	GLOBAL ERRORS
1- 643	INITIALIZATION CODE
2- 132	GLOBAL SUBROUTINES
2- 166	ROUTINE TO CHECK FOR CONTROLLER ERRORS
2- 228	LOAD RLCS
2- 465	**TEST 1** - WRITE NPR INTEGRITY
2- 516	**TEST 2** - WRITE FUNCTION
2- 572	**TEST 3** - WRITE FUNCTION INTERRUPT
2- 614	**TEST 4** - PROPER INCREMENT OF RLBA ON WRITE
2- 657	**TEST 5** - PROPER INCREMENT OF RLDA ON WRITE
2- 700	**TEST 6** - FORCE HEADER NOT FOUND WITH WRITE
2- 743	**TEST 7** - FORCE HEADER NOT FOUND WITH WRITE INTERRUPT
2- 799	**TEST 8** - CHECK OPI TIME WITH HDR NT FND
2- 862	**TEST 9** - MULTIPLE SECTOR TRANSFER ON WRITE
2- 915	**TEST 10** - CHECK DIRECTION OF WRITE NPR
2- 973	**TEST 11** - CHECK FULL RLBA INCREMENT
2-1023	**TEST 12** - BA BIT 16 INCREMENT
2-1079	**TEST 13** - BA BIT 17 INCREMENT
2-1135	**TEST 14** - TEST READ NPR INTEGRITY
2-1178	**TEST 15** - READ FUNCTION
2-1212	**TEST 16** - READ FUNCTION INTERRUPT
2-1252	**TEST 17** - CHECK READ NPR DIRECTION
2-1314	**TEST 18** - PROPER INCREMENT OF RLBA ON READ
2-1354	**TEST 19** - PROPER INCREMENT OF RLDA ON READ
2-1396	**TEST 20** - FORCE HEADER NOT FOUND WITH READ
2-1435	**TEST 21** - FORCE HEADER NOT FOUND WITH READ INTERRUPT
2-1484	**TEST 22** - CHECK HEADER COMPARE LOGIC
2-1622	**TEST 23** - CHECK MULTIPLE SECTORS ON READ
2-1681	**TEST 24** - FORCE HDR NT FND AT END OF TRACK
2-1717	**TEST 25** - FORCE NON-EXISTANT MEMORY ERROR
2-1760	**TEST 26** - FORCE NON-EXISTANT MEMORY ERROR INTERRUPT
2-1807	**TEST 27** - CHECK READ WRITE LOOP
2-1893	**TEST 28** - CHECK SILO LINES
2-1990	**TEST 29** - CHECK THROUGHPUT OF SILO
2-2086	**TEST 30** - CHECK ZERO FILL ON WRITE
2-2191	**TEST 31** - CHECK SECTOR BITS OF HEADER COMPARE
2-2303	**TEST 32** - WRITE CHECK NPR INTEGRITY
2-2386	**TEST 33** - WRITE CHECK FUNCTION
2-2451	**TEST 34** - WRITE CHECK FUNCTION INTERRUPT
2-2522	**TEST 35** - PROPER INCREMENT OF RLBA ON WRITE CHECK
2-2595	**TEST 36** - PROPER INCREMENT OF RLDA ON WRITE CHECK
2-2668	**TEST 37** - MULTIPLE SECTOR WRITE CHECK
2-2754	**TEST 38** - FORCE DCK WITH WRITE CHECK
2-2827	**TEST 39** - FORCE DCK WITH WRITE CHECK INTERRUPT
2-2911	**TEST 40** - CHECK ZERO FILL ON WRITE WITH WRITE CHECK
2-2988	**TEST 41** - EXTENDED CHECK OF WRITE CHECK FUNCTION
2-3080	**TEST 42** - EXTENDED CHECK OF WRITE CHECK FUNCTION
2-3169	**TEST 43** - READ WITHOUT HEADER COMPARE FUNCTION
2-3199	**TEST 44** - READ WITHOUT HEADER COMPARE FUNCTION INTERRUPT
2-3235	**TEST 45** - CHECK RD W/O HDR CMP ACTUALLY READS
3- 25	**TEST 46** - CHECK RLBA INCREMENT WITH RD W/O HDR CMP
3- 71	**TEST 47** - CHECK RLDA DOES INCREMENT WITH RD W/O HDR CMP

4- 1

DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP

M 3

SEQ 0033

1		.TITLE CZRLHAO RL11/RLV11 CTLR 2
2		.ENABLE AMA
3	000000	.ENABLE ABS
4		.NLIST ME,CND,MD
5		
6		.MCALL SVC
7		
18		
19	002000	.=2000
20		
21	002000	SVC
22	000000	SVCINS=0
23	000000	SVCTAG=0
24		
25	002000	POINTER BGNSW,BGNSFT,BGNDU
26		
27	002000	BGNMOD MDHEDR
28		
29	002000	HEADER CZRLH,A,0,60,60,4,RL01
	002000	.ASCII /C/
	002001	.ASCII /Z/
	002002	.ASCII /R/
	002003	.ASCII /L/
	002004	.ASCII /H/
	002005	.BYTE 0
	002006	.BYTE 0
	002007	.BYTE 0
	002010	.ASCII /A/
	002011	.ASCII /O/
	002012	.WORD 0
	002014	.WORD 4
	002016	.WORD L\$HARD
	002020	.WORD L\$SOFT
	002022	.WORD L\$HW
	002024	.WORD L\$SW
	002026	.WORD L\$LAST
	002030	.WORD 0
	002032	.WORD 0
	002034	.WORD 0
	002036	.WORD 0
	002040	.WORD L\$DISPATCH
	002042	.WORD 0
	002044	.WORD 0
	002046	.WORD 0
	002050	.BYTE C\$REVISION
	002051	.BYTE C\$EDIT
	002052	.WORD 60
	002054	.WORD 60
	002056	.WORD 0
	002060	.WORD 0
	002062	.WORD 0
	002064	.WORD L\$DVTYP
	002066	.WORD 0
	002070	.WORD L\$DR
	002072	.WORD L\$DRST
	002074	.WORD 0
	002076	.WORD L\$DU

```

002100 000014      .WORD 14
002102 000000      .WORD 0
002104 017352      .WORD L$INIT
002106 020226      .WORD L$CLEAN
30
31 002110          ENDMOD
32
33 002110          DEVREG
002110 000000      .WORD 0
                          .BLKW
34
35 002114          DEVTYP <RL01,RL02>
002114 122 114 060 .ASCIZ /RL01,RL02/
002117 061 054 122
002122 114 060 062
002125 000

36 002126          .EVEN
37 002126          BGNMOD GLBEQAT
                          EQUALS
38 000001          DRDY=BIT0      ;DRIVE READY (RLCS)
39 000100          INTEN=BIT6     ;INTERRUPT ENABLE (RLCS)
40 100000          ERR=BIT15     ;RL11 ERROR (RLCS)
41 040000          DERR=BIT14    ;RL01 DRIVE ERROR (RLCS)
42 002000          OPI=BIT10     ;OPERATION INCOMPLETE (RLCS)
43 000200          CRDY=BIT7     ;CONTROLLER READY (RLCS)
44 000040          BA17=BIT5     ;EXTENDED ADDRESS BIT 17 (RLCS)
45 000020          BA16=BIT4     ;EXTENDED ADDRESS BIT 16 (RLCS)
46 020000          NXM=BIT13     ;NON-EXISTANT MEMORY (RLCS)
47 000000          DS0=0         ;DRIVE SELECT 0 (RLCS)
48 000400          DS1=BIT8     ;DRIVE SELECT 1 (RLCS)
49 001000          DS2=BIT9     ;DRIVE SELECT 2 (RLCS)
50 001400          DS3=BIT8:BIT9 ;DRIVE SELECT 3 (RLCS)
51 000000          NOOP0=0      ;FUNCTION-NOOP(0)
52 000002          WRCHK=BIT1    ;WRITE CHECK FUNCTION
53 000004          GSTAT=BIT2    ;GET STATUS FUNCTION
54 000006          SEEK=BIT2:BIT1 ;SEEK FUNCTION
55 000010          RDHDR=BIT3     ;READ HEADER FUNCTION
56 000012          WRITE=BIT3:BIT1 ;WRITE DATA FUNCTION
57 000014          READ=BIT3:BIT2 ;READ DATA FUNCTION
58 000016          RDNHD=BIT3:BIT2:BIT1 ;READ W/O HEADER VERIFICATION
59 000202          GODRVR=BIT1:BIT7 ;CRDY AND DRDY
60 000010          DRST=BIT3     ;DRIVE RESET (RLDA)
61 000002          GSBIT=BIT1    ;GET STATUS BIT (RLDA)
62 000001          MK=BIT0       ;MARKER BIT (RLDA)
63 000004          SIGN=BIT2     ;SIGN BIT (RLDA)
64 000100          RHHS=BIT6     ;HEAD SELECT IN READ HEADER
65 000100          STHS=BIT6     ;HEAD SELECT IN STATUS BACK
66 000020          DAHS=BIT4     ;HEAD SELECT IN SEEK
67
68
69 000000          CSR=0
70 000002          VECT=2
71 000004          PRIOR=4
72 000006          TYPDR=6
73 000010          DRB1=10
74 000012          CNT=12
75

```

```

76          ;OFFSET FOR SOFTWARE P-TABLE
77
78          000000      DLT=0
79          000002      ELT=2
80          000004      SIZE=4
81          000006      DMPCK=6
82          000010      DLMT=10
83          000012      ANS=12
84
85 002126      BGNMOD      ENDMOD
86 002126      BGNMOD      GLBDAT
87
88          .SBITL      GLOBAL DATA
89 002126      000000      T.DRIVE:      .WORD      0
90
91 002130      000000      CHECK:      .WORD      0
92 002132      000000      T.CRC:      .WORD      0
93 002134      000000      WHY:      .WORD      0
94 002136      000000      CDCNT:      .WORD      0
95 002140      000004      ERRVEC:      .WORD      4
96 002142      000000      DRIVE:      .WORD      0
97 002144      000000      UUT:      .WORD      0
98 002146      000000      UNITST:      .WORD      0
99 002150      000000      TRPFLG:      .WORD      0
100 002152      000000      INTFLG:      .WORD      0      ; INTERRUPT OCCURANCE FLAG
101 002154      000000      LDCSR:      .WORD      0      ; LOCATION TO FORM RLCS
102 002156      000077      SECMSK:      .WORD      77      ; MASK OUT SECTOR
103 002160      120001      XPOLY:      .WORD      120001      ; POLYNOMIAL FOR CRC 16
104 002162      000000      BCCFBK:      .WORD      0      ; LOCATION USED BY "SIMBCC"
105 002164      000000      CALBCC:      .WORD      0      ; LOCATION USED BY "SIMBCC"
106 002166      000000      TMP0:      .WORD      0
107 002170      000000      TMP1:      .WORD      0
108 002172      000000      TMP2:      .WORD      0
109 002174      000000      GDDAT:      .WORD      0
110 002176      000000      BDDAT:      .WORD      0
111 002200      000000      TEMP2:      .WORD      0      ; LOCATION USED BY "SIMBCC"
112 002202      000000      TEMP3:      .WORD      0      ; LOCATION USED BY "SIMBCC"
113 002204      000000      TEMP4:      .WORD      0      ; LOCATION USED BY "SIMBCC"
114 002206      000000      FIRST:      .WORD      0      ; FIRST SECTOR READ
115 002210      177700      CYLMSK:      .WORD      177700      ; MASK CYLINDER AND HEAD SELECT
116 002212      000050      MXSEC1:      .WORD      40      ; MAX SECTOR ADDRESS +1
117 002214      000047      MAXSEC:      .WORD      39      ; MAX SECTOR ADDRESS
118 002216      000000      DWORD:      .WORD      0      ; DIFFERENCE WORD (SEEK)
119 002220      177600      MAXCYL:      .WORD      177600      ; MAXIMUM CYLINDER ADDRESS
120 002222      000000      SVHD:      .WORD      0      ; SAVE CURRENT HEAD SELECT
121 002224      000000      B.CS:      .WORD      0      ; CS - BEFORE OPERATION
122 002226      000000      B.BA:      .WORD      0      ; BA - BEFORE OPERATION
123 002230      000000      B.DA:      .WORD      0      ; DA - BEFORE OPERATION
124 002232      000000      B.MP:      .WORD      0      ; MP - BEFORE OPERATION
125 002234      000000      E.CS:      .WORD      0      ; CS - AT OCCURANCE OF ERROR
126 002236      000000      E.BA:      .WORD      0      ; BA - AT OCCURANCE OF ERROR
127 002240      000000      E.DA:      .WORD      0      ; DA - AT OCCURANCE OF ERROR
128 002242      000000      E.MP:      .WORD      0      ; MP - AT OCCURANCE OF ERROR
129 002244      000000      E.MP1:      .WORD      0
130 002246      000000      E.MP2:      .WORD      0
131 002250      000000      RLCS:      .WORD      0
132 002252      000000      RLBA:      .WORD      0

```

133 002254 000000
134 002256 000000
135 002260 000000
136 002262 000000
137 002264 000000
138 002266 000000
139 002270 000000
140 002272 000000
141 002274 000000
142 002276 001212
143 002300 000233
144 002302 000620
145 002304 000240
146 002306 000000
147 002310 000000
148 002312 000000
149 002314 000000
150 002316 000000
151 002320 000000
152 002322

RLDA: .WORD 0
RLMP: .WORD 0
BCSR: .WORD 0 ;CSR FROM P TABLE
BVEC: .WORD 0 ;VECTOR FROM P TABLE
BPRIOR: .WORD 0 ;BR LEVEVL FROM P TABLE
FNDFNC: .WORD 0
XMEM: .WORD 0
TRYFNC: .WORD 0
ERFLG: .WORD 0
LOPIMX: .WORD 650.
LOPIMN: .WORD 155.
UOPIMX: .WORD 400.
UOPIMN: .WORD 160.
OPIMN: .WORD 0
OPIMX: .WORD 0
PWRFLG: .WORD 0
T.CNTRL: .WORD 0
DERFLG: .WORD 0
ERPOINT: .WORD 0
ERCOUNT: .BLKW 64.

153
154
155
156 002522 000000
157 002524 000001
158 002526 000002
159 002530 000004
160 002532 000010
161 002534 000020
162 002536 000040
163 002540 000100
164 002542 000200
165 002544 000400
166 002546 001000
167 002550 002000
168 002552 004000
169 002554 010000
170 002556 020000
171 002560 040000
172 002562 000003
173 002564 000007
174 002566 000017
175 002570 000037
176 002572 000137
177 002574 000337
178 002576 000737
179 002600 001737
180 002602 003737
181 002604 007737
182 002606 017737
183 002610 037737
184 002612 077737
185 002614 077736
186 002616 077734
187 002620 077730
188 002622 077720
189 002624 077700

.SBTTL LIST TO CHECK HEADER COMPARE LOGIC
HDRTAB: .WORD 0 ;WALK 1
.WORD BIT0
.WORD BIT1
.WORD BIT2
.WORD BIT3
.WORD BIT4
.WORD BIT5
.WORD BIT6
.WORD BIT7
.WORD BIT8
.WORD BIT9
.WORD BIT10
.WORD BIT11
.WORD BIT12
.WORD BIT13
.WORD BIT14
.WORD 3 ;GROW 1
.WORD 7
.WORD 17
.WORD 37
.WORD 137
.WORD 337
.WORD 737
.WORD 1737
.WORD 3737
.WORD 7737
.WORD 17737
.WORD 37737
.WORD 77737
.WORD 77736 ;GROW 0
.WORD 77734
.WORD 77730
.WORD 77720
.WORD 77700

190	002626	077600	.WORD	77600
191	002630	077400	.WORD	77400
192	002632	077000	.WORD	77000
193	002634	076000	.WORD	76000
194	002636	074000	.WORD	74000
195	002640	070000	.WORD	70000
196	002642	060000	.WORD	60000
197	002644	040000	.WORD	40000
198	002646	077735	.WORD	77735
199	002650	077733	.WORD	77733
200	002652	077727	.WORD	77727
201	002654	077717	.WORD	77717
202	002656	077637	.WORD	77637
203	002660	077537	.WORD	77537
204	002662	077337	.WORD	77337
205	002664	076737	.WORD	76737
206	002666	075737	.WORD	75737
207	002670	073737	.WORD	73737
208	002672	067737	.WORD	67737
209	002674	057737	.WORD	57737
210	002676	037737	.WORD	37737
211	002700	000000	.WORD	0
212	002702	000000	.WORD	0
213	002704	000001	.WORD	BIT0
214	002706	000002	.WORD	BIT1
215	002710	000004	.WORD	BIT2
216	002712	000010	.WORD	BIT3
217	002714	000020	.WORD	BIT4
218	002716	000040	.WORD	BIT5
219	002720	000100	.WORD	BIT6
220	002722	000200	.WORD	BIT7
221	002724	000400	.WORD	BIT8
222	002726	001000	.WORD	BIT9
223	002730	002000	.WORD	BIT10
224	002732	004000	.WORD	BIT11
225	002734	010000	.WORD	BIT12
226	002736	020000	.WORD	BIT13
227	002740	040000	.WORD	BIT14
228	002742	100000	.WORD	BIT15
229	002744	000003	.WORD	3
230	002746	000007	.WORD	7
231	002750	000017	.WORD	17
232	002752	000037	.WORD	37
233	002754	000137	.WORD	137
234	002756	000337	.WORD	337
235	002760	000737	.WORD	737
236	002762	001737	.WORD	1737
237	002764	003737	.WORD	3737
238	002766	007737	.WORD	7737
239	002770	017737	.WORD	17737
240	002772	037737	.WORD	37737
241	002774	077737	.WORD	77737
242	002776	177737	.WORD	177737
243	003000	177736	.WORD	177736
244	003002	177734	.WORD	177734
245	003004	177730	.WORD	177730
246	003006	177720	.WORD	177720

HDREND:
HTAB:

;WALK 0

;WALK 1

;GROW 1

;GROW 0

247	003010	177700			.WORD	177700	
248	003012	177600			.WORD	177600	
249	003014	177400			.WORD	177400	
250	003016	177000			.WORD	177000	
251	003020	176000			.WORD	176000	
252	003022	174000			.WORD	174000	
253	003024	170000			.WORD	170000	
254	003026	160000			.WORD	160000	
255	003030	140000			.WORD	140000	
256	003032	100000			.WORD	100000	
257	003034	177735			.WORD	177735	
258	003036	177733			.WORD	177733	:WALK 0
259	003040	177727			.WORD	177727	
260	003042	177717			.WORD	177717	
261	003044	177637			.WORD	177637	
262	003046	177537			.WORD	177537	
263	003050	177337			.WORD	177337	
264	003052	176737			.WORD	176737	
265	003054	175737			.WORD	175737	
266	003056	173737			.WORD	173737	
267	003060	167737			.WORD	167737	
268	003062	157737			.WORD	157737	
269	003064	137737			.WORD	137737	
270	003066	000000		HEND:	.WORD	0	
271							
272							
273	003070	000001	000002	000004	DATPAT: .WORD	1,2,4,10,20,40,100,200,400,1000,2000,4000,10000,20000,40000,100000	
	003076	000010	000020	000040			
	003104	000100	000200	000400			
	003112	001000	002000	004000			
	003120	010000	020000	040000			
	003126	100000					
274	003130	177777	177776	177775	.WORD	177777,177776,177775,177773,177767,17757,177737,177677	
	003136	177773	177767	177757			
	003144	177737	177677				
275	003150	177577	177377	176777	.WORD	177577,177377,176777,175777,173777,167777,157777,137777	
	003156	175777	173777	167777			
	003164	157777	137777				
276	003170	077777	177774	177770	.WORD	77777,177774,177770,177760,177740,177700,177600,177400	
	003176	177760	177740	177700			
	003204	177600	177400				
277	003210	177000	176000	174000	.WORD	177000,176000,174000,170000,160000,140000,3,7,17,37,77	
	003216	170000	160000	140000			
	003224	000003	000007	000017			
	003232	000037	000077				
278	003236	000177	000377	000777	.WORD	177,377,777,1777,3777,7777,17777,37777,0	
	003244	001777	003777	007777			
	003252	017777	037777	000000			
279							
280							
281							
282					.SBTTL	BUFFER FOR READ/WRITE	
283	003260				BUF: .BLKW	1024.	
284							
285							
286	007260				ENDMOD		
287							

				.SBTTL	GLOBAL TEXT
288				BGNMOD	GLBTXT
289	007260			NORFS:	.ASCIZ /NO CONTROLLER/
293	007260	116	117	NORDY:	.ASCIZ /NO DRIVE/
294	007276	116	117	ARLCS:	.ASCIZ /CS: /
295	007307	103	123	ARLBA:	.ASCIZ / BA: /
296	007314	040	102	ARLDA:	.ASCIZ / DA: /
297	007322	040	104	ARLMP:	.ASCIZ / MP: /
298	007330	040	115	BEREG:	.ASCIZ /BEFORE COMMAND: /
299	007336	102	105	AFREG:	.ASCIZ /TIME OF ERROR: /
300	007357	124	111	CRTIM:	.ASCIZ /CONTROLLER TIMED OUT/
301	007400	103	117	DRTIM:	.ASCIZ /DRIVE READY TIMED OUT/
302	007425	104	122	DEMES:	.ASCIZ / DRV/
303	007453	040	104	NXMMES:	.ASCIZ / NXM/
304	007460	040	116	OPIMES:	.ASCIZ / OPI/
305	007465	040	117	HCRCMES:	.ASCIZ / HCRC/
306	007472	040	110	HNFMES:	.ASCIZ / HNF/
307	007500	040	110	DCKMES:	.ASCIZ / DCK/
308	007505	040	104	DLTMES:	.ASCIZ / DLT/
309	007512	040	104	LF:	.ASCIZ <15>
310	007517	015	000	MSCRLF:	.ASCIZ <15><12>
311	007521	015	012	COMP:	.ASCIZ / COMP/
312	007524	040	103	OPIERR:	.ASCIZ /FORCED OPI(GET STATUS) CAUSED OTHER ERRORS/
313	007532	106	117	NOPMES:	.ASCIZ /NOOP OPERATION-FLAG MODE/
314	007605	116	117	NOPINT:	.ASCIZ /NOOP OPERATION-INTR. MODE/
315	007636	116	117	WCKMES:	.ASCIZ /WRITE CHECK OPERATION-FLAG MODE/
316	007670	127	122	WCKINT:	.ASCIZ /WRITE CHECK OPERATION-INTR. MODE/
317	007730	127	122	RHDMES:	.ASCIZ /READ HEADER OPERATION-FLAG MODE/
318	007771	122	105	RHDINT:	.ASCIZ /READ HEADER OPERATION-INTR. MODE/
319	010031	122	105	SEKMES:	.ASCIZ /SEEK OPERATION-FLAG MODE/
320	010072	123	105	SEKINT:	.ASCIZ /SEEK OPERATION-INTR. MODE/
321	010123	123	105	GSTMES:	.ASCIZ /GET STATUS OPERATION-FLAG MODE/
322	010155	107	105	GSTINT:	.ASCIZ /GET STATUS OPERATION-INTR MODE/
323	010214	107	105	RDDMES:	.ASCIZ /READ OPERATION-FLAG MODE/
324	010253	122	105	RDDINT:	.ASCIZ /READ OPERATION-INTR MODE/
325	010304	122	105	WRMES:	.ASCIZ /WRITE OPERATION-FLAG MODE/
326	010335	127	122	WRTINT:	.ASCIZ /WRITE OPERATION-INTR MODE/
327	010367	127	122	RDNMES:	.ASCIZ %READ W/O HEADER - FLAG MODE%
328	010421	122	105	RDNINT:	.ASCIZ %READ W/O HEADER - INTR MODE%
329	010455	122	105	SKHOME:	.ASCIZ /CAN'T SEEK TO TRACK 0/
330	010511	103	101	WRLOCK:	.ASCIZ /WRITE LOCK ERROR/
331	010537	127	122	EM1:	.ASCIZ /RLCS CONTAINED FOLLOWING ERROR(S): /
332	010560	122	114	EM100:	.BLKB 120.
333	010625			EM4:	.ASCIZ /NO INTERRUPT ON READ OPERATION/
334	011015	116	117	EM5:	.ASCIZ /READ OPERATION DID NOT WRITE MEMORY/
335	011054	122	105	EM6:	.ASCIZ /RLBA DID NOT INCREMENT PROPERLY DURING READ/
336	011120	122	114	EM7:	.ASCIZ /SECTOR DID NOT INCREMENT PROPERLY AFTER READ/
337	011174	123	105	EM10:	.ASCIZ /HEADER NOT FOUND COULD NOT BE FORCED/
338	011251	110	105	EM11:	.ASCIZ /WRONG CYLINDER ON SEEK/
339	011316	127	122	EM12:	.ASCIZ /HEADER NOT FOUND WOULD NOT SET/
340	011345	110	105	EM13:	.ASCIZ /DRIVE READY WOULD NOT SET/
341	011404	104	122	EM14:	.ASCIZ /DISK ADDRESS INCORRECT AFTER MULTIPLE SECTOR READ/
342	011436	104	111	EM16:	.ASCIZ /DRIVE ERROR ON WRITE OPERATION/
343	011520	104	122	EM17:	.ASCIZ /NO INTERRUPT ON WRITE OPERATION/
344	011557	116	117	EM20:	.ASCIZ /RLBA DID NOT INCREMENT PROPERLY DURING WRITE/
345	011617	122	114	EM21:	.ASCIZ /SECTOR DID NOT INCREMENT PROPERLY AFTER WRITE/
346	011674	123	105	EM22:	.ASCIZ /DISK ADDRESS (RLDA) INCORRECT AFTER MULTIPLE SECTOR WRITE/
347	011752	104	111		

348	012044	110	104	122	EM23:	.ASCIZ	/HDR NOT FND COULD NOT BE FORCED AT END OF TRACK/
349	012124	116	117	116	EM24:	.ASCIZ	/NON-EXISTANT MEMORY ERROR COULD NOT BE FORCED/
350	012202	104	101	124	EM25:	.ASCIZ	%DATA COMPARISON ERROR - READ/WRITE ERROR%
351							
352	012253	127	122	111	EM26:	.ASCIZ	/WRITE OPERATION MODIFIED MEMORY/
353	012313	105	122	122	EM27:	.ASCIZ	/ERROR ON PARTIAL SECTOR WRITE - ZERO FILL CHECK/
354	012373	122	114	102	EM30:	.ASCIZ	/RLBA DID NOT INCREMENT PROPERLY/
355	012433	102	101	040	EM31:	.ASCIZ	/BA BIT 16 DID NOT SET ON INCREMENT/
356	012476	102	101	040	EM32:	.ASCIZ	/BA BIT 17 SET ON BA16 INCREMENT TEST/
357	012543	122	114	102	EM33:	.ASCIZ	/RLBA DID NOT INCREMENT WITH BA16/
358	012604	102	101	040	EM34:	.ASCIZ	/BA BIT 17 DID NOT SET ON INCREMENT/
359	012647	102	101	040	EM35:	.ASCIZ	/BA BIT 16 DID NOT CLEAR ON INCREMENT/
360	012714	122	114	102	EM36:	.ASCIZ	/RLBA DID NOT INCREMENT WITH BA17/
361	012755	122	105	101	EM40:	.ASCIZ	/READ(FUNCTION 7) DID NOT INTERRUPT/
362	013020	122	105	101	EM41:	.ASCIZ	/READ(FUNCTION 7) ERROR - BAD DATA/
363	013062	122	105	101	EM42:	.ASCIZ	/READ(FUNCTION 7) ERROR AT END OF TRACK/
364	013131	116	117	040	EM43:	.ASCIZ	/NO INTERRUPT WITH HDR NT FND FORCED/
365	013175	116	117	040	EM44:	.ASCIZ	/NO INTERRUPT WITH NXM FORCED/
366	013232	105	122	122	EM45:	.ASCIZ	%ERROR ON BIT BANG OF SILO%
367	013264	123	111	114	EM47:	.ASCIZ	/SILO OPERATION FAILURE/
368	013313	110	105	101	EM50:	.ASCIZ	/HEADER COMPARE FAILURE - SECTOR/
369	013353	127	122	111	EM51:	.ASCIZ	/WRITE NPR CAUSED BUS TRAP/
370	013405	122	105	101	EM52:	.ASCIZ	/READ NPR CAUSED BUS TRAP/
371	013436	122	105	101	EM55:	.ASCIZ	?READ W/O HDR CMP OPERATION DID NOT WRITE MEMORY?
372	013516	122	114	102	EM53:	.ASCIZ	?RLBA DID NOT INCREMENT PROPERLY DURING READ W/O HDR CMP?
373	013606	122	114	104	EM54:	.ASCIZ	?RLDA DID NOT INCREMENT AFTER READ W/O HDR CMP?
374	013664	117	120	111	EM56:	.ASCIZ	/OPI TIMING ERROR/
375	013705	127	122	111	EM57:	.ASCIZ	/WRITE CHECK NPR CAUSED BUS TRAP/
376	013745	127	122	111	EM60:	.ASCIZ	/WRITE CHECK DID NOT INTERRUPT/
377	014003	122	114	102	EM61:	.ASCIZ	/RLBA DID NOT INCREMENT PROPERLY DURING WRCHK/
378	014060	122	114	104	EM62:	.ASCIZ	/RLDA DID NOT INCREMENT PROPERLY DURING WRCHK/
379	014135	122	114	104	EM63:	.ASCIZ	/RLDA DID NOT INCREMENT PROPERLY AFTER A MULTIPLE SECTOR WRITE CHK/
380	014237	127	122	111	EM64:	.ASCIZ	/WRITE CHECK OF PARTIAL SECTOR WRITE FAILURE/
381	014313	103	101	116	EM65:	.ASCIZ	/CAN NOT FORCE DCK ON WRITE CHECK/
382	014354	103	101	116	EM66:	.ASCIZ	/CAN NOT FORCE INTERRUPT WITH DCK ON WRCHK/
383	014426	127	122	111	EM70:	.ASCIZ	/WRITE CHECK FAILURE/
384							
385					.EVEN		
386							
387							
391	014452				ENDMOD		
392							
393	014452				BGNMOD	GLBERR	
394							
395					.SBTTL	GLOBAL ERRORS	
396	014452				BGNMSG	ERRO	
397							
398	014452	004737	015464		JSR	PC,LINE1	
399	014456	004737	015520		JSR	PC,LINE2	
400							
401							
402	014462	004537	020344		JSR	R5,CKERLT	;INCREMENT ERROR AND CHECK LIMIT
403							
404	014466				ENDMSG		
	014466				L10000:		
	014466	104023			EMT	C\$MSG	
405							

```
406 014470          BGNMSG  ERR1
407
408 014470  004737  015464          JSR    PC,LINE1
409
410
411 014474  004537  020344          JSR    R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
412
413 014500          ENDMSG
    014500          L10001:
    014500  104023          EMT    C$MSG
414
415 014502          BGNMSG  ERR2
416
417 014502  004737  015464          JSR    PC,LINE1
418 014506          PRINTB  #FRMT4,GDDAT,BDDAT
    014506  013746  002176          MOV    BDDAT,-(SP)
    014512  013746  002174          MOV    GDDAT,-(SP)
    014516  012746  016141          MOV    #FRMT4,-(SP)
    014522  012746  000003          MOV    #3,-(SP)
    014526  010600          MOV    SP,R0
    014530  104014          EMT    C$PNTB
    014532  062706  000010          ADD    #10,SP
419
420
421 014536  004537  020344          JSR    R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
422
423 014542          ENDMSG
    014542          L10002:
    014542  104023          EMT    C$MSG
424
425 014544          BGNMSG  ERR3
426
427 014544  004737  015464          JSR    PC,LINE1
428 014550  004737  015520          JSR    PC,LINE2
429 014554          PRINTB  #FRMT5,TMPO,BDDAT,GDDAT
    014554  013746  002174          MOV    GDDAT,-(SP)
    014560  013746  002176          MOV    BDDAT,-(SP)
    014564  013746  002166          MOV    TMPO,-(SP)
    014570  012746  016177          MOV    #FRMT5,-(SP)
    014574  012746  000004          MOV    #4,-(SP)
    014600  010600          MOV    SP,R0
    014602  104014          EMT    C$PNTB
    014604  062706  000012          ADD    #12,SP
430
431
432 014610  004537  020344          JSR    R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
433
434 014614          ENDMSG
    014614          L10003:
    014614  104023          EMT    C$MSG
435
436 014616          BGNMSG  ERR4
437
438 014616  004737  015464          JSR    PC,LINE1
439 014622  004737  015520          JSR    PC,LINE2
440 014626          PRINTB  #FRMT4,GDDAT,BDDAT
    014626  013746  002176          MOV    BDDAT,-(SP)
```

	014632	013746	002174	MOV	GDDAT,-(SP)	
	014636	012746	016141	MOV	#FRMT4,-(SP)	
	014642	012746	000003	MOV	#3,-(SP)	
	014646	010600		MOV	SP,R0	
	014650	104014		EMT	C\$PNTB	
	014652	062706	000010	ADD	#10,SP	
441						
442						
443	014656	004537	020344	JSR	R5,CKERLT	;INCREMENT ERROR AND CHECK LIMIT
444						
445	014662			ENDMSG		
	014662			L10004:		
	014662	104023		EMT	C\$MSG	
446						
447	014664			BGNMSG	ERR5	
448						
449	014664	004737	015464	JSR	PC,LINE1	
450	014670			PRINTB	#FRMT3,RESTMS	
	014670	013746	020656	MOV	RESTMS,-(SP)	
	014674	012746	016134	MOV	#FRMT3,-(SP)	
	014700	012746	000002	MOV	#2,-(SP)	
	014704	010600		MOV	SP,R0	
	014706	104014		EMT	C\$PNTB	
	014710	062706	000006	ADD	#6,SP	
451						
452						
453	014714	004537	020344	JSR	R5,CKERLT	;INCREMENT ERROR AND CHECK LIMIT
454						
455	014720			ENDMSG		
	014720			L10005:		
	014720	104023		EMT	C\$MSG	
456						
457	014722			BGNMSG	ERR6	
458						
459	014722	004737	015464	JSR	PC,LINE1	
460	014726	004737	015742	JSR	PC,LINE3	
461	014732	004737	015520	JSR	PC,LINE2	
462						
463						
464	014736			PRINTB	#FRMT99	
	014736	012746	017075	MOV	#FRMT99,-(SP)	
	014742	012746	000001	MOV	#1,-(SP)	
	014746	010600		MOV	SP,R0	
	014750	104014		EMT	C\$PNTB	
	014752	062706	000004	ADD	#4,SP	
465	014756	004537	020344	JSR	R5,CKERLT	;INCREMENT ERROR AND CHECK LIMIT
466						
467	014762			ENDMSG		
	014762			L10006:		
	014762	104023		EMT	C\$MSG	
468						
469	014764			BGNMSG	ERR7	
470						
471						
472						
473	014764	004537	020344	JSR	R5,CKERLT	;INCREMENT ERROR AND CHECK LIMIT
474						

475	014770				ENDMSG	
	014770			L10007:		
	014770	104023			EMT	C\$MSG
476						
477						
478						
479	014772			BGNMSG	ERR8	
480						
481	014772	004737	015464		JSR	PC,LINE1
482	014776	004737	015520		JSR	PC,LINE2
483	015002				PRINTB	#FRMT6,TMP1,GDDAT,BDDAT
	015002	013746	002176		MOV	BDDAT,-(SP)
	015006	013746	002174		MOV	GDDAT,-(SP)
	015012	013746	002170		MOV	TMP1,-(SP)
	015016	012746	016250		MOV	#FRMT6,-(SP)
	015022	012746	000004		MOV	#4,-(SP)
	015026	010600			MOV	SP,R0
	015030	104014			EMT	C\$PNTB
	015032	062706	000012		ADD	#12,SP
484						
485						
486	015036	004537	020344		JSR	R5,CKERLT ;INCREMENT ERROR AND CHECK LIMIT
487						
488	015042				ENDMSG	
	015042			L10010:		
	015042	104023			EMT	C\$MSG
489						
490	015044			BGNMSG	ERR9	
491						
492	015044	004737	015464		JSR	PC,LINE1
493	015050	004737	015520		JSR	PC,LINE2
494	015054				PRINTB	#FRMT4,TMPO,R2
	015054	010246			MOV	R2,-(SP)
	015056	013746	002166		MOV	TMPO,-(SP)
	015062	012746	016141		MOV	#FRMT4,-(SP)
	015066	012746	000003		MOV	#3,-(SP)
	015072	010600			MOV	SP,R0
	015074	104014			EMT	C\$PNTB
	015076	062706	000010		ADD	#10,SP
495						
496						
497	015102	004537	020344		JSR	R5,CKERLT ;INCREMENT ERROR AND CHECK LIMIT
498						
499	015106				ENDMSG	
	015106			L10011:		
	015106	104023			EMT	C\$MSG
500						
501	015110			BGNMSG	ERR10	
502						
503	015110	004737	015464		JSR	PC,LINE1
504	015114	004737	015520		JSR	PC,LINE2
505	015120				PRINTB	#FRMT7,TMP1,GDDAT,BDDAT
	015120	013746	002176		MOV	BDDAT,-(SP)
	015124	013746	002174		MOV	GDDAT,-(SP)
	015130	013746	002170		MOV	TMP1,-(SP)
	015134	012746	016325		MOV	#FRMT7,-(SP)
	015140	012746	000004		MOV	#4,-(SP)

	015144	010600		MOV	SP,R0	
	015146	104014		EMT	C\$PNTB	
	015150	062706	000012	ADD	#12,SP	
506						
507						
508	015154	004537	020344	JSR	R5,CKERLT	:INCREMENT ERROR AND CHECK LIMIT
509						
510	015160			ENDMSG		
	015160			L10012:		
	015160	104023		EMT	C\$MSG	
511				BGNMSG	ERR11	
512	015162					
513						
514	015162	004737	015464	JSR	PC,LINE1	
515	015166	004737	015520	JSR	PC,LINE2	
516	015172			PRINTB	#FRMT8,TMPO,GDDAT,BDDAT	
	015172	013746	002176	MOV	BDDAT,-(SP)	
	015176	013746	002174	MOV	GDDAT,-(SP)	
	015202	013746	002166	MOV	TMPO,-(SP)	
	015206	012746	016377	MOV	#FRMT8,-(SP)	
	015212	012746	000004	MOV	#4,-(SP)	
	015216	010600		MOV	SP,R0	
	015220	104014		EMT	C\$PNTB	
	015222	062706	000012	ADD	#12,SP	
517						
518						
519	015226	004537	020344	JSR	R5,CKERLT	:INCREMENT ERROR AND CHECK LIMIT
520						
521	015232			ENDMSG		
	015232			L10013:		
	015232	104023		EMT	C\$MSG	
522				BGNMSG	ERR12	
523	015234					
524						
525	015234	004737	015464	JSR	PC,LINE1	
526	015240	004737	015520	JSR	PC,LINE2	
527	015244			PRINTB	#FRMT9,TMP1,R3,GDDAT,BDDAT	
	015244	013746	002176	MOV	BDDAT,-(SP)	
	015250	013746	002174	MOV	GDDAT,-(SP)	
	015254	010346		MOV	R3,-(SP)	
	015256	013746	002170	MOV	TMP1,-(SP)	
	015262	012746	016520	MOV	#FRMT9,-(SP)	
	015266	012746	000005	MOV	#5,-(SP)	
	015272	010600		MOV	SP,R0	
	015274	104014		EMT	C\$PNTB	
	015276	062706	000014	ADD	#14,SP	
528						
529						
530	015302	004537	020344	JSR	R5,CKERLT	:INCREMENT ERROR AND CHECK LIMIT
531						
532	015306			ENDMSG		
	015306			L10014:		
	015306	104023		EMT	C\$MSG	
533				BGNMSG	ERR13	
534	015310					
535						
536	015310	004737	015464	JSR	PC,LINE1	

```

537 015314          PRINTB  #FRMT10,OPIMN,OPIMX,BDDAT
      015314 013746 002176  MOV      BDDAT,-(SP)
      015320 013746 002310  MOV      OPIMX,-(SP)
      015324 013746 002306  MOV      OPIMN,-(SP)
      015330 012746 016623  MOV      #FRMT10,-(SP)
      015334 012746 000004  MOV      #4,-(SP)
      015340 010600          MOV      SP,R0
      015342 104014          EMT      C$PNTB
      015344 062706 000012  ADD      #12,SP

538
539
540 015350 004537 020344          JSR      R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
541
542 015354          ENDMSG
      015354          L10015:
      015354 104023          EMT      C$MSG

543
544 015356          BGNMSG  ERR14
545
546 015356 004737 015464          JSR      PC,LINE1
547 015362 004737 015520          JSR      PC,LINE2
548 015366          PRINTB  #FRMT14,TMP1,#BUF
      015366 012746 003260  MOV      #BUF,-(SP)
      015372 013746 002170  MOV      TMP1,-(SP)
      015376 012746 016447  MOV      #FRMT14,-(SP)
      015402 012746 000003  MOV      #3,-(SP)
      015406 010600          MOV      SP,R0
      015410 104014          EMT      C$PNTB
      015412 062706 000010  ADD      #10,SP

549
550
551 015416 004537 020344          JSR      R5,CKERLT          ;INCREMENT ERROR AND CHECK LIMIT
552
553 015422          ENDMSG
      015422          L10016:
      015422 104023          EMT      C$MSG

554
555 015424          BGNMSG  ERR15
556
557 015424 004737 015464          JSR      PC,LINE1
558 015430 004737 015520          JSR      PC,LINE2
559 015434          PRINTB  #FRMT15,R2
      015434 010246  MOV      R2,-(SP)
      015436 012746 017131  MOV      #FRMT15,-(SP)
      015442 012746 000002  MOV      #2,-(SP)
      015446 010600          MOV      SP,R0
      015450 104014          EMT      C$PNTB
      015452 062706 000006  ADD      #6,SP
560 015456 004537 020344          JSR      R5,CKERLT
561
562 015462          ENDMSG
      015462          L10017:
      015462 104023          EMT      C$MSG

563
564 015464          LINE1: PRINTB  #FRMT11,RLCS,<B,DRIVE+1>
      015464 005046  CLR      -(SP)
      015466 153716 002143  BISB    DRIVE+1,(SP)

```

	015472	013746	002250	MOV	RLCS,-(SP)
	015476	012746	016014	MOV	#FRMT1,-(SP)
	015502	012746	000003	MOV	#3,-(SP)
	015506	010600		MOV	SP,R0
	015510	104014		EMT	C\$PNTB
	015512	062706	000010	ADD	#10,SP
565	015516	000207		RTS	PC
566					
567	015520			LINE2: PRINTB	#FRMT2,#BEREG,#ARLCS,B.CS,#ARLBA,B.BA
	015520	013746	002226	MOV	B.BA,-(SP)
	015524	012746	007314	MOV	#ARLBA,-(SP)
	015530	013746	002224	MOV	B.CS,-(SP)
	015534	012746	007307	MOV	#ARLCS,-(SP)
	015540	012746	007336	MOV	#BEREG,-(SP)
	015544	012746	016053	MOV	#FRMT2,-(SP)
	015550	012746	000006	MOV	#6,-(SP)
	015554	010600		MOV	SP,R0
	015556	104014		EMT	C\$PNTB
	015560	062706	000016	ADD	#16,SP
568	015564			PRINTB	#FRMT2A,#ARLDA,B.DA,#ARLMP,B.MP
	015564	013746	002232	MOV	B.MP,-(SP)
	015570	012746	007330	MOV	#ARLMP,-(SP)
	015574	013746	002230	MOV	B.DA,-(SP)
	015600	012746	007322	MOV	#ARLDA,-(SP)
	015604	012746	016072	MOV	#FRMT2A,-(SP)
	015610	012746	000005	MOV	#5,-(SP)
	015614	010600		MOV	SP,R0
	015616	104014		EMT	C\$PNTB
	015620	062706	000014	ADD	#14,SP
569	015624			PRINTB	#FRMT2,#AFREG,#ARLCS,E.CS,#ARLBA,E.BA
	015624	013746	002236	MOV	E.BA,-(SP)
	015630	012746	007314	MOV	#ARLBA,-(SP)
	015634	013746	002234	MOV	E.CS,-(SP)
	015640	012746	007307	MOV	#ARLCS,-(SP)
	015644	012746	007357	MOV	#AFREG,-(SP)
	015650	012746	016053	MOV	#FRMT2,-(SP)
	015654	012746	000006	MOV	#6,-(SP)
	015660	010600		MOV	SP,R0
	015662	104014		EMT	C\$PNTB
	015664	062706	000016	ADD	#16,SP
570	015670			PRINTB	#FRMT2B,#ARLDA,E.DA,#ARLMP,E.MP,E.MP1,E.MP2
	015670	013746	002246	MOV	E.MP2,-(SP)
	015674	013746	002244	MOV	E.MP1,-(SP)
	015700	013746	002242	MOV	E.MP,-(SP)
	015704	012746	007330	MOV	#ARLMP,-(SP)
	015710	013746	002240	MOV	E.DA,-(SP)
	015714	012746	007322	MOV	#ARLDA,-(SP)
	015720	012746	016105	MOV	#FRMT2B,-(SP)
	015724	012746	000007	MOV	#7,-(SP)
	015730	010600		MOV	SP,R0
	015732	104014		EMT	C\$PNTB
	015734	062706	000020	ADD	#20,SP
571	015740	000207		RTS	PC
572					
573	015742			LINE3: PRINTB	#FRMT3,#EM1
	015742	012746	010560	MOV	#EM1,-(SP)
	015746	012746	016134	MOV	#FRMT3,-(SP)

015752 012746 000002
015756 010600
015760 104014
015762 062706 000006
574 015766
015766 012746 010625
015772 012746 016134
015776 012746 000002
016002 010600
016004 104014
016006 062706 000006
575 016012 000207

MOV #2,-(SP)
MOV SP,R0
EMT C\$PNTB
ADD #6,SP
PRINTB #FRMT3,#EM100
MOV #EM100,-(SP)
MOV #FRMT3,-(SP)
MOV #2,-(SP)
MOV SP,R0
EMT C\$PNTB
ADD #6,SP
RTS PC

576
577
581

582 016014 045 101 103 FRMT1: .ASCIZ /%ACONTROLLER: %06%A DRIVE: %01/
583 016053 045 116 045 FRMT2: .ASCIZ /%N%I%T%06%T%06/
584 016072 045 124 045 FRMT2A: .ASCIZ /%T%06%T%06/
585 016105 045 124 045 FRMT2B: .ASCIZ /%T%06%T%06%A %06%A %06/
586 016134 045 116 045 FRMT3: .ASCIZ /%N%T/
587 016141 045 116 045 FRMT4: .ASCIZ /%N%AEXP'D: %06%A REC'D: %06%N/
588 016177 045 116 045 FRMT5: .ASCIZ /%N%ALAST: %06%A PRES: %06%A EXP'D: %06%N/
589 016250 045 116 045 FRMT6: .ASCIZ /%N%ABUS ADR: %06%A EXP'D: %06%A REC'D: %06%N/
590 016325 045 116 045 FRMT7: .ASCIZ /%N%AWORD: %D3%A EXP'D: %06%A REC'D: %06%N/
591 016377 045 116 045 FRMT8: .ASCIZ /%N%ADA: %06%A REC'D: %06%A EXP'D: %06%N/
592 016447 045 116 045 FRMT14: .ASCIZ /%N%AWORDS WRITTEN: %D3%A BUS ADDR: %06%N/
593 016520 045 116 045 FRMT9: .ASCIZ /%N%AWORDS WRITTEN: %D3%A BUS ADDR: %06%A EXP'D: %06%A REC'D: %06%N/
594 016623 045 116 045 FRMT10: .ASCIZ /%N%ARANGE %D3%A - %D3%A MILLISECONDS WAS %D6%N/
595 016702 045 101 115 .ASCIZ /%AMAXIMUM TIMEOUT OF PROGRAM IS 3 SECONDS%N/
596 016756 045 116 045 FRMT11: .ASCIZ /%N%AERROR LIMIT EXCEEDED - DROPPED%N/
597 017023 045 101 104 FRMT98: .ASCII /%ADRIE DID NOT RECOVER FROM POWER FAILURE/
598 017075 045 116 000 FRMT99: .ASCIZ /%N/
599 017100 045 116 045 FRMT13: .ASCIZ /%N%T%A - WILL NOT TEST%N/
600 017131 045 116 045 FRMT15: .ASCIZ /%N%APATTERN WAS: %06/

601
602
603
607

608 017156
609 017156
610
611 017156

ENDMOD
BGNMOD HPTCODE

017156 000006
612 017160 174400
613 017162 000160
614 017164 000240
615 017166 000001
616 017170 000000
617 017172 000001

BGNHW
.WORD L10020-L\$HW/2
.WORD 174400 :CSR
.WORD 160 :VECTOR
.WORD 240 :PRIORITY
.WORD 1 :TYPE OF DRIVE RL01 OR RL02
.WORD 0 :DRIVE (BITS 8,9,10)
.WORD 1 :RL11=1 RLV11=0

618
619 017174
017174

ENDHW
L10020:

620
621 017174
622
623 017174
624

ENDMOD
BGNMOD SPTCODE

625	017174		BGNSW		
	017174	000006		.WORD	L10021-L&SW/2
626					
627	017176	000000	DROP:	.WORD	0
628	017200	000012	MERLMT:	.WORD	10.
629	017202	000000	T.SIZE:	.WORD	0
630	017204	000000	T.DMP:	.WORD	0
631	017206	000000	T.LMT:	.WORD	0
632	017210	000001	T.ANS:	.WORD	1
633					
634	017212		ENDSW		
	017212		L10021:		
635					
636	017212		ENDMOD		
637					
638	017212		BGNMOD	DSPCODE	
639					
640	017212		DISPATCH		47
	017212	000057		.WORD	47
	017214	022050		.WORD	T1
	017216	022320		.WORD	T2
	017220	022464		.WORD	T3
	017222	022614		.WORD	T4
	017224	022750		.WORD	T5
	017226	023102		.WORD	T6
	017230	023240		.WORD	T7
	017232	023436		.WORD	T8
	017234	023740		.WORD	T9
	017236	024130		.WORD	T10
	017240	024326		.WORD	T11
	017242	024500		.WORD	T12
	017244	024676		.WORD	T13
	017246	025076		.WORD	T14
	017250	025246		.WORD	T15
	017252	025350		.WORD	T16
	017254	025474		.WORD	T17
	017256	025670		.WORD	T18
	017260	026024		.WORD	T19
	017262	026156		.WORD	T20
	017264	026276		.WORD	T21
	017266	026456		.WORD	T22
	017270	027270		.WORD	T23
	017272	027464		.WORD	T24
	017274	027630		.WORD	T25
	017276	027744		.WORD	T26
	017300	030122		.WORD	T27
	017302	030522		.WORD	T28
	017304	031144		.WORD	T29
	017306	031572		.WORD	T30
	017310	032252		.WORD	T31
	017312	032704		.WORD	T32
	017314	033314		.WORD	T33
	017316	033546		.WORD	T34
	017320	034036		.WORD	T35
	017322	034332		.WORD	T36
	017324	034624		.WORD	T37
	017326	035216		.WORD	T38

017330	035516			.WORD	T39
017332	036056			.WORD	T40
017334	036370			.WORD	T41
017336	036714			.WORD	T42
017340	037240			.WORD	T43
017342	037330			.WORD	T44
017344	037462			.WORD	T45
017346	037660			.WORD	T46
017350	040016			.WORD	T47
641	017352			ENDMOD	
642					
643				.SBTTL	INITIALIZATION CODE
644					
645	017352			BGNMOD	INITCODE
646					
647	017352				BGNINIT
648					
649	017352			SETPRI	#PRI07
	017352	012700	000340	MOV	#PRI07,RO
	017356	104041		EMT	C\$SPRI
650					
651	017360			READEf	#EF.PWR
	017360	012700	000034	MOV	#EF.PWR,RO
	017364	104050		EMT	C\$REFG
652	017366			BNCOMPLETE	NOPWR
	017366	103004		BCC	NOPWR
653	017370	013737	002012 002312	MOV	LSUNIT,PWRFLG
654	017376	000475		BR	CONT
655	017400			NOPWR:	READEf #EF.RESTART
	017400	012700	000037	MOV	#EF.RESTART,RO
	017404	104050		EMT	C\$REFG
656	017406			BCOMPLETE	START1
	017406	103404		BCS	START1
657	017410			READEf	#EF.START
	017410	012700	000040	MOV	#EF.START,RO
	017414	104050		EMT	C\$REFG

```

1 017416          BNCOMPLETE CONTINUET
  017416 103010   BCC CONTINUET
2
3 017420 012700 002322   START1: MOV #ERCOUNT,RO
4 017424 012701 000100   MOV #64.,R1
5 017430 005020   i$: CLR (RO)+
6 017432 0053C1   DEC R1
7 017434 001375   BNE 1$
8 017436 000407   BR START
9
10 017440          CONTINUE: READEF #EF.CONTINUE
  017440 012700 000036   MOV #EF.CONTINUE,RO
  017444 104050   EMT C$REFG
11 017446          BCOMPLETE CONT
  017446 103451   BCS CONT
12 017450 005737 002144   NXT: TST UUT ;DONE WITH ALL UNITS
13 017454 001011   BNE XXX ;NO
14 017456 012737 177777 002146   START: MOV #-1,UNITST
15 017464 013737 002012 002144   MOV L$UNIT,UUT
16 017472 012737 002320 002320   MOV #ERCOUNT-2,ERPOINT
17
18 017500 005237 002146   XXX: INC UNITST
19 017504 062737 000002 002320   ADD #2,ERPOINT
20 017512 005337 002144   DEC UUT
21 017516          REST: GPHARD UNITST,RO
  017516 013700 002146   MOV UNITST,RO
  017522 104042   EMT C$GPHRD
22 017524          BCOMPLETE 2$
  017524 103406   BCS 2$
23 017526 005737 002312   TST PWRFLG
24 017532 001746   BEQ NXT
25 017534 005337 002312   DEC PWRFLG
26 017540 000743   BR NXT
27 017542 012037 002260   2$: MOV (RO)+,BCSR ;GET BUS ADDRESS
28 017546 012037 002262   MOV (RO)+,BVEC ;GET VECTOR
29
30 017552 012037 002264   MOV (RO)+,BPRIOR ;GET PRIORITY
31 017556 012037 002126   MOV (RO)+,T.DRIVE ;GET TYPE OF DRIVE
32
33 017562 012037 002142   MOV (RO)+,DRIVE ;GET DRIVE
34 017566 012037 002314   MOV (RO)+,T.CNTRL ;GET CONTROLLER TYPE
35
36 017572 013700 002260   CONT: MOV BCSR,RO ;CREATE REGISTERS
37 017576 010037 002250   MOV RO,RLCS
38 017602 062700 000002   ADD #2,RO
39 017606 010037 002252   MOV RO,RLBA
40 017612 062700 000002   ADD #2,RO
41 017616 010037 002254   MOV RO,RLDA
42 017622 062700 000002   ADD #2,RO
43 017626 010037 002256   MOV RO,RLMP
44
45 017632 005737 002312   TST PWRFLG
46 017636 001064   BNE 5$
47 017640 005737 017202   TST T.SIZE ;DO WE WANT TO CHECK UNITS??
48 017644 001461   BEQ 5$ ;NO
49 017646 005037 002150   CLR TRPFLG ;CLEAR OUT TRAP INDICATOR
50 017652          SETVEC ERRVEC,#TRPHAN,#340 ;SETUP TO CATCH TIMEOUT

```

	017652	012746	000340		MOV	#340,-(SP)		
	017656	012746	021566		MOV	#TRPHAN,-(SP)		
	017662	013746	002140		MOV	ERRVEC,-(SP)		
	017666	012746	00C003		MOV	#3,-(SP)		
	017672	104037			EMT	C\$SVEC		
	017674	062706	000010		ADD	#10,SP		
51	017700	005777	162344		TST	@RLCS	;ACCESS CONTROLLER	
52	017704				CLRVEC	ERRVEC		
	017704	013700	002140		MOV	ERRVEC,R0		
	017710	104036			EMT	C\$CVEC		
53	017712	005737	002150		TST	TRPFLG	;DID TRAP OCCUR??	
54	017716	001404			BEQ	7\$;NO, CHECK DRIVE	
55	017720	012737	007260	002134	MOV	#NORES,WHY		
56	017726	000415			BR	8\$		
57								
58	017730	012777	000200	162312	7\$:	MOV	#200,@RLCS	;NOW CHECK DRIVE FOR READY
59	017736	053777	002142	162304		BIS	DRIVE,@RLCS	
60	017744	032777	000001	162276		BIT	#1,@RLCS	
61	017752	001016				BNE	5\$	
62	017754	012737	007276	002134		MOV	#NORDY,WHY	
63	017762				8\$:	PRINTB	#FRMT13,WHY	
	017762	013746	002134			MOV	WHY,-(SP)	
	017766	012746	017100			MOV	#FRMT13,-(SP)	
	017772	012746	000002			MOV	#2,-(SP)	
	017776	010600				MOV	SP,R0	
	020000	104014				EMT	C\$PNTB	
	020002	062706	000006			ADD	#6,SP	
64	020006	030444				BR	6\$	
65								
66	020010	005737	002312		5\$:	TST	PWRFLG	;POWER UP
67	020014	001451				BEQ	END	;NO
68	020016	012777	000200	162224		MOV	#200,@RLCS	
69	020024	053777	002142	162216		BIS	DRIVE,@RLCS	
70	020032	012701	000074			MOV	#60.,R1	
71	020036				3\$:	WAITMS	#10.	
	020036	012700	000012			MOV	#10.,R0	
	020042	104026				EMT	C\$WTM	
72	020044	032777	000001	162176		BIT	#1,@RLCS	
73	020052	001032				BNE	END	
74	020054	005301				DEC	R1	
75	020056	001367				BNE	3\$	
76								
77	020060					PRINTF	#FRMT99	
	020060	012746	017075			MOV	#FRMT99,-(SP)	
	020064	012746	000001			MOV	#1,-(SP)	
	020070	010600				MOV	SP,R0	
	020072	104017				EMT	C\$PNTF	
	020074	062706	000004			ADD	#4,SP	
78	020100					PRINTF	#FRMT98	
	020100	012746	017023			MOV	#FRMT98,-(SP)	
	020104	012746	000001			MOV	#1,-(SP)	
	020110	010600				MOV	SP,R0	
	020112	104017				EMT	C\$PNTF	
	020114	062706	000004			ADD	#4,SP	
79	020120	004737	015464		6\$:	JSR	PC,LINE1	
80	020124					DODU	UNITST	
	020124	013700	002146			MOV	UNITST,R0	

```

81 020130 104053          EMT      C$DODU
      020132          DOCLN
82 020132 104044          EMT      C$DCLN
      020134 000137 017450 JMP      NXT
83
84 020140 013737 002304 002306 END:    MOV      UOPIMN,OPIMN
85 020146 013737 002302 002310      MOV      UOPIMX,OPIMX
86 020154 005737 002314      TST      T.CNTRL          ;RL11??
87 020160 001006          BNE      1$              ;YES, THEN KEEP LIMITS SET
88 020162 013737 002300 002306      MOV      LOPIMN,OPIMN
89 020170 013737 002276 002310      MOV      LOPIMX,OPIMX
90 020176          1$:
91 020176          SETVEC   BVEC,#INTSRV,#340
      020176 012746 000340      MOV      #340,-(SP)
      020202 012746 020326      MOV      #INTSRV,-(SP)
      020206 013746 002262      MOV      BVEC,-(SP)
      020212 012746 000003      MOV      #3,-(SP)
      020216 104037          EMT      C$SVEC
      020220 062706 000010      ADD      #10,SP
92
93
94
95 020224          ENDINIT
      020224          L10022:
      020224 104011          EMT      C$INIT
96
97 020226          ENDMOD
98
99 020226          BGNMOD  CLNCODE
100
101 020226          BGNCLN
102
103
104 020226          SETVEC   ERRVEC,#TRPHAN,#340
      020226 012746 000340      MOV      #340,-(SP)
      020232 012746 021566      MOV      #TRPHAN,-(SP)
      020236 013746 002140      MOV      ERRVEC,-(SP)
      020242 012746 000003      MOV      #3,-(SP)
      020246 104037          EMT      C$SVEC
      020250 062706 000010      ADD      #10,SP
105 020254 032777 000200 161766 1$:  BIT      #CRDY,@RLCS
106 020262 001774          BEQ      1$
107
108 020264 042777 000100 161756      BIC      #INTEN,@RLCS
109
110 020272          CLRVEC  BVEC
      020272 013700 002262      MOV      BVEC,R0
      020276 104036          EMT      C$CVEC
111 020300 005737 002312      TST      PWRFLG
112 020304 001402          BEQ      2$
113 020306 005337 002312      DEC      PWRFLG
114 020312          2$:
      020312 013700 002140      CLRVEC  ERRVEC
      020316 104036          MOV      ERRVEC,R0
      020316          EMT      C$CVEC
115
116
117

```

```

118 020320          ENDCLN
      020320          L10023: EMT      C$CLEAN
      020320 104012
119
120 020322          ENDMOD
121
122 020322          BGNMOD  DRPCODE
123
124 020322          BGNDU
125
126 020322 000240   NOP
127
128 020324          ENDDU
      020324          L10024: EMT      C$DU
      020324 104055
129
130 020326          ENDMOD
131
132          .SBTTL  GLOBAL SUBROUTINES
133
134 020326          BGNMCD  GLBSUB
135
136 020326          BGNSRV
137 020326 005237 002152  INTSRV: INC      INIFLG      ;SET INTERRUPT OCCURANCE FLAG
138
139 020332          ENDSRV
      020332          L10025: RTI
      020332 000002
140
141          ;ROUTINE USED IN TIMING OPI
142
143 020334 005237 002152  TIMSRV: INC      INTFLG
144 020340          ABORTWAIT
      020340 104021      EMT      C$ABRT
145 020342 000002      RTI
146
147 020344 000240   CKERLT: NOP
148 020346          INLOOP
      020346 104020      EMT      C$INLP
149 020350          BCOMPLETE 99$
      020350 103427      BCS      99$
150
151 020352 005737 017176   TST      DROP
152 020356 001424      BEQ      99$
153 020360 005277 161734   INC      @ERPOINT
154 020364 027737 161730 017200  CMP      @ERPOINT, MERLMT
155 020372 002416      BLT      99$
156
157 020374          PRINTF #FRMT11
      020374 012746 016756  MOV      #FRMT11, -(SP)
      020400 012746 000001  MOV      #1, -(SP)
      020404 010600      MOV      SP, R0
      020406 104017      EMT      C$PNTF
      020410 062706 000004  ADD      #4, SP
158 020414 004737 015464   JSR      PC, LINE1
159 020420          DODU   UNITST ;DROP THIS UNIT
      020420 013700 002146  MOV      UNITST, R0

```

160 020424 104053
160 020426
020426 104044
161
162 020430
163 020430 000205
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180 020432 005037 002132
181 020436 032737 176000 002234
182 020444 001001
183 020446 000205
184 020450 012701 010625
185 020454 005737 002234
186 020460 100003
187 020462 004537 021170
188 020466 007524
189 020470 032737 040000 002234
190 020476 001405
191 020500 005237 002316
192 020504 004537 021170
193 020510 007453
194 020512 032737 020000 002234
195 020520 001403
196 020522 004537 021170
197 020526 007460
198 020530 032737 002000 002234
199 020536 001422
200 020540 004537 021170
201 020544 007465
202 020546 032737 004000 002234
203 020554 001403
204 020556 004537 021170
205 020562 007472
206 020564 032737 010000 002234
207 020572 001424
208 020574 004537 021170
209 020600 007500
210 020602 000420
211 020604 032737 004000 002234
212 020612 001405
213 020614 005237 002132
214 020620 004537 021170

```

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.
:*
:*ROUTINE USES R0,R1 AND PICKS HEADER FROM R3
:*
:*      CALL      JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
:*
:*
CHERR:  CLR      T.CRC
        BIT      #176000,E.CS      ;ANY ERROR BITS SET?
        BNE     2$                ;YES,FIND OUT WHICH
        RTS     R5                ;NO EXIT
2$:     MOV      #EM100,R1        ;GET START OF STRING
        TST     E.CS              ;IS COMPOSITE ERROR SET?(BETTER BE)
        BPL     99$              ;IT'S NOT SOMETHING IS WRONG
        JSR     R5,FIX            ;YES, PUT "COMP" IN STRING
        COMP    ,                ;"COMP"
99$:    BIT      #DERR,E.CS        ;DRIVE ERROR SET?
        BEQ     3$                ;NO, CONTINUE
        INC     DERFLG
        JSR     R5,FIX            ;YES, PUT "DRV" INTO STRING
        DEMES   ,                ;"DRV"
3$:     BIT      #NXM,E.CS        ;NON-EXISTENT MEMORY ERROR?
        BEQ     4$                ;NO, CONTINUE
        JSR     R5,FIX            ;YES, PUT "NXM" INTO STRING
        NXMMES ,                ;"NXM"
4$:     BIT      #OPI,E.CS        ;IS OPI SET?
        BEQ     6$                ;NO, GO CHECK BITS 11 & 12
        JSR     R5,FIX            ;PUT "OPI" INTO STRING
        OPIMES ,                ;"OPI"
5$:     BIT      #BIT11,E.CS      ;HEADERCRC ERROR?
        BEQ     5$                ;NO, GO CHECK HEADER NOT FOUND
        JSR     R5,FIX            ;GO PUT "HCRC" IN STRING
        HCRCMES ,                ;"HCRC"
6$:     BIT      #BIT12,E.CS      ;HEADER NOT FOUND?
        BEQ     8$                ;NO, GO PUT "CRLF" IN STRING
        JSR     R5,FIX            ;PUT "HNF" IN STRING
        HNFMES ,                ;"HNF"
7$:     BR       8$                ;PUT "CRLF" IN STRING
6$:     BIT      #BIT11,E.CS      ;DATA CRC ERROR?
        BEQ     7$                ;NO, GO CHECK DATA LATE
        INC     T.CRC
        JSR     R5,FIX            ;PUT "DCK" IN STRING
    
```



```

215 020624 007505          DCKMES          ;"DCK"
216 020626 032737 010000 002234 7$: BIT          #BIT12,E.CS  ;DATA LATE ERROR?
217 020634 001403          BEQ           8$          ;NO, GO PUT IN "CRLF"
218 020636 004537 021170  JSR           R5,FIX    ;PUT "DLT" IN STRING
219 020642 007512          DLTMS          ;"DLT"
220 020644 004537 021170  8$: JSR           R5,FIX    ;PUT "CRLF" INTO STRING
221 020650 007521          MSCRLF         ;"CRLF"
222 020652 004537 021170  JSR           R5,FIX    ;MOVE HEADER
223 020656 000000          RESTMS: .WORD 0      ;HEADER FROM TEST
224 020660 105011          CLR          (R1)      ;PUT TERMINATOR IN
225 020662          ERRDF          300.,LF,ERR6
          020662 104462          TRAP          T$ERCODE
          020664 000454          .WORD          300
          020666 007517          .WORD          LF
          020670 014722          .WORD          ERR6
226 020672 000205          RTS           R5          ;EXIT ROUTINE
227
228
229
230          .SBTTL LOAD RLCS
231          ;*****
232          ;* ROUTINE TO LOAD RLCS WITH FUNCTION TO BE PERFORMED
233          ;* CALL: JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
234          ;* .WORD ;BITS TO BE LOADED, FUNCTION
235          ;* ;AND INTR ENABLE ONLY
236          ;*
237 020674 032777 040000 161346 LDFUNC: BIT          #BIT14,@RLCS ;DRIVE ERROR SET
238 020702 001426          BEQ           5$
239 020704 017737 161344 002230  MOV          @RLDA,B.DA
240 020712 012777 000013 161334  MOV          #13,@RLDA
241 020720 012737 000200 002224  MOV          #200,B.CS
242 020726 053737 002142 002224  BIS          DRIVE,B.CS
243 020734 013777 002224 161306  MOV          B.CS,@RLCS
244 020742 032777 000200 161300 6$: BIT          #200,@RLCS
245 020750 001774          BEQ           6$
246 020752 013777 002230 161274  MOV          B.DA,@RLDA
247 020760 012537 002154  5$: MOV          (R5)+,LDCSR ;GET BITS TO LOAD
248 020764 010346          MOV          R3,-(SP) ;SAVE R3
249 020766 042737 177661 002154  BIC          #177661,LDCSR ;CLEAR ALL BUT FUNC & INTR EN
250 020774 013737 002154 002266  MOV          LDCSR,FNDFNC ;SAVE FUNCTION
251 021002 042737 000100 002266  BIC          #INTEN,FNDFNC ;ONLY FUNCTION
252 021010 012703 021130  MOV          #HDRLIST,R3 ;GET HEADER LIST
253 021014 006237 002266  ASR          FNDFNC ;ALIGN TO LEFT
254 021020 001404          BEQ           2$ ;IF EQUAL TO ZERO, SET R3
255 021022 022323          1$: CMP          (R3)+,(R3)+ ;BUMP R3 BY 4
256 021024 005337 002266  DEC          FNDFNC ;DEC FUNCTION
257 021030 001374          BNE          1$ ;FOUND IT? NO-GO BACK
258 021032 032737 000100 002154 2$: BIT          #INTEN,LDCSR ;YES, DO WE WANT FLAG OR INTR?
259 021040 001401          BEQ           3$ ;FLAG BRANCH
260 021042 005723          TST          (R3)+ ;INTR POINT TO THAT ONE
261 021044 011303          3$: MOV          (R3),R3 ;SET HEADER
262 021046 010337 020656  MOV          R3,RESTMS ;SET UP HEADER
263 021052 010337 002272  MOV          R3,TRYFNC ;SAVE HEADER FOR LATER
264 021056 053737 002270 002154  BIS          XMEM,LDCSR ;LOAD E.A. BITS
265 021064 005037 002270  CLR          XMEM ;CLEAR OUT THE BITS
266 021070 053737 002142 002154  BIS          DRIVE,LDCSR ;SELECT DRIVE
267 021076 052737 000200 002154  BIS          #200,LDCSR
    
```

```

268 021104 013777 002154 161136      MOV      LDCSR,@RLCS      ;LOAD FUNCTION
269 021112 004537 021202                JSR      R5,BEFORE      ;READ REGISTERS
270 021116 042777 000200 161124 4$:  BIC      #200,@RLCS     ;ISSUE COMMAND
271 021124 012603                MOV      (SP)+,R3       ;RESTORE R3
272 021126 000205                RTS      R5              ;EXIT
273
274
275

```

```

276 021130 007605      HDRLST: NOPMES
277 021132 007636      NOPINT
278 021134 007670      WCKMES
279 021136 007730      WCKINT
280 021140 010155      GSTMES
281 021142 010214      GSTINT
282 021144 010072      SEKMES
283 021146 010123      SEKINT
284 021150 007771      RHDMES
285 021152 010031      RHDINT
286 021154 010335      WRTMES
287 021156 010367      WRTINT
288 021160 010253      RDDMES
289 021162 010304      RDDINT
290 021164 010421      RDNMES
291 021166 010455      RDNINT
292
293

```

```

294 :*****
295 :*ROUTINE TO MOVE ASCII STRINGS
296 :*USES REGISTERS R1 - WHERE STRING IS BEING BUILT
297 :*
298 :*      CALL      JSR      R5,FIX
299 :*      .WORD      ;ADDRESS OF STRING TO MOVE
300 021170 012504      FIX:  MOV      (R5)+,R4      ;GET ADDRESS AND MOVE RETURN
301 021172 112421      1$:  MOVB     (R4)+,(R1)+     ;GET BYTE AND UPDATE
302 021174 001376      BNE     1$                ;WATCH 0 BYTE TERMINATOR
303 021176 105741      TSTB   -(R1)             ;BACK UP OVER ZERO BYTE
304 021200 000205      RTS      R5              ;EXIT
305
306

```

```

307 ;ROUTINE TO READ REGISTERS PRIOR TO OPERATION
308 ;CALL: JSR R5,BEFORE
309

```

```

310 021202 017737 161042 002224 BEFORE: MOV      @RLCS,B.CS      ;READ CS
311 021210 017737 161036 002226      MOV      @RLBA,B.BA      ;      BA
312 021216 017737 161032 002230      MOV      @RLDA,B.DA      ;      DA
313 021224 017737 161026 002232      MOV      @RLMP,B.MP      ;      MP
314 021232 000205      RTS      R5
315

```

```

316 ;ROUTINE TO READ REGISTERS AT TIME OF ERROR
317 ;CALL: JSR R5,AFTER
318

```

```

319 021234 017737 161010 002234 AFTER:  MOV      @RLCS,E.CS      ;READ CS
320 021242 017737 161004 002236      MOV      @RLBA,E.BA      ;      BA
321 021250 017737 161000 002240      MOV      @RLDA,E.DA      ;      DA
322 021256 017737 160774 002242      MOV      @RLMP,E.MP      ;      MP
323 021264 017737 160766 002244      MOV      @RLMP,E.MP1     ;      MP
324 021272 017737 160760 002246      MOV      @RLMP,E.MP2     ;      MP

```

```

325 021300 000205          RTS      R5
326
327
328 021302 010046          SIMBCC: MOV     R0,-(SP)      ;SAVE R0
329 021304 010146          MOV     R1,-(SP)      ;SAVE R1
330 021306 010246          MOV     R2,-(SP)      ;SAVE R2
331 021310 012537 002200    MOV     (R5)+,TEMP2    ;GET NUMBER OF BITS
332 021314 012537 002202    MOV     (R5)+,TEMP3    ;GET DATA FOR CRC CALCULATION
333 021320 012537 002204    MOV     (R5)+,TEMP4    ;GET STARTING CRC
334 021324 005037 002162    1$:   CLR     BCCFBK      ;
335 021330 013700 002204    MOV     TEMP4,R0      ;GET PRESENT CRC
336 021334 006037 002202    ROR     TEMP3          ;ROTATE NEW DATA
337 021340 005500          ADC     RC             ;MERGE NEW WITH OLD
338 021342 032700 000001    BIT     #1,R0         ;BIT 0 SET
339 021346 001402          BEQ     2$            ;IF NOT CONTINUE
340 021350 005137 002162    COM     BCCFBK        ;
341 021354 013700 002160    2$:   MOV     XPOLY,R0     ;GET CRC POLYNOMIAL (CRC-16)
342 021360 005100          COM     R0            ;COMPLIMENT POLYNOMIAL
343 021362 040037 002162    BIC     R0,BCCFBK
344 021366 000241          CLC                    ;CLEAR CARRY
345 021370 006037 002204    RCR     TEMP4
346 021374 013700 002162    MOV     BCCFBK,R0
347 021400 013701 002204    MOV     TEMP4,R1
348 021404 010102          MOV     R1,R2
349 021406 040100          BIC     R1,R0
350 021410 043702 002162    BIC     BCCFBK,R2
351 021414 050200          BIS     R2,R0
352 021416 043737 002160 002204 BIC     XPOLY,TEMP4
353 021424 050037 002204    BIS     R0,TEMP4
354 021430 005337 002200    DEC     TEMP2
355 021434 001333          BNE     1$
356
357 021436 013737 002204 002164    MOV     TEMP4,CALBCC
358 021444 012602          MOV     (SP)+,R2
359 021446 012601          MOV     (SP)+,R1
360 021450 012600          MOV     (SP)+,R0
361 021452 000205          RTS      R5          ;RETURN
362
363
364          ;ROUTINE TO WAIT FOR DRIVE READY
365
366
367
368
369 021454 012701 000144    WTD RDY: MOV     #100.,R1
370 021460 032777 000001 160562 1$:   BIT     #DRDY,@RLCS
371 021466 001011          BNE     2$
372
373 021470          WAITUS #20.
374 021470 012700 000024    MOV     #20.,R0
375 021474 104027          EMT     C$WTU
376 021476 005301          DEC     R1
377 021500 001367          BNE     1$
378
379 021502          ERRDF 200.,DRTIM,ERR5
380 021502 104462          TRAP   T$ERCODE
381 021504 000310          .WORD 200

```

```

021506 007425          .WORD  DRTIM
021510 014664          .WORD  ERR5
378
379 021512 00G205      2$:   RTS    R5
380
381          ;ROUTINE TO WAIT FOR CONTROLLER
382
383 021514 012701 000620 WTCRDY: MOV    #400.,R1
384 021520 032777 000200 160522 1$:   BIT    #CRDY,@RLCS
385 021526 001014          BNE    2$
386
387 021530          WAITUS #20.
      021530 012700 000024      MOV    #20.,R0
      021534 104027          EMT    C$WTU
388 021536 005301          DEC    R1
389 021540 001367          BNE    1$
390 021542 004537 021234      JSR    R5,AFTER
391
392 021546          ERRDF  100.,CRTIM,ERR5
      021546 104462          TRAP  T$ERCODE
      021550 000144          .WORD  100
      021552 007400          .WORD  CRTIM
      021554 014664          .WORD  ERR5
393 021556 000205          RTS    R5
394
395 021560 004537 021234      2$:   JSR    R5,AFTER
396 021564 000205          RTS    R5
397
398
399 021566 005237 002150      TRPHAN: INC   TRPFLG
400 021572 000002          RTI
401
402 021574          HDHOME:
403
404 021574          BGNSEG          ;%%START OF SEGMENT%%
      021574 104004          EMT    C$BSEG
405          ;ISSUE DRIVE RESET
406
407 021576 012737 000001 002274      MOV    #1,ERFLG          ;SET ERROR FLAG
408 021604 012777 000013 160442      MOV    #DRST!MK!GSBIT,@RLDA
409 021612 004537 020674          JSR    R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
410 021616 000004          GSTAT
411 021620 004537 021514          JSR    R5,WTCRDY
412 021624          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      021624 104010          EMT    C$ESCAPE
      021626 000216          .WORD  10000$-.
413 021630 004537 020432          JSR    R5,CHERR          ;CHECK CNTLR FOR ERRORS
414 021634          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      021634 104010          EMT    C$ESCAPE
      021636 000206          .WORD  10000$-.
415
416
417 021640 004537 020674          JSR    R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
418 021644 000010          RDHDR
419 021646          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      021646 104010          EMT    C$ESCAPE
      021650 000174          .WORD  10000$-.

```

```

420 021652 004537 021514      JSR      R5,WTCRDY
421 021656      ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
    021656 104010      EMT      C$ESCAPE
    021660 000164      .WORD   10000$-.
422
423 021662 004537 020432      JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
424 021666      ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
    021666 104010      EMT      C$ESCAPE
    021670 000154      .WORD   10000$-.
425
426 021672 013737 002242 002166  MOV      E.MP, TMPO      ;GET HEADER
427 021700 042737 000077 002166  BIC      #77, TMPO
428 021706 001424      BEQ      99$             ;SEEK IS NOT NECESSARY
429 021710 042737 000100 002166  BIC      #100, TMPO
430 021716 012777 000001 160330  MOV      #MK, @RLDA
431 021724 053777 002166 160322  BIS      TMPO, @RLDA    ;SET TO SEEK
                                ;SET IN DIFFERENCE
432
433 021732 004537 020674      JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
434 021736 000006      SEEK
435 021740 004537 021514      JSR      R5,WTCRDY
436 021744      ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
    021744 104010      EMT      C$ESCAPE
    021746 000076      .WORD   10000$-.
437
438 021750 004537 020432      JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
439 021754      ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
    021754 104010      EMT      C$ESCAPE
    021756 000066      .WORD   10000$-.
440
441 021760 004537 020674      JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
442 021764 000010      RDHDR
443 021766 004537 021514      JSR      R5,WTCRDY
444 021772      ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
    021772 104010      EMT      C$ESCAPE
    021774 000050      .WORD   10000$-.
445 021776 004537 020432      JSR      R5,CHERR
446 022002      ESCAPE  SEG
    022002 104010      EMT      C$ESCAPE
    022004 000040      .WORD   10000$-.
447
448 022006 013737 002242 002166  MOV      E.MP, TMPO      ;GET HEADER
449 022014 043737 002156 002166  BIC      SECMSK, TMPO    ;IGNORE SECTOR
450 022022 001404      BEQ      1$             ;ON ZERO
451
452 022024      ERRDF   400., SKHOME, ERRO ;CAN'T SEEK TO TRACK 0
    022024 104462      TRAP   T$ERCODE
    022026 000620      .WORD   400
    022030 010511      .WORD   SKHOME
    022032 014452      .WORD   ERRO
453
454 022034      ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
    022034 104010      EMT      C$ESCAPE
    022036 000006      .WORD   10000$-.
455
456 022040 005037 002274      CLR      ERFLG          ;INDICATE SUCCESS BACK TO MAIN PROGRAM
457
458

```

```

459 022044          ENDSEG          :%%END OF SEGMENT%%
      022044          10000$: EMT      C$ESEG
460 022044 104005
461 022046 000207          RTS      PC
462
463 022050          ENDMOD
464
465          .SBTTL  **TEST 1** - WRITE NPR INTEGRITY
466
467 022050          BGNTST          : **START OF TEST**
468
469 022050          STARS
      :*****
470          :CHECK THAT NPR WILL NOT INTERFERE WITH THE OPERATION OF THE
471          :UNIBUS. WE SET UP LOCATION 4 TO HANDLE THE TRAP IF IT HAPPENS.
472 022050          STARS
      :*****
473
474
475 022050 004737 021574          JSR      PC,HDHOME          :HEADS OVER TRACK 0
476 022054          CKERFG          :HEADS GO HOME OKAY
      022062 104032          EMT      C$EXIT
      022064 000232          .WORD   L10026-.
477
478 022066          BGNSEG          :%%START OF SEGMENT%%
      022066 104004          EMT      C$BSEG
479
480 022070          1$: SETVEC  ERRVEC,#TRPHAN,#340          :SET UP FOR TRAP
      022070 012746 000340          MOV      #340,-(SP)
      022074 012746 021566          MOV      #TRPHAN,-(SP)
      022100 013746 002140          MOV      ERRVEC,-(SP)
      022104 012746 000003          MOV      #3,-(SP)
      022110 104037          EMT      C$SVEC
      022112 062706 000010          ADD      #10,SP
481 022116 005037 002150          CLR      TRPFLG          :CLEAR TRAP OCCURANCE
482 022122 012777 003260 160122          MOV      #BUF,@RLBA          :BUS ADDRESS
483 022130 005077 160120          CLR      @RLDA          :LOAD DISK ADDRESS
484 022134 012777 177777 160114          MOV      #-1,@RLMP          :WORD COUNT OF 1
485 022142 005037 002174          CLR      GDDAT          :SET UP CSR TO LOAD
486 022146 013737 002142 002174          MOV      DRIVE,GDDAT          :SET IN DRIVE
487 022154 052737 000012 002174          BIS      #WRITE,GDDAT          :SET IN FUNCTION
488 022162 004537 021202          JSR      R5,BEFORE          :LOAD FOR ERROR PRINTOUT
489 022166 013737 002174 002224          MOV      GDDAT,B.CS          :SET IN COMMAND
490 022174 052737 000201 002224          BIS      #201,B.CS          :LOAD CRDY
491 022202 042737 002000 002224          BIC      #OPI,B.CS          :CLEAR (BIT 10)
492 022210 013777 002174 160032          MOV      GDDAT,@RLCS          :ISSUE WRITE
493 022216 012701 000144          MOV      #100.,R1          :WAIT FOR CRDY
494 022222 032777 000200 160020 5$: BIT      #CRDY,@RLCS          :NPR DONE
495 022230 001013          BNE      6$          :YES, 6$
496 022232          WAITUS  #20.          :WAIT A WHILE
      022232 012700 000024          MOV      #20.,R0
      022236 104027          EMT      C$WTU
497 022240 005301          DEC      R1          :A WHILE UP
498 022242 001367          BNE      5$          :NO, GO BACK
499
500 022244 004537 021234          JSR      R5,AFTER

```

```
501 022250          ERRDF  0.,CRTIM,ERR5          ;CONTROLLER TIMED OUT
    022250 104462    TRAP    T$ERCODE
    022252 000000    .WORD  0
    022254 007400    .WORD  CRTIM
    022256 014664    .WORD  ERR5
502 022260          6$: CLRVEC  ERRVEC          ;CLEAR VECTOR
    022260 013700 002140 MOV    ERRVEC,R0
    022264 104036    EMT    C$CVEC
503 022266          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
    022266 104010    EMT    C$ESCAPE
    022270 000024    .WORD  10001$-.
504
505 022272 005737 002150 TST    TRPFLG          ;DID TRAP OCCUR?
506 022276 001406          REQ    7$          ;NO
507 022300 004537 021234 JSR    R5,AFTER
508 022304          ERRSF  1.,EM51,ERRO          ;TRAP ON WRITE
    022304 104461    TRAP    T$ERCODE
    022306 000001    .WORD  1
    022310 013353    .WORD  EM51
    022312 014452    .WORD  ERRO
509 022314          7$:
510
511
512 022314          ENDSEG          ;%%END OF SEGMENT%%
    022314          10001$:
    022314 104005    EMT    C$ESEG
513
514 022316          ENDTST          ;**END OF TEST**
    022316          L10026:
    022316 104001    EMT    C$ETST
515
516          .SBTTL  **TEST 2** - WRITE FUNCTION
517
518 022320          BGNTST          ;**START OF TEST**
519
520
521
522 022320          STARS
    ;*****
523          ;CHECK OF WRITE LOGIC UNDER FLAG MODE, WE WILL FIRST ISSUE A
524          ;READ HEADER SO THAT WE DON'T WRITE ON THE BAD SECTOR
525          ;FILE TRACK. WE WILL WRITE A FULL SECTOR (128 WORDS) FROM
526          ;MEMORY (BUF). WE CHECK THAT NO ERRORS OCCUR. IF WE
527          ;HAVE A DRIVE ERROR WE WILL DO A "GET STATUS" TO SEE
528          ;IF WRITE PROTECT IS SET IF IT IS WE WILL ABORT THE
529          ;TEST. AN ERROR ON THE WRITE WILL LOOP ON JUST THE
530          ;WRITE PORTION. LOOP ON TEST WILL READ HEADER, SEFK (IF
531          ;NECESSARY) AND WRITE.
532 022320          STARS
    ;*****
533
534
535 022320 004737 021574 JSR    PC,HDHOME          ;HEADS OVER TRACK 0
536 022324          CKERFG          ;HEADS GO HOME OKAY
    022332 104032    EMT    C$EXIT
    022334 000126    .WORD  L10027-.
537
```

```

538 022336          BGNSEG          ;%%START OF SEGMENT%%
      022336 104004          EMT      C$BSEG
539
540 022340          3$:
541 022340 005077 157710          CLR      @RLDA          ;SET DISK ADDRESS
542 022344 012777 177600 157704  MOV      #-128.,@RLMP    ;WORD COUNT
543 022352 012777 003260 157672  MOV      #BUF,@RLBA     ;BUS ADDRESS
544 022360 004537 020674          JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
545 022364 000012          WRITE     ;WRITE
546
547 022366 004537 021514          JSR      R5,WTCRDY     ;WAIT FOR CONTROLLER READY
548 022372          ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      022372 104010          EMT      C$ESCAPE
      022374 000064          .WORD   10000$-.
549
550
551 022376 032777 040000 157644  BIT      #DERR,@RLCS    ;DRIVE ERROR SET?
552 022404 001425          BEQ      4$          ;BRANCH IF NOT
553
554 022406 012777 000003 157640  MOV      #MK!GSBIT,@RLDA ;SET GET STATUS OF DRIVE
555 022414 004537 020674          JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
556 022420 000004          GSTAT     ;GET STATUS
557 022422 004537 021514          JSR      R5,WTCRDY     ;WAIT FOR CONTROLLER READY
558 022426          ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      022426 104010          EMT      C$ESCAPE
      022430 000030          .WORD   10000$-.
559
560 022432 013737 002242 002174  MOV      E.MP,GDDAT     ;READ DRIVE STATUS
561 022440 032737 020000 002174  BIT      #BIT13,GDDAT   ;WRITE LOCK ERROR?
562 022446 001404          BEQ      4$          ;NO, BRANCH
563
564
565 022450          ERRSF   3.,WRLOCK,ERRO    ;WRITE LOCK ERROR
      022450 104461          TRAP    T$ERCODE
      022452 000003          .WORD   3
      022454 010537          .WORD   WRLOCK
      022456 014452          .WORD   ERRO
566 022460          4$:
567
568
569 022460          ENDSEG          ;%%END OF SEGMENT%%
      022460 10000$:
      022460 104005          EMT      C$ESEG
570 022462          ENDTST          ;**END OF TEST**
      022462 L10027:
      022462 104001          EMT      C$ETST
571
572          .SBTTL  **TEST 3** - WRITE FUNCTION INTERRUPT
573
574 022464          BGNST          ;**START OF TEST**
575
576 022464          STARS
      ;:*****
577          ;CHECK OF WRITE LOGIC UNDER INTERRUPT MODE, WE WILL ISSUE A
578          ;READ HEADER SO THAT WE DON'T WRITE ON THE BAD SECTOR FILE
579          ;TRACK. WE WILL WRITE A FULL SECTOR (128 WORDS) FROM MEMORY (BUF).
580          ;WE CHECK THAT NO ERRORS OCCUR. WE DO NOT CHECK RLDA OR RLBA
  
```



```

581                                     ;INCREMENT AT THIS TIME.
582 022464                               STARS
                                           ;:*****
583
584
585 022464 004737 021574                 JSR    PC,HDHOME           ;HEADS OVER TRACK 0
586 022470                                     CKERFG           ;HEADS GO HOME OKAY
      022476 104032                         EMT    C$EXIT
      022500 000112                         .WORD  L10030-.
587
588 022502                                     BGNSEG
      022502 104004                         EMT    C$BSEG           ;%%START OF SEGMENT%%
589
590
591 022504 005037 002152                 CLR    INTFLG           ;CLEAR INTERRUPT OCCURANCE FLAG
592 022510 005077 157540                 CLR    @RLDA
593 022514 012777 177600 157534         MOV    #-128.,@RLMP     ;SET UP WORD COUNT
594 022522 012777 003260 157522         MOV    #BUF,@RLBA      ;SET UP BUS ADDRESS
595
596 022530                                     SETPRI #PRI00           ;PRIORITY TO 0
      022530 012700 000000                 MCV   #PRI00,RO
      022534 104041                         EMT    C$SPRI
597 022536 004537 020674                 JSR    R5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
598 022542 000112                         WRITE!INTEN           ;WRITE UNDER INTERRUPT
599 022544 004537 021514                 JSR    R5,WTCRDY       ;WAIT FOR INTERRUPT
600 022550                                     ESCAPE SEG            ;CHECK FOR FL:LOE, ELSE EXIT SEG
      022550 104010                         EMT    C$ESCAPE
      022552 000036                         .WORD  10000$-.
601
602 022554                                     SETPRI #PRI07           ;SET PRIORITY TO 7
      022554 012700 000340                 MOV   #PRI07,RO
      022560 104041                         EMT    C$SPRI
603 022562 005737 002152                 TST   INTFLG           ;DID INTERRUPT OCCUR?
604 022566 001004                         BNE   2$              ;YES-BRANCH NO-REPORT
605
606 022570                                     ERRDF 4.,EM17,ERR0     ;WRITE DID NOT INTERRUPT
      022570 104462                         TRAP  T$ERCODE
      022572 000004                         .WORD 4
      022574 011557                         .WORD EM17
      022576 014452                         .WORD ERRO
607 022600                                     2$: ESCAPE SEG            ;CHECK FOR FL:LOE, ELSE EXIT SEG
      022600 104010                         EMT    C$ESCAPE
      022602 000006                         .WORD  10000$-.
608
609 022604 004537 020432                 JSR    R5,CHERR         ;CHECK CNTLR FOR ERRORS
610
611 022610                                     ENDSEG                ;%%END OF SEGMENT%%
      022610                                     10000$:
      022610 104005                         EMT    C$ESEG
612 022612                                     ENDTST                ;**END OF TEST**
      022612                                     L10030:
      022612 104001                         EMT    C$ETST
613
614                                     .SBTTL **TEST 4** - PROPER INCREMENT OF RLBA ON WRITE
615
616 022614                                     BGNTST                ;**START OF TEST**
617

```

```

618
619 022614                    STARS
                              ;*****
620                            ;CHECK THAT THE RLBA WILL INCREMENT PROPERLY AFTER THE
621                            ;WRITE WAS FINISHED THE RLBA SHOULD BE 128 WORDS (256 BYTES)
622                            ;CREATER. STARTING RLBA IS 'BUF', ENDING SHOULD BE 'BUF + 256.'"
623                            ;WE WILL MONITOR ALL ERRORS AND REPORT THEM ACCORDINGLY
624 022614                    STARS
                              ;*****
625
626
627 022614    004737    021574            JSR    PC,HDHOME            ;HEADS OVER TRACK 0
628 022620                                CKERFG                    ;HEADS GO HOME OKAY
                              EMT    C$EXIT
                              .WORD   L10031-.
629
630 022632                                BGNSEG                    ;%%START OF SEGMENT%%
                              EMT    C$BSEG
631
632 022634                                3$:
633 022634    005077    157414            CLR    @RLDA
634 022640    012777    003260    157404    MOV    #BUF,@RLBA            ;SET UP BUS ADDRESS
635 022646    012777    177600    157402    MOV    #-128.,@RLMP        ;WORD COUNT
636 022654    012737    003260    002174    MOV    #BUF,GDDAT        ;FORM EXPECTED BUS ADDRESS
637 022662    062737    000400    002174    ADD    #256.,GDDAT        ;AFTER WRITE
638
639 022670    004537    020674            JSR    R5,LDFUNC            ;LOAD THE FUNCTION IN NEXT WORD
640 022674    000012                                WRITE                    ;WRITE
641 022676    004537    021514            JSR    R5,WTCRDY            ;WAIT FOR CONTROLLER READY
642 022702                                ESCAPE                    ;CHECK FOR FL:LOE, ELSE EXIT SEG
                              EMT    C$ESCAPE
                              .WORD   10000$-.
643
644 022706    004537    020432            JSR    R5,CHERR            ;CHECK CNTLR FOR ERRORS
645 022712                                ESCAPE                    ;CHECK FOR FL:LOE, ELSE EXIT SEG
                              EMT    C$ESCAPE
                              .WORD   10000$-.
646 022716    017737    157330    002176    MOV    @RLBA,BDDAT        ;READ 'RLBA' FOR PRESENT ADDRESS
647 022724    023737    002176    002174    CMP    BDDAT,GDDAT        ;DID 'BA' INCREMENT PROPERLY?
648 022732    001404                                BEQ    2$                ;YES, CONTINUE
649
650 022734                                ERRDF                    ;BA DID NOT INCREMENT
                              TRAP   T$ERCODE
                              .WORD   5
                              .WORD   EM20
                              .WORD   ERR4
651
652 022744                                2$:
653
654 022744                                ENDSEG                    ;%%END OF SEGMENT%%
                              10000$:
                              EMT    C$ESEG
655 022746                                ENDTST                    ;**END OF TEST**
                              L10031:
                              EMT    C$ETST
656
657                                .SBTTL   **TEST 5** - PROPER INCREMENT OF RLDA ON WRITE
  
```

```

658
659 022750          BGNTST                      ;**START OF TEST**
660
661 022750          STARS
:*****
:CHECK THAT THE SECTOR INCREMENTS AFTER THE WRITE WAS FINISHED.
:WE RANDOMLY PICK A SECTOR (OTHER THAN LAST TRACK) AND ISSUE
:A FULL SECTOR WRITE THE RLDA SHOULD REFLECT AN INCREMENT
:OF THE SECOTR. "GDDAT" WAS THE EXPECTED RLDA.
STARS
:*****

667
668
669 022750 004737 021574      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
670 022754          CKERFG          ;HEADS GO HOME OKAY
      022762 104032          EMT      C$EXIT
      022764 000114          .WORD   L10032-.

671
672 022766          BGNSEG          ;%%START OF SEGMENT%%
      022766 104004          EMT      C$BSEG

673
674 022770          3$:
675 022770 005037 002174      CLR      GDDAT
676 022774 013777 002174 157252  MOV     GDDAT,@RLDA    ;SETUP DISK ADDRESS
677 023002 005237 002174      INC     GDDAT          ;CREATE EXPECTED SECTOR
678 023006 012777 177600 157242  MOV     #-128.,@RLMP   ;WORD COUNT
679 023014 012777 003260 157230  MOV     #BUF,@RLBA    ;SETUP BUS ADDRESS

680
681 023022 004537 020674      JSR     R5,LDFUNC     ;LOAD THE FUNCTION IN NEXT WORD
682 023026 0G0012          WRITE          ;WRITE
683 023030 004537 021514      JSR     R5,WTCRDY    ;WAIT FOR CONTROLLER READY
684 023034          ESCAPE SEG          ;CHECK FOR FL:LGE, ELSE EXIT SEG
      023034 104010          EMT      C$ESCAPE
      023036 000040          .WORD   10000$-.

685
686 023040 004537 020432      JSR     R5,CHERR     ;CHECK CNTLR FOR ERRORS
687 023044          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      023044 104010          EMT      C$ESCAPE
      023046 000030          .WORD   10000$-.

688
689 023050 013737 002240 002176  MOV     E.DA,BDDAT    ;READ DISK ADDRESS
690 023056 023737 002174 002176  CMP     GDDAT,BDDAT   ;DID SECTOR INCREMENT PROPERLY
691 023064 001404          BEQ     2$          ;YES, BRANCH NO, REPORT ERROR
692
693 023066          ERRDF 6.,EM21,ERR4 ;DA DID NOT INCREMENT
      023066 104462          TRAP   T$ERCODE
      023070 000006          .WORD   6
      023072 011674          .WORD   EM21
      023074 014616          .WORD   ERR4

694
695 023076          2$:
696
697 023076          ENDSEG          ;%%END OF SEGMENT%%
      023076          10000$:
      023076 104005          EMT      C$ESEG

698 023100          ENDTST          ;**END OF TEST**
      023100          L10032:

```

```

023100 104001          EMT      C$ETST
699
700          .SBTTL  **TEST 6** - FORCE HEADER NOT FOUND WITH WRITE
701
702 023102          BGNTST          ;**START OF TEST**
703
704 023102          STARS
          ;:*****
705          ;FORCE HEADER NOT FOUND ERROR TO OCCUR. THIS IS DONE
706          ;BY SETTING SECTOR 40 OF THE RLDA AND ISSUING A
707          ;WRITE. SECTOR 40 DOES NOT EXIST ON THE RL01 PACK
708          ;THEREFORE HDR NT FOUND SHOULD SET.
709 023102          STARS
          ;:*****
710
711 023102 004737 021574          JSR      PC,HDHOME          ;HEADS OVER TRACK 0
712 023106          CKERFG          ;HEADS GO HOME OKAY
          023114 104032          EMT      C$EXIT
          023116 000120          .WORD   L10033-.
713
714 023120          BGNSEG          ;%%START OF SEGMENT%%
          023120 104004          EMT      C$BSEG
715
716
717 023122 012777 000050 157124          MOV     #40.,@RLDA          ;INSURE NOT TO FIND HEADER BY
718 023130 012777 003260 157114          MOV     #BUF,@RLBA          ;SETTING SECTOR 40 OF CYL. ADDR.
719 023136 012777 177777 157112          MOV     #-1,@RLMP          ;WORD COUNT
720
721 023144 004537 020674          JSR     R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
722 023150 000012          WRITE          ;WRITE
723 023152 004537 021514          JSR     R5,WTCRDY          ;WAIT FOR CONTROLLER READY
724 023156          ESCAPE          ;CHECK FOR FL:LOE, ELSE EXIT SEG
          023156 104010          EMT      C$ESCAPE
          023160 000054          .WORD   10000$-.
725
726 023162 013737 002234 002166          MOV     E.CS, TMPO          ;GET RLCS
727 023170 042737 001777 002166          BIC     #1777, TMPO          ;SAVE ERROR BITS
728 023176 022737 112000 002166          CMP     #BIT15!BIT12!BIT10, TMPO ;HDR NOT FOUND SET.
729 023204 001402          BEQ     1$          ;YES, CONTINUE
730
731 023206 004537 020432          JSR     R5,CHERR
732 023212          1$:          CKLOOP
          023212 104006          EMT      C$CLP1
733
734 023214 022737 112000 002166          CMP     #BIT15!BIT12!BIT10, TMPO
735 023222 001404          BEQ     2$
736 023224          ERRDF          23., EM10, ERRO
          023224 104462          TRAP   T$ERCODE
          023226 000027          .WORD   23
          023230 011251          .WORD   EM10
          023232 014452          .WORD   ERRO
737
738 023234          2$:
739
740 023234          10000$:          ENDSEG          ;%%END OF SEGMENT%%
          023234          EMT      C$ESEG
          023234 104005

```

TEST 6 - FORCE HEADER NOT FOUND WITH WRITE

SEG 0068

```

741 023236          ENDTST          ;**END OF TEST**
      023236          L10033:
      023236 104001          EMT      C$ETST
742
743          .SBTTL  **TEST 7** - FORCE HEADER NOT FOUND WITH WRITE INTERRUPT
744
745 023240          BGNST          ;**START OF TEST**
746
747
748 023240          STARS
      ;*****
749          ;TEST THAT HEADER NOT FOUND ERROR WILL GENERATE AN INTERRUPT
750          ;ON OCCURANCE.  HEADER NOT FOUND WILL BE FORCED BY SETTING
751          ;SECTOR 40 OF RLDA AND ISSUING A WRITE
752 023240          STARS
      ;*****
753
754
755 023240 004737 021574          JSR      PC,HDHOME          ;HEADS OVER TRACK 0
756 023244          CKERFG          ;HEADS GO HOME OKAY
      023252 104032          EMT      C$EXIT
      023254 000160          .WORD   L10034-.
757
758 023256          BGNSEG          ;%%START OF SEGMENT%%
      023256 104004          EMT      C$BSEG
759
760 023260          SETPRI #PRI00
      023260 012700 000000          MOV     #PRI00,R0
      023264 104041          EMT      C$SPRI
761 023266 005037 002152          CLR     INTFLG          ;CLEAR INTERRUPT OCCURANCE FLAG
762 023272 012777 000050 156754          MOV     #40.,@RLDA          ;INSURE NOT TO FIND HEADER BY
763 023300 012777 003260 156744          MOV     #BUF,@RLBA          ;SETTING SECTOR 40 OF CYL. ADDR.
764 023306 012777 177777 156742          MOV     #-1,@RLMP          ;WORD COUNT
765
766 023314 004537 020674          JSR     R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
767 023320 000112          WRITE!INTEN          ;WRITE
768 023322 004537 021514          JSR     R5,WTCRDY          ;WAIT FOR CONTROLLER READY
769 023326          CKLOOP
      023326 104006          EMT      C$CLP1
770 023330          SETPRI #PRI07
      023330 012700 000340          MOV     #PRI07,R0
      023334 104041          EMT      C$SPRI
771
772 023336 005737 002152          TST     INTFLG          ;DID INTERRUPT OCCUR
773 023342 001004          BNE     2$          ;YES OKAY
774
775 023344          ERRDF 24.,EM43,ERRO          ;NO INTERRUPT FROM OPI
      023344 104462          TRAP   T$ERCODE
      023346 000030          .WORD   24
      023350 013131          .WORD   EM43
      023352 014452          .WORD   ERRO
776
777 023354          2$: ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      023354 104010          EMT      C$ESCAPE
      023356 000054          .WORD   10000$-.
778
779

```

```

780 023360 013737 002234 002166      MOV      E.CS, TMPO      ;GET RLCS
781 023366 042737 001777 002166      BIC      #1777, TMPO     ;SAVE ERROR BITS
782 023374 022737 112000 002166      CMP      #BIT15:BIT12:BIT10, TMPO ;WDR NOT FOUND SET.
783 023402 001402                BEQ      1$              ;YES, CONTINUE
784
785 023404 004537 020432                JSR      R5, CHERR
786 023410                1$:    CKLOOP
      023410 104006                EMT      C$CLP1
787
788 023412 022737 112000 002166      CMP      #BIT15:BIT12:BIT10, TMPO
789 023420 001404                BEQ      3$
790 023422                ERRDF   25., EM10, ERRO
      023422 104462                TRAP    T$ERCODE
      023424 000031                .WORD  25
      023426 011251                .WORD  EM10
      023430 014452                .WORD  ERRO
791
792 023432                3$:
793
794 023432                ENDSEG                    ;%%END OF SEGMENT%%
      023432                10000$:
      023432 104005                EMT      C$ESEG
795 023434                ENDTST                    ;**END OF TEST**
      023434                L10034:
      023434 104001                EMT      C$ETST
796
797
798
799                .SBTTL  **TEST 8** - CHECK OPI TIME WITH HDR NT FND
800
801 023436                BGNST                    ;**START OF TEST**
802
803 023436                STARS
      ;*****
804                ;CHECK OPI TIME IT SHOULD BE AROUND 200 MILLISECONDS (ON UNIBUS)
805                ;CHECK THIS BY TIMING OPI ON A FORCED HEADER NOT FOUND
806                ;ISSUE WRITE WITH SECTOR 40 SET IN THE DISK ADDRESS
807 023436                STARS
      ;*****
808
809 023436 004737 021574                JSR      PC, HDHOME      ;HEADS OVER TRACK 0
810 023442                CKERFG                    ;HEADS GO HOME OKAY
      023450 104032                EMT      C$EXIT
      023452 000264                .WORD  L10035-.
811
812 023454                BGNSEG                    ;%%START OF SEGMENT%%
      023454 104004                EMT      C$BSEG
813
814 023456                CLRVEC                    ;CLEAR PRESENT INTERRUPT VECTOR
      023456 013700 002262                MOV      BVEC, R0
      023462 104036                EMT      C$CVEC
815 023464                SETVEC                    ;SET INTR. VEC. WITH ABORT WAIT
      023464 012746 000340                MOV      BVEC, #TIMSRV, #340
      023470 012746 020334                MOV      #340, -(SP)
      023474 013746 002262                MOV      #TIMSRV, -(SP)
      023500 012746 000003                MOV      BVEC, -(SP)
      023504 104037                MOV      #3, -(SP)
      EMT      C$SVEC
  
```

```

816 023506 062706 000010      ADD      #10,SP
      023512                    SETPRI   #PRI00
      023512 012700 000000      MOV      #PRI00,RO
      023516 104041            EMT      C$SPRI
817 023520 005037 002152      CLR      INTFLG          ;CLEAR INTERRUPT FLAG
818 023524 012777 000050 156522  MOV      #40.,@RLDA      ;SET UP FOR HDR NT FND
819 023532 012777 003260 156512  MOV      #BUF,@RLBA      ;BUS ADDRESS
820 023540 012777 177777 156510  MOV      #-1,@RLMP       ;WORD COUNT
821
822 023546 004537 020674      JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
823 023552 000112      WRITE!INTEN
824
825 023554 013700 002310      MOV      OPIMX,RO
826 023560 006300      ASL      RO
827 023562 006300      ASL      RO
828 023564 006300      ASL      RO
829 023566 063700 002310      ADD      OPIMX,RO
830 023572 063700 002310      ADD      OPIMX,RO
831 023576                    WAITUS   RO          ;WAIT MAX MILLISECONDS
      023576 104027            EMT      C$WTU
832 023600 010037 002176      MOV      RO,BDDAT        ;SETUP FOR WORST CASE
833 023604 005737 002152      TST     INTFLG          ;DID INTERRUPT OCCUR
834 023610 001427            BEQ     4$              ;NO, REPORT ERROR
835
836 023612                    GETTIM   BDDAT        ;GET TIME EXPIRED
      023612 104052            EMT      C$GTIM
      023614 010037 002176      MOV      RO,BDDAT
837 023620 005000            CLR     RO          ;DIVIDE
838 023622 162737 000012 002176 1$:  SUB     #10.,BDDAT      ;ANSWER
839 023630 100402            BMI     3$              ;BY 10 TO GET
840 023632 005200            INC     RO          ;RIGHT ANSWER
841 023634 000772            BR      1$
842 023636 010037 002176      3$:  MOV     RO,BDDAT
843
844                    ;CHECK THAT OPI TIME IS WITHIN LIMITS
845
846 023642                    2$:  SETPRI   #PRI07
      023642 012700 000340      MOV      #PRI07,RO
      023646 104041            EMT      C$SPRI
847 023650 023737 002310 002176  CMP     OPIMX,BDDAT      ;IS IT WITHIN LIMITS
848 023656 002404            BLT     4$              ;NO, REPORT ERROR
849
850 023660 023737 002306 002176  CMP     OPIMN,BDDAT      ;WITHIN LIMITS
851 023666 003404            BLE     5$              ;YES
852
853 023670                    4$:  ERRDF   974.,EM56,ERR13 ;OPI TIMING INCORRECT
      023670 104462            TRAP    T$ERCODE
      023672 001716            .WORD   974
      023674 013664            .WORD   EM56
      023676 015310            .WORD   ERR13
854
855 023700                    5$:  CLRVEC   BVEC          ;CLEAR PRESENT VECTOR
      023700 013700 002262      MOV      BVEC,RO
      023704 104036            EMT      C$CVEC
856 023706                    SETVEC   BVEC,#INTSRV,#340 ;SET IN OLD VECTOR
      023706 012746 000340      MOV      #340,-(SP)
      023712 012746 020326      MOV      #INTSRV,-(SP)
  
```

```

023716 013746 002262      MOV      BVEC,-(SP)
023722 012746 000003      MOV      #3,-(SP)
023726 104037      EMT      C$SVEC
023730 062706 000010      ADD      #10,SP
857
858 023734      ENDSEG      ;%%END OF SEGMENT%%
      023734      10000$:
      023734 104005      EMT      C$ESEG
859
860 023736      ENDTST      ;**END OF TEST**
      023736      L10035:
      023736 104001      EMT      C$ETST
861
862      .SBTTL  **TEST 9** - MULTIPLE SECTOR TRANSFER ON WRITE
863
864 023740      BGNST      ;**START OF TEST**
865
866 023740      STARS
      ;:*****
      ;CHECK FOR MULTIPLE SECTOR TRANSFER ON WRITE. THIS TEST CHECKS
      ;THAT TWO SECTORS CAN BE SUCCESSFULLY WRITTEN. WE LOAD
      ;A WORD COUNT OF 129 WORDS (ONE SECTOR + 1 WORD) STARTING AT
      ;SECTOR 0 THRU SECTOR 37 AND VERIFY THAT THE RLDA DOES
      ;A DOUBLE INCREMENT EACH TIME.
      ;:*****
867
868
869
870
871
872 023740      STARS
      ;:*****
873
874
875
876 023740 004737 021574      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
877 023744      CKERFG      ;HEADS GO HOME OKAY
      023752 104032      EMT      C$EXIT
      023754 000152      .WORD    L10036-.
878
879 023756 005037 002166      CLR      TMPO      ;CLEAR TEMP LOCATIONS
880 023762 005037 002170      CLR      TMP1
881
882 023766      BGNSEG      ;%%START OF SEGMENT%%
      023766 104004      EMT      C$BSEG
883
884
885 023770 013737 002170 002174 1$:      MOV      TMP1,GDDAT      ;GET CYLINDER
886 023776 053737 002166 002174      BIS      TMPO,GDDAT      ;GET SECTOR
887 024004 013777 002174 156242      MOV      GDDAT,@RLDA      ;SET DISK ADDRESS-SECTOR 0
888 024012 062737 000002 002174      ADD      #2,GDDAT      ;SET EXPECTED + 2
889 024020 012777 003260 156224      MOV      #BUF,@RLBA      ;SET BUS ADDRESS
890 024026 012777 177577 156222      MOV      #-129.,@RLMP      ;WORD COUNT-SECTOR+1 WORD
891
892 024034 004537 020674      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
893 024040 000012      WRITE      ;WRITE
894 024042 004537 021514      JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY?
895 024046      ESCAPE      SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      024046 104010      EMT      C$ESCAPE
      024050 000054      .WORD    10000$-.
896
897 024052 004537 020432      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
898 024056      ESCAPE      SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
  
```



```

024056 104010      EMT      C$ESCAPE
024060 000044      .WORD    10000$-.
899
900 024062 013737 002240 002176      MOV      E.DA,BDDAT      ;READ DISK ADDRESS
901 024070 023737 002176 002174      CMP      BDDAT,GDDAT    ;IS DISK ADDRESS CORRECT
902 024076 001404      BEQ      2$              ;YES, BRANCH NO, REPORT ERROR
903
904 024100      ERRDF   7.,EM22,ERR4    ;DISK ADDRESS NOT CORRECT
    024100 104462      TRAP    T$ERCODE
    024102 000007      .WORD    7
    024104 011752      .WORD    EM22
    024106 014616      .WORD    ERR4
905
906 024110      2$:
907
908 024110 005237 002166 002166      INC      TMO0            ;NEXT SECTOR
909 024114 022737 000046 002166      CMP      #46,TMO0       ;AT END?
910 024122 001322      BNE     1$              ;NO, GO BACK
911
912 024124      ENDSEG                      ;%%END OF SEGMENT%%
    024124      10000$:
    024124 104005      EMT      C$ESEG
913 024126      ENDTST                      ;**END OF TEST**
    024126 L10036:
    024126 104001      EMT      C$ETST
914
915      .SBTTL  **TEST 10** - CHECK DIRECTION OF WRITE NPR
916
917 024130      B$NTST                      ;**START OF TEST**
918
919 024130      STARS
    :*****
    :VERIFY THAT A WRITE IS WRITING NOT READING. WE WRITE A
    :KNOWN PATTERN IN "BUF" (128 WORD), WE THEN ISSUE A WRITE.
    :ONCE THE WRITE IS FINISHED WE CHECK THAT "BUF" IS INTACT.
    :THIS IS DONE TO PROVE THAT THE NPR IS GOING THE RIGHT
    :WAY.
    :STARS
    :*****
920
921
922
923
924
925 024130
926
927
928 024130 004737 021574      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
929 024134      CKERFG                      ;HEADS GO HOME OKAY
    024142 104032      EMT      C$EXIT
    024144 000160      .WORD    L10037-.
930
931 024146      BGNSEG                      ;%%START OF SEGMENT%%
    024146 104004      EMT      C$BSEG
932
933 024150      2$:
934 024150 012702 003260      MOV      #BUF,R2        ;WRITE BUFFER FOR WRITE OPERATION
935 024154 012701 000200      MOV      #128,R1        ;ONE SECTOR'S WORTH
936 024160 012722 125252      MOV      #125252,(R2)+  ;WRITE BUFFER
937 024164 005301      DEC     R1              ;DONE?
938 024166 001374      BNE     3$              ;NO, GO BACK
939
940 024170 005077 156060      CLR     @RLDA           ;LOAD DISK ADDRESS
  
```

```

941 024174 012777 177600 156054      MOV    #-128.,@RLMP      ;WORD COUNT
942 024202 012777 003260 156042      MOV    #BUF,@RLBA      ;BUS ADDRESS
943 024210 004537 020674              JSR    R5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
944 024214 000012              WRITE                   ;WRITE SOME DATA
945 024216 004537 021514              JSR    R5,WTCRDY        ;WAIT FOR IT TO FINISH
946 024222              ESCAPE SEG              ;CHECK FOR FL:LOE, ELSE EXIT SEG
    024222 104010      EMT    C$ESCAPE
    024224 000076      .WORD  10000$-.
947
948 024226 004537 020432              JSR    R5,CHERR          ;CHECK CNTLR FOR ERRORS
949 024232              ESCAPE SEG              ;CHECK FOR FL:LOE, ELSE EXIT SEG
    024232 104010      EMT    C$ESCAPE
    024234 000066      .WORD  10000$-.
950
951 024236 012702 003260      MOV    #BUF,R2          ;SET UP TO CHECK BUFFER
952 024242 012701 000200      MOV    #128.,R1        ;CHECK 128 WORDS
953
954 024246              BGNSEG                   ;%%START OF SEGMENT%%
    024246 104004      EMT    C$BSEG
955
956 024250 012737 125252 002174      MOV    #125252,GDDAT    ;DATA SHOULD BE 125252
957 024256 011237 002176 4$:      MOV    (R2),BDDAT       ;LOAD DATA INTO BDDAT
958 024262 023737 002174 002176      CMP    GDDAT,BDDAT     ;IS IT OKAY?
959 024270 001406      BEQ    5$              ;YES, CONTINUE
960
961 024272 010237 002170      MOV    R2,TMP1 ;LOAD MEMORY LOCATION OF FAILURE
962 024276              ERRDF 8.,EM26,ERR8
    024276 104462      TRAP  T$ERCODE
    024300 000010      .WORD  8
    024302 012253      .WORD  EM26
    024304 014772      .WORD  ERR8
963
964 024306 5$:      ESCAPE SEG              ;CHECK FOR FL:LOE, ELSE EXIT SEG
    024306 104010      EMT    C$ESCAPE
    024310 000010      .WORD  10001$-.
965 024312 005722 6$:      TST   (R2)+             ;NEXT!
966 024314 005301              DEC    R1               ;DONE?
967 024316 001357              BNE   4$               ;NO, GO BACK
968
969 024320              ENDSEG                   ;%%END OF SEGMENT%%
    024320 10001$:      EMT    C$ESEG
    024320 104005      ENDSEG                   ;%%END OF SEGMENT%%
970 024322 10000$:      EMT    C$ESEG
    024322 104005      ENDSEG                   ;%%END OF TEST%%
971 024324 104001      ENDTST L10037:        ;**END OF TEST**
    024324 104001      EMT    C$ETST
972
973      .SBTTL **TEST 11** - CHECK FULL RLBA INCREMENT
974
975 024326      BGNTST                   ;**START OF TEST**
976
977 024326      STARS
    :*****
978      ;TEST THAT THE RLBA WILL INCREMENT, WE DO NOT DO A FULL 16
979      ;BIT INCREMENT WE CHECK THAT EACH BIT WILL TOGGLE 0 TO 1
  
```

```

980 ;AND 1 TO 0. WE DO CHECK ALL BITS EVEN IF ALL MEMORY
981 ;IS NOT AVAILABLE. (WE IGNORE NON-EXISTANT MEMORY ERRORS).
982 ;WE USE THE SAME DISK ADDRESS (RANDOM) AND A 1 WORD TRANSFER.
983 024326 STARS
;*****

984
985
986 024326 004737 021574 JSR PC,HDHOME ;HEADS OVER TRACK 0
987 024332 CKERFG ;HEADS GO HOME OKAY
024340 104032 EMT C$EXIT
024342 000134 .WORD L10040-.

988
989
990 024344 005037 002170 CLR TMP1 ;CLEAR LOCATION
991
992 024350 BGNSEG ;%%START OF SEGMENT%%
024350 104004 EMT C$BSEG

993
994 024352 3$:
995 024352 012777 177777 155676 MOV #-1,@RLMP ;ONLY ONE (1) WORD
996 024360 005077 155670 CLR @RLDA ;LOAD DISK ADDRESS
997 024364 013777 002170 155660 MOV TMP1,@RLBA ;BUS ADDRESS
998
999 024372 004537 020674 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
1000 024376 000012 WRITE
1001 024400 004537 021514 JSR R5,WTCRDY ;WAIT FOR WRITE TO FINISH
1002 024404 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
024404 104010 EMT C$ESCAPE
024406 000066 .WORD 10000$-.

1003
1004 024410 013737 002170 002174 4$: MOV TMP1,GDDAT ;SET UP EXPECTED RLBA
1005 024416 062737 000002 002174 ADD #2,GDDAT ;PREVIOUS RLBA+2
1006 024424 013737 002236 002176 MOV E.BA,BDDAT ;READ RLBA
1007 024432 023737 002174 002176 CMP GDDAT,BDDAT ;WAS IT UPDATED PROPERLY?
1008 024440 001404 BEQ 5$ ;YES, CONTINUE
1009
1010 024442 ERRDF 9.,EM30,ERR4 ;BA INCREMENT ERROR
024442 104462 TRAP T$ERCODE
024444 000011 .WORD 9
024446 012373 .WORD EM30
024450 014616 .WORD ERR4
1011 024452 5$: ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
024452 104010 EMT C$ESCAPE
024454 000020 .WORD 10000$-.

1012
1013 024456 006337 002170 ASL TMP1 ;NEXT PATTERN TO TEST RLBA
1014 024462 103404 BCS 6$ ;DONE?
1015 024464 052737 000002 002170 BIS #BIT1,TMP1 ;NO, SET IN BIT 1
1016 024472 000727 BR 3$ ;GO CHECK NEXT.
1017
1018 024474 6$: ;END TEST
1019
1020 024474 ENDSEG ;%%END OF SEGMENT%%
024474 10000$:
024474 104005 EMT C$ESEG

1021 024476 ENDTST ;**END OF TEST**
024476 L10040:

```

```

024476 104001          EMT      C$ETST
1022
1023          .SBTTL  **TEST 12** - BA BIT 16 INCREMENT
1024
1025 024500          BGNTST          ;**START OF TEST**
1026
1027 024500          STARS
1028          ;*****
1029          ;CHECK THAT BA BIT 16 WILL INCREMENT. WE WILL LOAD THE
1030          ;RLBA WITH 177776 AND ISSUE A ONE WORD WRITE WE THEN
1031          ;CHECK BA BIT 16 TO SET, BA 17 TO STAY A 0 AND THE RLBA
1032 024500          ;TO GO TO ZERO
          STARS
          ;*****
1033
1034
1035 024500 004737 021574      JSR      PC,HDHOME          ;HEADS OVER TRACK 0
1036 024504          CKERFG          ;HEADS GO HOME OKAY
          024512 104032      EMT      C$EXIT
          024514 000160      .WORD   L10041-.
1037
1038 024516          BGNSEG          ;%%START OF SEGMENT%%
          024516 104004      EMT      C$BSEG
1039
1040 024520          2$:
1041 024520 012777 177776 155524      MOV      #177776,@RLBA      ;SET MAX BA TO INC. BA16
1042 024526 005037 002270          CLR      XMEM              ;WE DON'T WANT TO LOAD ANY EA
1043 024532 012777 177777 155516      MOV      #-1,@RLMP        ;ONE WORD TRANSFER
1044 024540 005077 155510          CLR      @RLDA
1045 024544 004537 020674          JSR      R5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
1046 024550 000012          WRITE
1047 024552 004537 021514          JSR      R5,WTCRDY        ;WAIT FOR WRITE TO FINISH
1048 024556          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
          024556 104010      EMT      C$ESCAPE
          024560 000112      .WORD   10000$-.
1049 024562 032737 020000 002234      BIT      #NXM,E.CS        ;NON-EXISTANT MEMORY ERROR?
1050 024570 001002          BNE     3$                ;YES, CONTINUE
1051
1052 024572 004537 020432          JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
1053 024576          3$:      ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
          024576 104010      EMT      C$ESCAPE
          024600 000072      .WORD   10000$-.
1054
1055 024602 032737 000020 002234      BIT      #BA16,E.CS        ;DID BA16 SET?
1056 024610 001004          BNE     4$                ;YES, CONTINUE
1057
1058 024612          ERRDF 10.,EM31,ERRO ;BA 16 DID NOT INCREMENT
          024612 104462      TRAP  T$ERCODE
          024614 000012      .WORD   10
          024616 012433      .WORD   EM31
          024620 014452      .WORD   ERRO
1059
1060 024622          4$:      CKLOOP
          024622 104006      EMT      C$CLP1
1061
1062 024624 032737 000040 002234      BIT      #BA17,E.CS        ;DID BA17 SET ALSO?
1063 024632 001404          BEQ     5$                ;NO, GOOD CONTINUE
  
```

```
1064
1065 024634          ERRDF  11.,EM32,ERRO ;BA 17 GOT CARRIED AWAY
      024634 104462  TRAP    T$ERCODE
      024636 000013  .WORD  11
      024640 012476  .WORD  EM32
      024642 014452  .WORD  ERRO
1066
1067 024644          5$:  CKLOOP
      024644 104006  EMT    C$CLP1
1068
1069 024646 005037 002174  CLR    GD$AT          ;CHECK THAT BA15-BA0 IS CLEAR
1070 024652 013737 002236 002176  MOV    E.BA,BDDAT    ;READ BA
1071 024660 001404          BEQ    6$            ;IS BA ZERO?
1072 024662          ERRDF  12.,EM33,ERR4 ;BA SHOULD BE ZERO
      024662 104462  TRAP    T$ERCODE
      024664 000014  .WORD  12
      024666 012543  .WORD  EM33
      024670 014616  .WORD  ERR4
1073
1074 024672          6$:
1075
1076 024672          ENDSEG          ;%%END OF SEGMENT%%
      024672          10000$:
      024672 104005  EMT    C$ESEG
1077 024674          ENDTST          ;**END OF TEST**
      024674          L10041:
      024674 104001  EMT    C$ETST
1078
1079          .SBTTL  **TEST 13** - BA BIT 17 INCREMENT
1080
1081 024676          BGNTST          ;**START OF TEST**
1082
1083 024676          STARS
      ;:*****
      ;CHECK THAT BA BIT 17 WILL INCREMENT. WE WILL LOAD THE
      ;RLBA WITH 177776 AND BA 16 SET, WE WILLISSUE A ONE WORD
      ;WRITE. WE THEN CHECK BA17 TO SET, BA16 TO CLEAR AND
      ;BA15 - BAO TO CLEAR.
      STARS
      ;:*****
1084
1085
1086
1087
1088 024676
1089
1090
1091
1092 024676 004737 021574  JSR    PC,HDHOME     ;HEADS OVER TRACK 0
1093 024702          CKERFG          ;HEADS GO HOME OKAY
      024710 104032  EMT    C$EXIT
      024712 000162  .WORD  L10042-.
1094
1095 024714          BGNSEG          ;%%START OF SEGMENT%%
      024714 104004  EMT    C$BSEG
1096
1097 024716          2$:
1098 024716 012777 177776 155326  MOV    #177776,@RLBA ;SET MAX BA TO INC. BA16
1099 024724 012737 000020 002270  MOV    #BA16,XMEM    ;SET BA16 IN RLCS
1100 024732 012777 177777 155316  MOV    #-1,@RLMP    ;ONE WORD TRANSFER
1101 024740 005077 155310  CLR    @RLDA
1102 024744 004537 020674  JSR    R5,LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD
```

```

1103 024750 000012
1104 024752 004537 021514 WRITE
1105 024756 104010 JSR R5,WTCRDY ;WAIT FOR WRITE TO FINISH
    024756 000112 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
    024760 000112 EMT C$ESCAPE
1106 024762 032737 020000 002234 .WORD 10000$-
1107 024770 0010G2 BIT #NXM,E.CS ;NON-EXISTANT MEMORY ERROR?
1108 BNE 3$ ;YES, CONTINUE
1109 024772 004537 020432 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
1110 024776 3$: ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
    024776 104010 EMT C$ESCAPE
    025000 000072 .WORD 10000$-
1111
1112 025002 032737 000040 002234 BIT #BA17,E.CS ;DID BA17SET?
1113 025010 001004 BNE 4$ ;YES, CONTINUE
1114
1115 025012 ERRDF 13.,EM34,ERRO ;BA 17 DID NOT SET
    025012 104462 TRAP T$ERCODE
    025014 000015 .WORD 13
    025016 012604 .WORD EM34
    025020 014452 .WORD ERRO
1116
1117 025022 4$: CKLOOP
    025022 104006 EMT C$CLP1
1118
1119 025024 032737 000020 002234 BIT #BA16,E.CS ;DID BA16 SET ALSO?
1120 025032 001404 BEQ 5$ ;NO, GOOD CONTINUE
1121
1122 025034 ERRDF 14.,EM35,ERRO ;BA 16 DIDN'T KNOW WHEN TO QUIT.
    025034 104462 TRAP T$ERCODE
    025036 000016 .WORD 14
    025040 012647 .WORD EM35
    025042 014452 .WORD ERRO
1123 025044 5$: CKLOOP
    025044 104006 EMT C$CLP1
1124
1125 025046 005037 002174 CLR GDDAT ;CHECK THAT BA15-BA0 IS CLEAR
1126 025052 013737 002236 002176 MOV E.BA,BDDAT ;READ BA
1127 025060 001404 BEQ 6$ ;IS BA ZERO?
1128 025062 ERRDF 15.,EM36,ERR4 ;BA SHOULD BE ZERO
    025062 104462 TRAP T$ERCODE
    025064 000017 .WORD 15
    025066 012714 .WORD EM36
    025070 014616 .WORD ERR4
1129
1130 025072 6$:
1131
1132 025072 ENDSEG ;%%END OF SEGMENT%%
    025072 10000$: EMT C$ESEG
1133 025074 ENDTST ;**END OF TEST**
    025074 L10042: EMT C$ETST
    025074 104001
1134
1135 .SBITL **TEST 14** - TEST READ NPR INTEGRITY
1136
1137 025076 BGTST ;**START OF TEST**
  
```

1138
 1139
 1140
 1141
 1142 025076
 1143
 1144
 1145 025076
 1146
 1147
 1148 025076 004737 021574
 1149 025102
 025110 104032
 025112 000132
 1150
 1151 025114
 025114 104004
 1152
 1153
 1154 025116
 025116 012746 000340
 025122 012746 021566
 025126 013746 002140
 025132 012746 000003
 025136 104037
 025140 062706 000010
 1155 025144 005037 002150
 1156 025150 012777 003260 155074
 1157 025156 005077 155072
 1158 025162 012777 177777 155066
 1159 025170 004537 020674
 1160 025174 000014
 1161 025176 004537 021514
 1162 025202
 025202 013700 002140
 025206 104036
 1163 025210
 025210 104010
 025212 000030
 1164 025214 004537 020432
 1165 025220
 025220 104010
 025222 000020
 1166
 1167 025224 005737 002150
 1168 025230 001404
 1169 025232
 025232 104462
 025234 000021
 025236 013405
 025240 014452
 1170 025242
 1171
 1172
 1173 025242

```

STARS
:*****
:CHECK THAT NPR WILL NOT INTERFERE WITH THE OPERATION OF THE UNIBUS
:WE SETUP LOCATION 4 TO HANDLE THE TRAP IF IT HAPPENS
STARS
:*****

JSR PC,HDHOME ;HEADS OVER TRACK 0
CKERFG ;HEADS GO HOME OKAY
EMT C$EXIT
.WORD L10043-.

BGNSEG ;%%START OF SEGMENT%%
EMT C$BSEG

1$: SETVEC ERRVEC,#TRPHAN,#340 ;SET UP VECTOR
MOV #340,-(SP)
MOV #TRPHAN,-(SP)
MOV ERRVEC,-(SP)
MOV #3,-(SP)
EMT C$SVEC
ADD #10,SP
CLR TRPFLG ;CLEAR TRAP PLAY
MOV #BUF,@RLBA ;LOAD BA
CLR @RLDA ;LOAD DA
MOV #-1,@RLMP ;LOAD WC
JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
READ
JSR R5,WTCRDY ;
CLRVEC ERRVEC ;CLEAR OUT VECTOR
MOV ERRVEC,R0
EMT C$CVEC
ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
EMT C$ESCAPE
.WORD 10000$-.
JSP R5,CHERR ;CHECK CNTLR FOR ERRORS
ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
EMT C$ESCAPE
.WORD 10000$-.

TST TRPFLG ;DID TRAP OCCUR?
BEQ 7$ ;NO, OKAY
ERRDF 17.,EM52,ERRO ;YES, PRINT ERROR
TRAP T$ERCODE
.WORD 17
.WORD EM52
.WORD ERRO

7$:
ENDSEG ;%%END OF SEGMENT%%
  
```

```
025242 104005 10000$: EMT C$ESEG
1174 025242 104005
1175
1176 025244 ENDTST ;**END OF TEST**
025244 L10043:
025244 104001 EMT C$ETST
1177
1178 .SBTTL **TEST 15** - READ FUNCTION
1179
1180 025246 BGNTST ;**START OF TEST**
1181
1182 025246 STARS
1183 ;*****
1184 ;CHECK OF THE READ FUNCTION. WE WILL FIRST DO A READ
1185 ;HEADER TO FIND OUT WHERE WE ARE AND THEN ISSUE
1186 ;A FULL SECTOR READ, WAIT FOR READY AND CHECK FOR
1187 025246 ;ANY ERRORS
STARS
;*****
1188
1189
1190 025246 004737 021574 JSR PC,HDHOME ;HEADS OVER TRACK 0
1191 025252 CKERFG ;HEADS GO HOME OKAY
025260 104032 EMT C$EXIT
025262 000064 .WORD L10044-.
1192
1193 025264 BGNSEG ;%%START OF SEGMENT%%
025264 104004 EMT C$BSEG
1194
1195 025266 012737 001750 002166 MOV #1000.,TMPO
1196 025274 005077 154754 1$: CLR @RLDA ;LOAD DISK ADDRESS
1197 025300 012777 177600 154750 MOV #-128.,@RLMP ;SET WORD LENGTH
1198 025306 012777 003260 154736 MOV #BUF,@RLBA ;SET BUS ADDRESS
1199
1200 025314 004537 020674 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
1201 025320 000014- READ ;READ
1202 025322 004537 021514 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY
1203 025326 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
025326 104010 EMT C$ESCAPE
025330 000014 .WORD 10000$-.
1204
1205 025332 004537 020432 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
1206
1207 025336 005337 002166 DEC TMPO
1208 025342 001354 BNE 1$
1209 025344 ENDSEG ;%%END OF SEGMENT%%
025344
10000$: EMT C$ESEG
1210 025346 ENDTST ;**END OF TEST**
025346 L10044:
025346 104001 EMT C$ETST
1211
1212 .SBTTL **TEST 16** - READ FUNCTION INTERRUPT
1213
1214 025350 BGNTST ;**START OF TEST**
1215
```


1216 025350

STARS

1217
 1218
 1219
 1220

```

:*****
:CHECK OF THE READ FUNCTION UNDER INTERRUPT CONTROL, WE WILL
:ISSUE A READ HEADER TO GET POSITION AND THEN READ
:A FULL SECTOR WAITING FOR THE INTERRUPT. CHECK FOR
:ERRORS ON INTERRUPT.

```

1221 025350

STARS

1222
 1223

```

:*****

```

1224 025350 004737 021574

JSR PC,HDHOME ;HEADS OVER TRACK 0

1225 025354

CKERFG ;HEADS GO HOME OKAY

025362 104032

EMT C\$EXIT

025364 000106

.WORD L10045-

1226

1227 025366

BGNSEG ;%%START OF SEGMENT%%

025366 104004

EMT C\$BSEG

1228

1229 025370 005037 002152

CLR INTFLG ;CLEAR INTERRUPT INDICATOR

1230 025374 005077 154654

CLR @RLDA ;SET DISK ADDRESS

1231 025400 012777 177600 154650

MCV #-128,@RLMP ;SET UP WORD COUNT

1232 025406 012777 003260 154636

MOV #BUF,@RLBA ;SET UP BUS ADDRESS

1233

1234 025414

SETPRI #PRI00 ;PRIORITY TO 0

025414 012700 000000

MOV #PRI00,R0

025420 104041

EMT C\$SPRI

1235 025422 004537 020674

JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD

1236 025426 000114

READ!INTEN ;READ UNDER INTERRUPT

1237 025430 004537 021514

JSR R5,WTCRDY ;WAIT FOR INTERRUPT

1238 025434

CKLOOP

025434 104006

EMT C\$CLP1

1239 025436

SFTPRI #PRI07 ;PRIORITY TO 7

025436 012700 000340

MOV #PRI07,R0

025442 104041

EMT C\$SPRI

1240

1241 025444 005737 002152

TST INTFLG ;DID INTERRUPT OCCUR?

1242 025450 001004

BNE 1\$;YES-BRANCH NO-REPORT

1243

1244 025452

ERRDF 19,EM4,ERR0 ;READ DID NOT INTERRUPT

025452 104462

TRAP T\$ERCODE

025454 000023

.WORD 19

025456 011015

.WORD EM4

025460 014452

.WORD ERRO

1245 025462

1\$: CKLOOP ;CHECK FOR LOOP

025462 104006

EMT C\$CLP1

1246

1247 025464 004537 020432

JSR R5,CHERR ;CHECK CNTLR FOR ERRORS

1248

1249 025470

ENDSEG ;%%END OF SEGMENT%%

025470

10000\$:

025470 104005

EMT C\$ESEG

1250 025472

ENDTST ;**END OF TEST**

025472

L10045:

025472 104001

EMT C\$ETST

1251

1252

.SBTTL **TEST 17** - CHECK READ NPR DIRECTION

1253


```

1302
1303 025642 005237 002170      INC      TMP1      ;INC PASS COUNT
1304 025646 005137 002166      COM      TMP0      ;COMPLIMENT PATTERN
1305 025652 000725              BR        1$        ;GO DO IT AGAIN
1306
1307 025654          5$:      ERRDF  20.,EM5,ERR9 ;READ DID NOT MODIFY MEMORY
      025654 104462      TRAP   T$ERCODE
      025656 000024      .WORD  20
      025660 011054      .WORD  EM5
      025662 015044      .WORD  ERR9
1308
1309 025664          6$:
1310
1311 025664          ENDSEG          ;%%END OF SEGMENT%%
      025664          10000$:
      025664 104005      EMT      C$ESEG
1312 025666          ENDTST          ;**END OF TEST**
      025666          L10046:
      025666 104001      EMT      C$ETST
1313
1314          .SBTTL  **TEST 18** - PROPER INCREMENT OF RLBA ON READ
1315
1316 025670          BGNTST          ;**START OF TEST**
1317
1318 025670          STARS
      :*****
1319          :CHECK THAT THE RLBA WILL INCREMENT WITH THE READ
1320          :THE RLBA SHOULD CONTAIN "BUF +256." AFTER A FULL SECTOR
1321          :READ.
1322 025670          STARS
      :*****
1323
1324
1325 025670 004737 021574      JSR      PC,HDHOME ;HEADS OVER TRACK 0
1326 025674          CKERFG          ;HEADS GO HOME OKAY
      025702 104032      EMT      C$EXIT
      025704 000116      .WORD  L10047-.
1327
1328 025706          BGNSEG          ;%%START OF SEGMENT%%
      025706 104004      EMT      C$BSEG
1329
1330 025710 005077 154340      CLR      @RLDA      ;SET UP DISK ADDRESS
1331 025714 012777 003260 154330  MOV      #BUF,@RLBA ;SET UP BUS ADDRESS
1332 025722 012777 177600 154326  MOV      #-128.,@RLMP ;WORD COUNT
1333 025730 012737 003260 002174  MOV      #BUF,GDDAT ;FORM EXPECTED BUS ADDRESS
1334 025736 062737 000400 002174  ADD      #256.,GDDAT ;AFTER READ
1335
1336 025744 004537 020674      JSR      R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
1337 025750 000014          READ      ;READ
1338 025752 004537 021514      JSR      R5,WTCRDY ;WAIT FOR CONTROLLER READY
1339 025756          ESCAPE  SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      025756 104010      EMT      C$ESCAPE
      025760 000040      .WORD  10000$-.
1340
1341 025762 004537 020432      JSR      R5,CHERR ;CHECK CNTLR FOR ERRORS
1342 025766          ESCAPE  SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      025766 104010      EMT      C$ESCAPE
  
```

```

1343 025770 000030          .WORD 10000$-.
1343 025772 013737 002236 002176  MOV   E.BA,BDDAT      ;READ 'RLBA' FOR PRESENT ADDRESS
1344 026000 023737 002176 002174  CMP   BDDAT,GDDAT    ;DID 'BA' INCREMENT PROPERLY?
1345 026006 001404          BEQ   1$              ;YES, CONTINUE
1346
1347 026010          ERRDF 21.,EM6,ERR4    ;BA DID NOT INCREMENT PROPERLY
      026010 104462      TRAP  T$ERCODE
      026012 000025      .WORD 21
      026014 011120      .WORD EM6
      026016 014616      .WORD ERR4
1348
1349 026020          1$:
1350
1351 026020          ENDSEG          ;%%END OF SEGMENT%%
      026020          10000$:
      026020 104005      EMT   C$ESEG
1352 026022          ENDTST          ;**END OF TEST**
      026022          L10047:
      026022 104001      EMT   C$ETST
1353
1354          .SBTTL **TEST 19** - PROPER INCREMENT OF RLDA ON READ
1355
1356 026024          BGNTST          ;**START OF TEST**
1357
1358 026024          STARS
      ;*****
1359          ;CHECK THAT THE RLDA INCREMENTS BY ONE AFTER A
1360          ;FULL SECTOR READ, WE FIRST READ A HEADER TO FIND
1361          ;OUT WHERE WE ARE, THEN ISSUE A READ AFTER
1362          ;THE READ THE RLDA SHOULD BE RLDA (START) + 1
1363 026024          STARS
      ;*****
1364
1365 026024 004737 021574      JSR   PC,HDHOME      ;HEADS OVER TRACK 0
1366 026030          CKERFG          ;HEADS GO HOME OKAY
      026036 104032      EMT   C$EXIT
      026040 000114      .WORD L10050-.
1367
1368 026042          BGNSEG          ;%%START OF SEGMENT%%
      026042 104004      EMT   C$BSEG
1369
1370
1371 026044 005037 002174      CLR   GDDAT
1372 026050 013777 002174 154176  MOV   GDDAT,@RLDA    ;SETUP DISK ADDRESS
1373 026056 005237 002174      INC   GDDAT          ;CREATE EXPECTED SECTOR
1374 026062 012777 177600 154166  MOV   #-128.,@RLMP   ;WORD COUNT
1375 026070 012777 003260 154154  MOV   #BUF,@RLBA     ;SETUP BUS ADDRESS
1376
1377 026076 004537 020674      JSR   R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1378 026102 000014          READ          ;READ
1379 026104 004537 021514      JSR   R5,WTCRDY     ;WAIT FOR CONTROLLER READY
1380 026110          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      026110 104010      EMT   C$ESCAPE
      026112 000040      .WORD 10000$-.
1381
1382 026114 004537 020432      JSR   R5,CHERR      ;CHECK CNTLR FOR ERRORS
1383 026120          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
  
```

```

026120 104010          EMT      C$ESCAPE
026122 000030          .WORD   10000$-.
1384
1385 026124 013737 002240 002176      MOV      E.DA,BDDAT      ;READ DISK ADDRESS
1386 026132 023737 002174 002176      CMP      GDDAT,BDDAT    ;DID SECTOR INCREMENT PROPERLY
1387 026140 001404          BEQ      1$              ;YES, BRANCH NO, REPORT ERROR
1388
1389 026142          ERRDF   22.,EM7,ERR4  ;DISK ADDRESS DID NOT INCREMENT
      026142 104462          TRAP    T$ERCODE
      026144 000026          .WORD   22
      026146 011174          .WORD   EM7
      026150 014616          .WORD   ERR4
1390
1391 026152          1$:
1392
1393 026152          ENDSEG                      ;%%END OF SEGMENT%%
      026152          10000$:
1394 026154 104005          EMT      C$ESEG
      026154          ENDTST                      ;**END OF TEST**
      026154 104001          L10050:
      026154          EMT      C$ETST
1395
1396          .SBTTL  **TEST 20** - FORCE HEADER NOT FOUND WITH READ
1397
1398 026156          BGNTST                      ;**START OF TEST**
1399
1400 026156          STARS
      ;*****
      ;FORCE HEADER NOT FOUND ERROR TO OCCUR. THIS IS DONE
      ;BY SETTING SECTOR 40 OF THE RLDA AND ISSUING A
      ;READ. SECTOR 40 DOES NOT EXIST ON THE RL01 PACK
      ;THEREFORE HDR NT FOUND SHOULD SET.
      STARS
      ;*****
1401
1402
1403
1404
1405 026156
1406
1407 026156 004737 021574          JSR      PC,HDHOME      ;HEADS OVER TRACK 0
1408 026162          CKERFG                      ;HEADS GO HOME OKAY
      026170 104032          EMT      C$EXIT
      026172 000102          .WORD   L10051-.
1409
1410 026174          BGNSEG                      ;%%START OF SEGMENT%%
      026174 104004          EMT      C$BSEG
1411
1412
1413 026176 012777 000050 154050      MOV      #40.,@RLDA     ;INSURE NOT TO FIND HEADER BY
1414 026204 012777 003260 154040      MOV      #BUF,@RLBA    ;SETTING SECTOR 40 OF CYL. ADDR.
1415 026212 012777 177777 154036      MOV      #-1,@RLMP     ;WORD COUNT
1416
1417 026220 004537 020674          JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1418 026224 000014          READ                      ;READ
1419 026226 004537 021514          JSR      R5,WTCRDY     ;WAIT FOR CONTROLLER READY
1420 026232          ESCAPE   SEG              ;CHECK FOR FL:LOE, ELSE EXIT SEG
      026232 104010          EMT      C$ESCAPE
      026234 000036          .WORD   10000$-.
1421
1422 026236 013737 002234 002166      MOV      E.CS,IMPO      ;GET RLCS
1423 026244 042737 001777 002166      BIC      #1777,IMPO     ;SAVE ERROR BITS
  
```

TEST 20 - FORCE HEADER NOT FOUND WITH READ

SEQ 0085

```

1424 026252 022737 112000 002166      CMP      #BIT15:BIT12:BIT10,IMPO ;HDR NOT FOUND SET.
1425 026260 001404                      BEQ      1$                       ;YES, CONTINUE
1426
1427 026262                      ERRDF   23.,EM10,ERRO ;HEADER NOT FOUND WOULD NOT SET
      026262 104462                      TRAP   T$ERCODE
      026264 000027                      .WORD  23
      026266 011251                      .WORD  EM10
      026270 014452                      .WORD  ERRO
1428
1429 026272                      1$:
1430
1431
1432 026272                      ENDSEG                          ;%%END OF SEGMENT%%
      026272
      026272 104005                      10000$:
1433 026274                      EMT      C$ESEG
      026274                      ENDTST
      026274 104001                      L10051:                          ;**END OF TEST**
      026274                      EMT      C$ETST
1434
1435
1436
1437 026276                      .SBTTL  **TEST 21** - FORCE HEADER NOT FOUND WITH READ INTERRUPT
1438
1439
1440 026276                      BGNST
1441
1442
1443
1444 026276                      STARS
      ;:*****
      ;:TEST THAT HEADER NOT FOUND ERROR WILL GENERATE AN INTERRUPT
      ;:ON OCCURANCE. HEADER NOT FOUND WILL BE FORCED BY SETTING
      ;:SECTOR 40 OF RLDA AND ISSUING A READ
      ;:STARS
      ;:*****
1445
1446
1447 026276 004737 021574          JSR      PC,HDHOME ;HEADS OVER TRACK 0
1448 026302                      CKERFG ;HEADS GO HOME OKAY
      026310 104032                      EMT      C$EXIT
      026312 000142                      .WORD  L10052-.
1449
1450 026314                      BGNSEG                          ;%%START OF SEGMENT%%
      026314 104004                      EMT      C$BSEG
1451
1452 026316                      SETPRI  #PRI00
      026316 012700 000000          MOV     #PRI00,R0
      026322 104041                      EMT     C$SPRI
1453 026324 005037 002152          CLR     INTFLG ;CLEAR INTERRUPT OCCURANCE FLAG
1454 026330 012777 000050 153716          MOV     #40.,@RLDA ;INSURE NOT TO FIND HEADER BY
1455 026336 012777 003260 153706          MOV     #BUF,@RLBA ;SETTING SECTOR 40 OF CYL. ADDR.
1456 026344 012777 177777 153704          MOV     #-1,@RLMP ;WORD COUNT
1457
1458 026352 004537 020674          JSR     R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
1459 026356 000114                      READ!INTEN ;READ
1460 026360 004537 021514          JSR     R5,WTCRDY ;WAIT FOR CONTROLLER READY
1461 026364                      CKLOOP
      026364 104006                      EMT     C$CLP1
1462 026366                      SETPRI  #PRI07
      026366 012700 000340          MOV     #PRI07,R0
      026372 104041                      EMT     C$SPRI

```

```

1463
1464 026374 005737 002152          TST      INTFLG          ;DID INTERRUPT OCCUR
1465 026400 001004                  BNE      2$             ;YES
1466
1467 026402          ERRDF    24.,EM43,ERRO ;HNF DID NOT INTERRUPT
      026402 104462          TRAP    T$ERCODE
      026404 000030          .WORD   24
      026406 013131          .WORD   EM43
      026410 014452          .WORD   ERRO
1468
1469 026412          2$:      ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      026412 104010          EMT     C$ESCAPE
      026414 000036          .WORD   10000$-.
1470
1471
1472 026416 013737 002234 002166    MOV     E.CS, TMPO      ;GET RLCS
1473 026424 042737 001777 002166    BIC     #1777, TMPO     ;SAVE ERROR BITS
1474 026432 022737 112000 002166    CMP     #BIT15:BIT12:BIT10, TMPO ;WDR NOT FOUND SET.
1475 026440 001404                  BEQ     1$             ;YES, CONTINUE
1476
1477 026442          ERRDF    25.,EM10,ERRO
      026442 104462          TRAP    T$ERCODE
      026444 000031          .WORD   25
      026446 011251          .WORD   EM10
      026450 014452          .WORD   ERRO
1478
1479 026452          1$:
1480
1481 026452          ENDSEG          ;%%END OF SEGMENT%%
      026452          10000$:
      026452 104005          EMT     C$ESEG
1482 026454          ENDTST          ;**END OF TEST**
      026454          L10052:
      026454 104001          EMT     C$ETST
1483
1484          .SBTTL  **TEST 22** - CHECK HEADER COMPARE LOGIC
1485
1486 026456          BGNTST          ;**START OF TEST**
1487
1488 026456          STARS
      ;:*****
1489          ;CHECK THE HEADER COMPARE LOGIC WORKS. UP TO THIS POINT WE
1490          ;KNOW THAT THE LOGIC FUNCTIONS PROPERLY BUT NOW WE WILL
1491          ;CHECK ALL THE BITS IN THE HEADER WORD. FOUR PATTERNS
1492          ;ARE USED A WALKING 1, GROWING 1, WALKING 0, GROWING 0. A SEEN
1493          ;IS ISSUED BEFORE EACH READ TO INSURE WE ARE ON THE PROPER
1494          ;TRACK. ONCE WE ARE ON THE RIGHT TRACKWE LOAD THE RLDA
1495          ;AND ISSUE THE READ. UPON COMPLETION WE WILLCHECK FOR ERRORS
1496          ;WE THEN LOAD THE COMPLIMENT PATTERN INTO THE RLDA
1497          ;EXPECTING A HEADER NOT FOUND TO SET
1498 026456          STARS
      ;:*****
1499
1500
1501 026456 004737 021574          JSR     PC, HDHOME     ;HEADS OVER TRACK 0
1502 026462          CKERFG          ;HEADS GO HOME OKAY
      026470 104032          EMT     C$EXIT
  
```

1503	026472	000574			.WORD	L10053-	
1504	026474				BGNSEG		;%START OF SEGMENT%
	026474	104004			EMT	C\$BSEG	
1505							
1506	026476				SETPRI	#PRI07	;PRIORITY TO 7
	026476	012700	000340		MOV	#PRI07,R0	
	026502	104041			EMT	C\$SPRI	
1507	026504	022737	000001	002126	CMP	#1,T.DRIVE	;CHECK TYPE OF DRIVE (RL01 OR RL02)
1508	026512	001003			BNE	22\$;RL02? THEN BRANCH
1509	026514	012703	002522		MOV	#HDRTAB,R3	;MOV ADDRESS OF BEG PATTERN TO R3
1510	026520	000402			BR	33\$; THEN BRANCH
1511	026522	012703	002702		MOV	#HTAB,R3	;MOV ADDRESS OF BEG PATTERN TO R3
1512	026526				BGNSEG		;START OF SEGMENT
	026526	104004			EMT	C\$BSEG	
1513	026530						
1514	026530	004537	020674		JSR	R5,LDFUNC	;LOAD THE FUNCTION IN NEXT WORD
1515	026534	000010			RDHDR		;READ HEADER
1516	026536	004537	021514		JSR	R5,WTCRDY	;WAIT FOR CONTROLLRE READY
1517	026542				ESCAPE	SEG	;CHECK FOR FL:LOE, ELSE EXIT SEG
	026542	104010			EMT	C\$ESCAPE	
	026544	000516			.WORD	10001\$-	
1518							
1519	026546	004537	020432		JSR	R5,CHERR	;CHECK CNTLR FOR ERRORS
1520	026552				ESCAPE	SEG	;CHECK FOR FL:LOE, ELSE EXIT SEG
	026552	104010			EMT	C\$ESCAPE	
	026554	000506			.WORD	10001\$-	
1521	026556	013737	002242	002170	MOV	E.MP,TMP1	;READ AND SAVE HEADER
1522							
1523	026564	042737	000177	002170	BIC	#177,TMP1	;CLEAR OUT SECTOR AND H.S.
1524	026572	012777	000001	153454	MOV	#1,@RLDA	;SETUP MARKER FOR SEEK
1525	026600	011337	002172		MOV	(R3),TMP2	;GET HEADER PATTERN
1526	026604	042737	000177	002172	BIC	#177,TMP2	;CLEAR OUT SECTOR AND H.S.
1527	026612	163737	002170	002172	SUB	TMP1,TMP2	;CALCULATE DIFFERENCE TO SEEK
1528	026620	103404			BCS	2\$;BRANCH FOR SEEK OUT
1529	026622	052777	000004	153424	BIS	#SIGN,@RLDA	;SEEK TOWARDS SPINDLE
1530	026630	000402			BR	3\$;GO PUT IN DIFFERENCE WORD
1531	026632	005437	002172		NEG	TMP2	;WE HAVE TO NEGATE DIFFERENCE
1532	026636	053777	002172	153410	BIS	TMP2,@RLDA	;SET IN DIFFERENCE WORD
1533	026644	032713	000100		BIT	#RHHS,(R3)	;DO WE WANT HEAD SELECT AS 0?
1534	026650	001403			BEQ	4\$;YES, SKIP OVER SETTING H.S.
1535	026652	052777	000020	153374	BIS	#DAHS,@RLDA	;SET HEAD SELECT TO ONE
1536	026660	004537	020674		JSR	R5,LDFUNC	;LOAD THE FUNCTION IN NEXT WORD
1537	026664	000006			SEEK		;SEEK
1538							
1539							
1540	026666	004537	021514		JSR	R5,WTCRDY	;WAIT FOR CONTROLLER READY
1541	026672				ESCAPE	SEG	;CHECK FOR FL:LOE, ELSE EXIT SEG
	026672	104010			EMT	C\$ESCAPE	
	026674	000366			.WORD	10001\$-	
1542							
1543	026676	004537	020432		JSR	R5,CHERR	;CHECK CNTLR FOR ERRORS
1544	026702				ESCAPE	SEG	;CHECK FOR FL:LOE, ELSE EXIT SEG
	026702	104010			EMT	C\$ESCAPE	
	026704	000356			.WORD	10001\$-	
1545							
1546	026706	004537	021454		JSR	R5,WTCRDY	;WAIT FOR DRIVE READY

1547	026712			ESCAPE	SEG		:CHECK FOR FL:LOE, ELSE EXIT SEG
	026712	104010		EMT	C\$ESCAPE		
	026714	000346		.WORD	10001\$-		
1548	026716	004537	020674	JSR	R5,LDIFUNC		:LOAD THE FUNCTION IN NEXT WORD
1549	026722	000010		RDHDR			:READ HEADER (VERIFY SEEK)
1550	026724	004537	021514	JSR	R5,WTCRDY		:WAIT FOR CONTROLLER READY
1551	026730			ESCAPE	SEG		:CHECK FOR FL:LOE, ELSE EXIT SEG
	026730	104010		EMT	C\$ESCAPE		
	026732	000330		.WORD	10001\$-		
1552							
1553	026734	004537	020432	JSR	R5,CHERR		:CHECK CNTLR FOR ERRORS
1554	026740			ESCAPE	SEG		:CHECK FOR FL:LOE, ELSE EXIT SEG
	026740	104010		EMT	C\$ESCAPE		
	026742	000320		.WORD	10001\$-		
1555							
1556	026744	013737	002242	MOV	E.MP,BDDAT		:READ HEADER
1557	026752	043737	002156	BIC	SECMSK,BDDAT		:SAVE CYLINDER FOR COMPARE
1558	026760	011337	002174	MOV	(R3),GDDAT		:GET EXPECTED HEADER
1559	026764	043737	002156	BIC	SECMSK,GDDAT		:SAVE CYLINDER FOR COMPARE
1560	026772	023737	002174	CMP	GDDAT,BDDAT		:SEEK END UP OKAY
1561	027000	001404		BEQ	5\$:YES, CONTINUE
1562							
1563	G27002			ERRDF	27.,EM11,ERR4		:SEEK INCORRECT
	027002	104462		TRAP	T\$ERCODE		
	027004	000033		.WORD	27		
	027006	011316		.WORD	EM11		
	027010	014616		.WORD	ERR4		
1564							
1565	027012		5\$:	ESCAPE	SEG		:CHECK FOR FL:LOE, ELSE EXIT SEG
	027012	104010		EMT	C\$ESCAPE		
	027014	000246		.WORD	10001\$-		
1566							
1567	027016	011377	153232	MOV	(R3),@RLDA		:SET UP DISK ADDRESS
1568	027022	042777	000077	BIC	#77,@RLDA		
1569	027030	012777	177777	MOV	#-1,@RLMP		:WORD COUNT
1570	027036	012777	003260	MOV	#BUF,@RLBA		:BUS ADDRESS
1571							
1572	027044	004537	020674	JSR	R5,LDIFUNC		:LOAD THE FUNCTION IN NEXT WORD
1573	027050	000014		READ			:READ
1574	027052	004537	021514	JSR	R5,WTCRDY		:WAIT FOR CONTROLLER READY
1575	027056			ESCAPE	SEG		:CHECK FOR FL:LOE, ELSE EXIT SEG
	027056	104010		EMT	C\$ESCAPE		
	027060	000202		.WORD	10001\$-		
1576							
1577	027062	004537	020432	JSR	R5,CHERR		:CHECK CNTLR FOR ERRORS
1578	027066			ESCAPE	SEG		:CHECK FOR FL:LOE, ELSE EXIT SEG
	027066	104010		EMT	C\$ESCAPE		
	027070	000172		.WORD	10001\$-		
1579							
1580	027072	011377	153156	MOV	(R3),@RLDA		:SET UP DISK ADDRESS AS
1581	027076	005177	153152	COM	@RLDA		:COMPLIMENT TO CAUSE HDR NT FND
1582	027102	012777	177777	MOV	#-1,@RLMP		:WORD COUNT
1583	027110	012777	003260	MOV	#BUF,@RLBA		:BUS ADDRESS
1584							
1585	027116	004537	020674	JSR	R5,LDIFUNC		:LOAD THE FUNCTION IN NEXT WORD
1586	027122	000014		READ			:READ
1587	027124	004537	021514	JSR	R5,WTCRDY		:WAIT FOR CONTROLLER READY

```

1588 027130          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      027130 104010  EMT C$ESCAPE
      027132 000130  .WORD 10001$-.
1589 027134 013737 002234 002166  MOV E,CS, TMPO ;GET CS
1590 027142 042737 001777 002166  BIC #1777, TMPO ;SAVE ERROR BITS
1591 027150 022737 112000 002166  CMP #BIT15!BIT12!BIT10, TMPO ;DID HEADER NOT FOUND SET
1592 027156 001402  BEQ 8$ ;YES, CONTINUE
1593 027160 004537 020432  JSR R5, CHERR
1594 027164          8$: CKLOOP
      027164 104006  EMT C$CLP1
1595
1596 027166 022737 112000 002166  CMP #BIT15!BIT12!BIT10, TMPO
1597 027174 001413  BEQ 6$
1598
1599 027176 011337 002174  MOV (R3), GDDAT ;SET UP DATA FOR ERROR
1600 027202 013737 002174 002176  MOV GDDAT, BDDAT ;PRINT OUT
1601 027210 005137 002176  COM BDDAT
1602
1603 027214          ERRDF 28., EM12, ERR4 ;HDR NOT FOUND WOULD NOT SET
      027214 104462  TRAP T$ERCODE
      027216 000034  .WORD 28
      027220 011345  .WORD EM12
      027222 014616  .WORD ERR4
1604
1605 027224          6$: ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      027224 104010  EMT C$ESCAPE
      027226 000034  .WORD 10001$-.
1606
1607 027230 005723  TST (R3)+ ;GET NEXT PATTERN
1608 027232 022737 000001 002126  CMP #1, T.DRIVE ;TYPE OF DRIVE RL01 OR RL02
1609 027240 001003  BNE 60$ ;RL02 ? THEN BRANCH
1610 027242 020327 002700  CMP R3, #HDREND ;CMP IT WITH #HDREND
1611 027246 000402  BR 77$ ;THEN BRANCH
1612 027250 020327 003066  60$: CMP R3, #HEND ;CMP IT WITH #HEND
1613 027254 001402  77$: BEQ 7$ ;YES, EXIT TEST
1614 027256 000137 026530  JMP 1$ ;NO, GO BACK
1615
1616 027262          7$:
1617 027262          ENDSEG ;%%END OF SEGMENT%%
      027262 10001$:
      027262 104005  EMT C$ESEG
1618
1619 027264          ENDSEG ;%%END OF SEGMENT%%
      027264 10000$:
      027264 104005  EMT C$ESEG
1620 027266          ENDTST ;**END OF TEST**
      027266 L10053:
      027266 104001  EMT C$ETST
1621
1622          .SBTTL **TEST 23** - CHECK MULTIPLE SECTORS ON READ
1623
1624 027270          BGNTST ;**START OF TEST**
1625
1626 027270          STARS
      ;:*****
1627          ;:VERIFY THAT MULTIPLE SECTORS CAN BE READ, WE WILL CHECK
1628          ;:THAT THE RLDA INCREMENTS PROPERLY.
  
```

```

1629 027270          STARS
1630                ;;*****
1631
1632 027270 004737 021574      JSR    PC,HDHOME      ;HEADS OVER TRACK 0
1633 027274                CKERFG      ;HEADS GO HOME OKAY
      027302 104032      EMT    C$EXIT
      027304 000156      .WORD  L10054-.
1634
1635
1636 027306 005037 002166      CLR    TMPO          ;CLEAR LOCATIONS
1637 027312 005037 002170      CLR    TMP1
1638
1639 027316                BGNSEG      ;%%START OF SEGMENT%%
      027316 104004      EMT    C$BSEG
1640
1641 027320                1$:
1642 027320 013737 002170 002174  MOV    TMP1,GDDAT    ;GET CYLINDER
1643 027326 053737 002166 002174  BIS    TMPO,GDDAT    ;GET SECTOR
1644 027334 013777 002174 152712  MOV    GDDAT,@RLDA   ;SET DISK ADDRESS-SECTOR 0
1645 027342 062737 000002 002174  ADD    #2,GDDAT      ;SET EXPECTED + 2
1646 027350 012777 003260 152674  MOV    #BUF,@RLBA    ;SET BUS ADDRESS
1647 027356 012777 177577 152672  MOV    #-129.,@RLMP  ;WORD COUNT-SECTOR+1 WORD
1648
1649 027364 004537 020674      JSR    R5,LDFUNC     ;LOAD THE FUNCTION IN NEXT WORD
1650 027370 000014                READ      ;READ
1651 027372 004537 021514      JSR    R5,WTCRDY    ;WAIT FOR CONTROLLER READY?
1652 027376                ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      027376 104010      EMT    C$ESCAPE
      027400 000060      .WORD  10000$-.
1653
1654 027402 004537 020432      JSR    R5,CHERR     ;CHECK CNTLR FOR ERRORS
1655 027406                ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      027406 104010      EMT    C$ESCAPE
      027410 000050      .WORD  10000$-.
1656
1657 027412 013737 002240 002176  MOV    E.DA,BDDAT    ;READ DISK ADDRESS
1658 027420 023737 002176 002174  CMP    BDDAT,GDDAT   ;IS DISK ADDRESS CORRECT
1659 027426 001404      BEQ    2$          ;YES, BRANCH NO, REPORT ERROR
1660
1661 027430                ERRDF 29.,EM14,ERR4 ;DA DID NOT INCREMENT
      027430 104462      TRAP  T$ERCODE
      027432 000035      .WORD  29
      027434 011436      .WORD  EM14
      027436 014616      .WORD  ERR4
1662
1663 027440                2$:
      027440 104010      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      027442 000016      EMT    C$ESCAPE
      .WORD  10000$-.
1664
1665 027444 005237 002166      INC    TMPO          ;NEXT SECTOR?
1666 027450 022737 000046 002166  CMP    #46,TMPO      ;DONE?
1667 027456 001320      BNE    1$          ;NO, GO BACK
1668
1669
1670 027460                ENDSEG      ;%%END OF SEGMENT%%
      027460                10000$:
  
```

```

1671 027460 104005          EMT      C$ESEG
                                ENDTST          ;**END OF TEST**
                                L10054:
1672 027462 104001          EMT      C$ETST
                                STARS
                                ;*****
                                ;CHECK THAT WE CAN FORCE A HEADER NOT FOUND AT THE
                                ;END OF A TRACK DOING A MULTIPLE SECTOR READ. WE
                                ;SET UP TO READ TWO SECTORS STARTING AT SECTOR 39
                                ;WE SHOULD TRANSFER 128 WORDS THEN ABORT WITH A
                                ;HEADER NOT FOUND FOR SECTOR 40
                                ;*****
1673 027464
1674
1675
1676
1677
1678 027464
                                STARS
                                ;*****

1679
1680
1681          .SBTTL  **TEST 24** - FORCE HDR NT FND AT END OF TRACK
1682
1683 027464          BGNTST          ;**START OF TEST**
1684
1685
1686 027464 004737 021574      JSR      PC,HDHOME          ;HEADS OVER TRACK 0
1687 027470          CKERFG          ;HEADS GO HOME OKAY
                                EMT      C$EXIT
                                .WORD    L10055-.
1688
1689 027502          BGNSEG          ;%%START OF SEGMENT%%
                                EMT      C$BSEG
1690
1691 027504 012737 000047 002174  MOV      #39.,GDDAT          ;CREATE LAST SECTOR
1692 027512 013777 002174 152534  MOV      GDDAT,@RLDA          ;LOAD DISK ADDRESS
1693 027520 012777 177577 152530  MOV      #-129.,@RLMP          ;WORD COUNT
1694 027526 012777 003260 152516  MOV      #BUF,@RLBA          ;BUS ADDRESS
1695 027534 004537 020674          JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
1696 027540 000014          READ          ;READ
1697 027542 004537 021514          JSR      R5,WTCRDY          ;WAIT FOR CONTROLLER READY
1698 027546          ESCAPE          ;CHECK FOR FL:LOE, ELSE EXIT SEG
                                EMT      C$ESCAPE
                                .WORD    10000$-.
1699
1700 027552 013737 002234 002176  MOV      E.CS,BDDAT          ;READ CS
1701 027560 042737 001777 002176  BIC      #1777,BDDAT          ;SAVE ERROR BITS
1702 027566 022737 112000 002176  CMP      #112000,BDDAT          ;HDR NOT FOUND SET?
1703 027574 001402          BEQ      4$
1704 027576 004537 020432          JSR      R5,CHERR
1705 027602          4$:          CKLOOP
                                EMT      C$CLP1
                                027602 104006
1706
1707 027604 022737 112000 002176  CMP      #112000,BDDAT
1708 027612 001404          BEQ      1$
1709
1710 027614          ERRDF          ;HEADER NOT FOUND DID NOT SET
                                TRAP      T$ERCODE
                                .WORD    30
                                .WORD    EM23
                                .WORD    ERRO
1711
1712 027624          1$:
  
```

```

1713
1714 027624          ENDSEG          ;%%END OF SEGMENT%%
      027624          10000$:
      027624 104005  EMT      C$ESEG
1715 027626          ENDTST          ;**END OF TEST**
      027626          L10055:
      027626 104001  EMT      C$ETST
1716
1717          .SBTTL  **TEST 25** - FORCE NON-EXISTANT MEMORY ERROR
1718
1719 027630          BGNTST          ;**START OF TEST**
1720
1721
1722 027630          STARS
      ;:*****
      ;FORCE A NON-EXISTANT MEMORY ERROR,
      ;WE SET THE RLBA TO EQUAL THE
      ;LAST ADDRESS IN MEMORY AND ISSUE A READ.  THE
      ;READ SHOULD ABORT AFTER ONE WORD TRANSFERRED
      STARS
      ;:*****
1723
1724
1725
1726
1727 027630
1728
1729
1730 027630 004737 021574      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
1731 027634      CKERFG          ;HEADS GO HOME OKAY
      027642 104032      EMT      C$EXIT
      027644 000076      .WORD   L10056-.
1732
1733 027646          BGNSEG          ;%%START OF SEGMENT%%
      027646 104004  EMT      C$BSEG
1734
1735
1736
1737
1738 027650 012777 177774 152374      MOV      #177774,@RLBA  ;LEAD BA
1739 027656 012737 000060 002270      MOV      #BA16!BA17,XMEM ;SET EA BIT
1740 027664 005077 152364          CLR      @RLDA          ;LOAD DISK AVAILABLE
1741 027670 012777 177600 152360      MOV      #-128.,@RLMP  ;WORD COUNT
1742 027676 004537 020674          JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1743 027702 000014          READ          ;READ
1744 027704 004537 021514          JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER
1745 027710          ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      027710 104010      EMT      C$ESCAPE
      027712 000026      .WORD   10000$-.
1746
1747 027714 032737 020000 002234      BIT      #NXM,E.CS      ;DID NXM SET?
1748 027722 001004          BNE      3$              ;YES, CONTINUE
1749
1750 027724          ERRDF 31.,EM24,ERRO ;NXM DID NOT SET
      027724 104462      TRAP  T$ERCODE
      027726 000037      .WORD   31
      027730 012124      .WORD   EM24
      027732 014452      .WORD   ERRO
1751
1752 027734          3$:  ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      027734 104010      EMT      C$ESCAPE
      027736 000002      .WORD   10000$-.
  
```

```

1753
1754
1755
1756
1757 027740          ENDSEG          ;%%END OF SEGMENT%%
      027740          10000$:
      027740 104005  EMT      C$ESEG
1758 027742          ENDTST          ;**END OF TEST**
      027742          L10056:
      027742 104001  EMT      C$ETST
1759
1760          .SBTTL  **TEST 26** - FORCE NON-EXISTANT MEMORY ERROR INTERRUPT
1761
1762 027744          BGNTST          ;**START OF TEST**
1763 027744          STARS
      ;*****
1764          ;CHECK THAT WE CAN FORCE AN INTERRUPT WITH A
1765          ;NON-EXISTANT MEMORY ERROR.
1766 027744          STARS
      ;*****
1767
1768
1769 027744 004737 021574      JSR      PC,HDHOME      ;HEADS OVER TRACK 0
1770 027750          CKERFG          ;HEADS GO HOME OKAY
      027756 104032      EMT      C$EXIT
      027760 000140      .WORD    L10057-.
1771
1772 027762          BGNSEG          ;%%START OF SEGMENT%%
      027762 104004      EMT      C$BSEG
1773
1774 027764 005037 002152      CLR      INTFLG          ;CLEAR INTERRUPT OCCURANCE FLAG
1775
1776
1777
1778 027770          SETPRI  #PRI00
      027770 012700 000000      MOV      #PRI00,R0
      027774 104041      EMT      C$SPRI
1779 027776 012777 177774 152246      MOV      #177774,@RLBA      ;PRELOAD BA
1780 030004 012737 000060 002270      MOV      #BA16!BA17,XMEM  ;SET EA BITS
1781 030012 005077 152236          CLR      @RLDA          ;LOAD DA
1782 030016 012777 177777 152232      MOV      #-1,@RLMP      ;WORD COUNT
1783 030024 004537 020674          JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
1784 030030 000114          READ!INTEN          ;READ
1785 030032 004537 021514          JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER
1786 030036          SETPRI  #PRI07          ;PRIORITY TO 7
      030036 012700 000340      MOV      #PRI07,R0
      030042 104041      EMT      C$SPRI
1787 030044          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      030044 104010      EMT      C$ESCAPE
      030046 000050      .WORD    10000$-.
1788
1789 030050 005737 002152          TST      INTFLG          ;INTERRUPT OCCUR?
1790 030054 001004          BNE      4$          ;YES OKAY
1791
1792 030056          ERRDF  32.,EM44,ERRO ;NO INTERRUPT W/NXM
      030056 104462      TRAP   T$ERCODE
      030060 000040      .WORD    32
  
```

```

030062 013175      .WORD  EM44
030064 014452      .WORD  ERRO
1793
1794 030066          4$:  ESCAPE  SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
    030066 104010    EMT      C$ESCAPE
    030070 000026    .WORD  10000$-.
1795
1796 030072 032737 020000 002234  BIT      #NXM,E.CS    ;DID NXM SET?
1797 030100 001004    BNE      3$        ;YES, CONTINUE
1798
1799 030102          ERRDF  33.,EM24,ERRO  ;NO NXM
    030102 104462    TRAP   T$ERCODE
    030104 000041    .WORD  33
    030106 012124    .WORD  EM24
    030110 014452    .WORD  ERRO
1800
1801 030112          3$:  ESCAPE  SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
    030112 104010    EMT      C$ESCAPE
    030114 000002    .WORD  10000$-.
1802
1803
1804 030116          ENDSEG                          ;%%END OF SEGMENT%%
    030116          10000$:
    030116 104005    EMT      C$ESEG
1805 030120          ENDTST                          ;**END OF TEST**
    030120          L10057:
    030120 104001    EMT      C$ETST
1806
1807          .SBTTL  **TEST 27** - CHECK READ WRITE LOOP
1808
1809 030122          BGNTST                          ;**START OF TEST**
1810
1811 030122          STARS
    ;:*****
    ;:VERIFY THAT THE WRITE ACTUALLY WRITES.  AT THIS
    ;:TIME WE KNOW THAT THE WRITE FUNCTION GOES THRU
    ;:THE MOTIONS BUT WE DON'T KNOW THAT THE DATA
    ;:ACTUALLY GETS RECORDED ON THE PLATTER.
    ;:STARS
    ;:*****
1812
1813
1814
1815
1816 030122
1817
1818
1819 030122 004737 021574  JSR      PC,HDHOME    ;HEADS OVER TRACK 0
1820 030126          CKERFG  ;HEADS GO HOME OKAY
    030134 104032    EMT      C$EXIT
    030136 000362    .WORD  L10060-.
1821
1822 030140          BGNSEG                          ;%%START OF SEGMENT%%
    030140 104004    EMT      C$BSEG
1823
1824 030142 012700 003260  MOV      #BUF,R0      ;SET UP WRITE BUFFER
1825 030146 012701 000200  MOV      #128.,R1     ;128 WORDS/ONE SECTOR
1826 030152 012720 125252  3$:  MOV      #125252,(R0)+ ;WRITE PATTERN TO BUFFER
1827 030156 005301          DEC      R1          ;DONE?
1828 030160 001374          BNE      3$          ;NO, BRANCH BACK
1829 030162 005077 152066  CLR      @RLDA        ;DISK ADDRESS
1830 030166 012777 177600 152062  MOV      #-128.,@RLMP ;WORD COUNT
  
```

```

1831 030174 012777 003260 152050    MOV    #BUF,@RLBA    ;BUS ADDRESS
1832 030202 004537 020674                JSR    R5,LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD
1833 030206 000012                WRITE                ;WRITE THE PATTERN
1834 030210 004537 021514                JSR    R5,WTCRDY    ;WAIT FOR CONTROLLER READY
1835 030214                ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      030214 104010    EMT    C$ESCAPE
      030216 0003C0    .WORD 10000$-.

1836
1837 030220 004537 020432                JSR    R5,CHERR          ;CHECK CNTLR FOR ERRORS
1838 030224                ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      030224 104010    EMT    C$ESCAPE
      030226 000270    .WORD 10000$-.

1839 030230                BGNSEG                ;%%START OF SEGMENT%%
      030230 104004    EMT    C$BSEG

1840 030232 012700 003260                MOV    #BUF,R0        ;CLEAR OUT BUFFER BEFORE
1841 030236 012701 000200                MOV    #128.,R1       ;READING
1842 030242 005020                CLR    (R0)+          ;CLEAR BUFFER
1843 030244 005301                DEC    R1              ;DONE
1844 030246 001375                BNE    4$             ;NO, BRANCH BACK
1845
1846 030250 005077 152000                CLR    @RLDA          ;LOAD DISK ADDRESS
1847 030254 012777 177600 151774    MOV    #-128.,@RLMP   ;WORD COUNT/ONE SECTION
1848 030262 012777 003260 151762    MOV    #BUF,@RLBA    ;LOAD BUS ADDRESS
1849 030270 004537 020674                JSR    R5,LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD
1850 030274 000014                READ                ;GO READ
1851 030276 004537 021514                JSR    R5,WTCRDY    ;WAIT FOR CONTROLLER READY
1852 030302                ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      030302 104010    EMT    C$ESCAPE
      030304 000210    .WORD 10001$-.

1853
1854 030306 004537 020432                JSR    R5,CHERR          ;CHECK CNTLR FOR ERRORS
1855 030312 005737 002132                TST    T.CRC          ;WAS ERROR A DCK??
1856 030316 001003                BNE    8$             ;YES,SEE IF WE A DUMP
1857 030320                ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      030320 104010    EMT    C$ESCAPE
      030322 000172    .WORD 10001$-.

1858 030324 000404                BR     99$           ;SKIP AROUND
1859 030326 005737 017204                TST    T.DMP          ;DO WE STILL WANT TO CHECK IT
1860 030332 001772                BEQ    10$           ;NO
1861 030334                CKLOOP              ;YES, CHECK FOR LOOP FIRST
      030334 104006    EMT    C$CLP1

1862
1863 030336 005037 002136                CLR    CDCNT          ;CLEAR NUMBER WE'RE TO PRINT
1864 030342 005037 002130                CLR    CHECK          ;ALLOW HEADER ON FIRST PRINT
1865 030346 012702 003260                MOV    #BUF,R2        ;COMPARE BUFFER TO CHECK WRITE
1866 030352 012701 000200                MOV    #128.,R1       ;128 WORDS
1867 030356 012737 125252 002174    MOV    #125252,GDDAT ;SET UP EXPECTED
1868 030364 011237 002176                MOV    (R2),BDDAT     ;GET DATA
1869 030370 023737 002174 002176    CMP    GDDAT,BDDAT   ;IS DATA OKAY
1870 030376 001442                BEQ    6$             ;YES, CONTINUE
1871 030400 010237 002170                MOV    R2,TMP1        ;LOAD BAD MEM LOCATION
1872 030404 023737 002136 017206    CMP    CDCNT,T.LMT   ;CHECKED ENOUGH??
1873 030412 001002                BNE    333$          ;NO
1874 030414                ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      030414 104010    EMT    C$ESCAPE
      030416 000076    .WORD 10001$-.

1875 030420 005237 002136                333$: INC    CDCNT    ;ACCOUNT FOR IT
  
```



```
1876
1877 030424 005737 002130      TST      CHECK      :HEADER OR JUST DATA
1878 030430 001007              BNE      9$          :JUST DATA
1879 030432              ERRDF   34.,EM25,ERR8 :BAD DATA
      030432 104462      TRAP    T$ERCODE
      030434 000042      .WORD   34
      030436 0122C2      .WORD   EM25
      030440 014772      .WORD   ERR8
1880 030442 005237 002130      INC      CHECK      :ACCOUNT FOR PRINT OF HEADER
1881 030446 000416      BR       6$
1882
1883 030450          9$:      PRINTB #FRMT6,TMP1,GDDAT,BDDAT
      030450 013746 002176      MOV     BDDAT,-(SP)
      030454 013746 002174      MOV     GDDAT,-(SP)
      030460 013746 002170      MOV     TMP1,-(SP)
      030464 012746 016250      MOV     #FRMT6,-(SP)
      030470 012746 000004      MOV     #4,-(SP)
      030474 010600      MOV     SP,R0
      030476 104014      EMT     C$PNTB
      030500 062706 000012      ADD     #12,SP
1884
1885 030504          6$:      CKLOOP
      030504 104006      EMT     C$CLP1
1886 030506 005722          7$:      TST     (R2)+      :BUMP BUFFER POINTER
1887 030510 005301          DEC     R1          :DONE?
1888 030512 001324          BNE     5$          :NO, GO BACK
1889 030514          ENDSEG      :%%END OF SEGMENT%%
      030514
      030514 104005          10001$:      EMT     C$ESEG
1890 030516          ENDSEG      :%%END OF SEGMENT%%
      030516
      030516 104005          10000$:      EMT     C$ESEG
1891 030520          ENDTST      :**END OF TEST**
      030520          L10060:
      030520 104001          EMT     C$ETST
1892
1893          .SBTTL  **TEST 28** - CHECK SILO LINES
1894
1895 030522          BGNTST      :**START OF TEST**
1896
1897
1898
1899 030522          STARS
      :*****
      :TEST THAT LINES IN / TO SILO ARE GOOD, THAT IS THAT EACH LINE IS
      :GOOD AND CAN BE AT EITHER A 1 OR A 0 STATE INDEPENTENTLY OF EACH
      :OTHER BIT POSITION THIS IS DONE BY WRITING PATTERNS OF FLOATING 1,
      :FLOATING 0, WALKING 0, WALKING 1
      :*****
1900
1901
1902
1903
1904 030522
1905
1906
1907 030522 004737 021574      JSR     PC,HDHOME   :HEADS OVER TRACK 0
1908 030526          CKERFG      :HEADS GO HOME OKAY
      030534 104032      EMT     C$EXIT
      030536 000404      .WORD   L10061-.
1909
```

1910	030540	012703	003070		MOV	#DATPAT,R3	
1911							
1912							
1913	030544				BGNSEG		;%START OF SEGMENT%
	030544	104004			EMT	C\$BSEG	
1914	030546	012700	003260	6\$:	MOV	#BUF,R0	;WRITE PATTERN INTO MEMORY
1915	030552	012701	000200		MOV	#128.,R1	;128 WORDS
1916	030556	011320		2\$:	MOV	(R3),(R0)+	;WRITE THE PATTERN
1917	030560	005301			DEC	R1	;DONE?
1918	030562	001375			BNE	2\$;NO GO BACK
1919							
1920	030564	012777	003260	151460	MOV	#BUF,@RLBA	;SETUP TO WRITE PATTERN ONTO DISK
1921	030572	005077	151456		CLR	@RLDA	;LOAD DA
1922	030576	012777	177600	151452	MOV	#-128.,@RLMP	;WORD COUNT
1923	030604	004537	020674		JSR	R5,LDFUNC	;LOAD THE FUNCTION IN NEXT WORD
1924	030610	000012			WRITE		
1925	030612	004537	021514		JSR	R5,WTCRDY	
1926	030616				ESCAPE	SEG	;CHECK FOR FL:LOE, ELSE EXIT SEG
	030616	104010			EMT	C\$ESCAPE	
	030620	000320			.WORD	10000\$-	
1927	030622	004537	020432		JSR	R5,CHERR	;CHECK CNTLR FOR ERRORS
1928	030626				ESCAPE	SEG	;CHECK FOR FL:LOE, ELSE EXIT SEG
	030626	104010			EMT	C\$ESCAPE	
	030630	000310			.WORD	10000\$-	
1929	030632				BGNSEG		;%START OF SEGMENT%
	030632	104004			EMT	C\$BSEG	
1930	030634	012700	003260		MOV	#BUF,R0	;CLEAR MEMORY BEFORE READING IT BACK
1931	030640	012701	000200		MOV	#128.,R1	;128 WORDS
1932	030644	005020		3\$:	CLR	(R0)+	;CLEAR
1933	030646	005301			DEC	R1	;EONE
1934	030650	001375			BNE	3\$;NO
1935							
1936	030652	012777	003260	151372	MOV	#BUF,@RLBA	;SETUP TO READ IT BACK
1937	030660	012777	177600	151370	MOV	#-128.,@RLMP	;128 WORDS
1938	030666	005077	151362		CLR	@RLDA	;SECTOR ZERO
1939	030672	004537	020674		JSR	R5,LDFUNC	;LOAD THE FUNCTION IN NEXT WORD
1940	030676	000014			READ		
1941	030700	004537	021514		JSR	R5,WTCRDY	
1942	030704				ESCAPE	SEG	;CHECK FOR FL:LOE, ELSE EXIT SEG
	030704	104010			EMT	C\$ESCAPE	
	030706	000224			.WORD	10001\$-	
1943	030710	004537	020432		JSR	R5,CHERR	;CHECK CNTLR FOR ERRORS
1944	030714	005737	002132		TST	T.CRC	;WAS ERROR A DCK??
1945	030720	001003			BNE	8\$;YES,SEE IF WE A DUMP
1946	030722			10\$:	ESCAPE	SEG	;CHECK FOR FL:LOE, ELSE EXIT SEG
	030722	104010			EMT	C\$ESCAPE	
	030724	000206			.WORD	10001\$-	
1947	030726	000404			BR	99\$;SKIP AROUND
1948	030730	005737	017204	8\$:	TST	T.DMP	;DO WE STILL WANT TO CHECK IT
1949	030734	001772			BEQ	10\$;NO
1950	030736				CKLOOP		;YES, CHECK FOR LOOP FIRST
	030736	104006			EMT	C\$CLP1	
1951							
1952	030740	005037	002136	99\$:	CLR	CDCNT	;CLEAR NUMBER WE'RE TO PRINT
1953	030744	005037	002130		CLR	CHECK	;ALLOW HEADER ON FIRST PRINT
1954	030750	011337	002174		MOV	(R3),GDDAT	;COMPARE WHAT WE READ BACK
1955	030754	012737	003260	002172	MOV	#BUF,IMP2	;BUFFER START

```

1956 030762 012737 000001 002170      MOV      #1,TMP1          ;START WITH FIRST
1957
1958 030770 017737 151176 002176 5$:   MOV      @TMP2,BDDAT      ;GET DATA
1959 030776 023737 002174 002176      CMP      GDDAT,BDDAT     ;GOOD?
1960 031004 001440                BEQ      4$              ;YES, BRANCH
1961
1962 031006 023737 002136 017206      CMP      CDCNT,T.LMT     ;CHECKED ENOUGH??
1963 031014 001002                BNE      333$           ;NO
1964 031016                ESCAPE  SEG              ;CHECK FOR FL:LOF, ELSE EXIT SEG
      EMT      C$ESCAPE
      .WORD  10001$-
1965 031022 005237 002136      333$:   INC      CDCNT           ;ACCOUNT FOR IT
1966
1967 031026 005737 002130      TST      CHECK           ;HEADER OR JUST DATA
1968 031032 001007                BNE      9$            ;JUST DATA
1969 031034                ERRDF   35,,EM45,ERR10  ;BAD DATA BACK
      TRAP  T$ERCODE
      .WORD  35
      .WORD  EM45
      .WORD  ERR10
1970
1971 031044 005237 002130      INC      CHECK           ;ACCOUNT FOR PRINT OF HEADER
1972 031050 000416                BR       4$
1973
1974 031052                9$:   PRINTB #FRMT7,TMP1,GDDAT,BDDAT
      MOV      BDDAT,-(SP)
      MOV      GDDAT,-(SP)
      MOV      TMP1,-(SP)
      MOV      #FRMT7,-(SP)
      MOV      #4,-(SP)
      MOV      SP,R0
      EMT      C$PNTB
      ADD      #12,SP
1975 031106                4$:   CKLOOP
      EMT      C$CLP1
      MOV      031052 013746 002176
      MOV      031056 013746 002174
      MOV      031062 013746 002170
      MOV      031066 012746 016325
      MOV      031072 012746 000004
      MOV      031076 010600
      MOV      031100 104014
      EMT      031102 062706 000012
1976
1977 031110 062737 000002 002172      ADD      #2,TMP2         ;NEXT LOCATION
1978 031116 005237 002170                INC      TMP1           ;NEXT WORD
1979 031122 023727 002170 000201      CMP      TMP1,#129.     ;DONE
1980 031130 001317                BNE      5$            ;NO, GO BACK
1981
1982 031132                ENDSEG                  ;%%END OF SEGMENT%%
      10001$:
      EMT      C$ESEG
1983
1984 031134 005723                TST      (R3)+         ;DONE ALL PATTERNS
1985 031136 001203                BNE      6$            ;NO, GO BACK
1986
1987 031140                ENDSEG                  ;%%END OF SEGMENT%%
      10000$:
      EMT      C$ESEG
1988 031142                ENDTST
      031142 104001      L10061:                ;**END OF TEST**
      EMT      C$ETST
1989
1990                .SBTTL **TEST 29** - CHECK THROUGHPUT OF SILO
1991

```

```

1992 031144          BGNTST          ;**START OF TEST**
1993
1994
1995
1996 031144          STARS
:*****
:TEST THAT THE SILO OPERATES CORRECTLY, WE WILL WRITE A PATTERN THAT CONTAINS
:A UNIQUE PATTERN IN EACH LOCATION. WE EXPECT IT BACK IN PROPER
:ORDER, WE DO A ONE SECTOR TRANSFER
STARS
:*****

2001
2002
2003 031144 004737 021574          JSR      PC,HDHOME          ;HEADS OVER TRACK 0
2004 031150          CKERFG          ;HEADS GO HOME OKAY
      031156 104032          EMT      C$EXIT
      031160 000410          .WORD   L10062-.

2005
2006 031162          BGNSEG          ;%%START OF SEGMENT%%
      031162 104004          EMT      C$BSEG

2007
2008
2009 031164 012700 000001          MOV      #1,R0             ;INITIAL 1
2010 031170 012701 000200          MOV      #128.,R1         ;128 WORDS
2011 031174 012702 003260          MOV      #BUF,R2          ;BUFFER
2012 031200 010022          2$: MOV      R0,(R2)+        ;WRITE A WORD
2013 031202 005200          INC      R0               ;NEXT PATTERN (1-128)
2014 031204 005301          DEC      R1               ;DONE
2015 031206 001374          BNE      2$              ;NO

2016
2017 031210 012777 003260 151034    MOV      #BUF,@RLBA        ;SETUP TO WRITE
2018 031216 012777 177600 151032    MOV      #-128.,@RLMP      ;128 WORDS
2019 031224 005077 151024          CLR      @RLDA             ;DISK ADDRESS 0
2020 031230 004537 020674          JSR      R5,LDFUNC         ;LOAD THE FUNCTION IN NEXT WORD
2021 031234 000012          WRITE
2022 031236 004537 021514          JSR      R5,WTCRDY
2023 031242          F$ESCAPE SEG             ;CHECK FOR FL:LOE, ELSE EXIT SEG
      031242 104010          EMT      C$ESCAPE
      031244 000322          .WORD   10000$-.

2024
2025 031246 004537 020432          JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
2026 031252          ESCAPE SEG             ;CHECK FOR FL:LOE, ELSE EXIT SEG
      031252 104010          EMT      C$ESCAPE
      031254 000312          .WORD   10000$-.

2027 031256          BGNSEG          ;%%START OF SEGMENT%%
      031256 104004          EMT      C$BSEG

2028 031260 012700 003260          MOV      #BUF,R0             ;CLEAR BUFFER
2029 031264 012701 000200          MOV      #128.,R1         ;128 IN LENGTH
2030 031270 005020          3$: CLR      (R0)+        ;CLEAR
2031 031272 005301          DEC      R1               ;DOWN COUNT
2032 031274 001375          BNE      3$              ;DONE?

2033
2034 031276 012777 003260 150746    MOV      #BUF,@RLBA        ;BUS ADDRESS
2035 031304 012777 177600 150744    MOV      #-128.,@RLMP      ;WORD COUNT
2036 031312 005077 150736          CLR      @RLDA             ;DISK ADDRESS
2037 031316 004537 020674          JSR      R5,LDFUNC         ;LOAD THE FUNCTION IN NEXT WORD
2038 031322 000014          READ
  
```

2039	031324	004537	021514		JSR	R5,WTCRDY	
2040	031330				ESCAPE	SEG	:CHECK FOR FL:LOE, ELSE EXIT SEG
	031330	104010			EMT	C\$ESCAPE	
	031332	000232			.WORD	10001\$-	
2041							
2042	031334	004537	020432		JSR	R5,CHERR	:CHECK CNTLR FOR ERRORS
2043	031340	005737	002132		TST	T.CRC	:WAS ERROR A DCK??
2044	031344	001003			BNE	8\$:YES,SEE IF WE A DUMP
2045	031346			10\$:	ESCAPE	SEG	:CHECK FOR FL:LOE, ELSE EXIT SEG
	031346	104010			EMT	C\$ESCAPE	
	031350	000214			.WORD	10001\$-	
2046	031352	000404			BR	99\$:SKIP AROUND
2047	031354	005737	017204	8\$:	TST	T.DMP	:DO WE STILL WANT TO CHECK IT
2048	031360	001772			BEQ	10\$:NO
2049	031362				CKLOOP		:YES, CHECK FOR LOOP FIRST
	031362	104006			EMT	C\$CLP1	
2050							
2051	031364	005037	002136	99\$:	CLR	CDcnt	:CLEAR NUMBER WE'RE TO PRINT
2052	031370	005037	002130		CLR	CHECK	:ALLOW HEADER ON FIRST PRINT
2053	031374	012737	000001 002174		MOV	#1,GDDAT	:START GOOD AT 1
2054	031402	012737	003260 002172		MOV	#BUF,TMP2	:START OF BUFFER
2055	031410	012737	000001 002170		MOV	#1,TMP1	:FIRST WORD
2056							
2057	031416	017737	150550 002176	4\$:	MOV	@TMP2,BDDAT	:GET WORD
2058	031424	023737	002176 002174		CMP	BDDAT,GDDAT	:CORRECT?
2059	031432	001440			BEQ	6\$:YES
2060							
2061	031434	023737	002136 017206		CMP	CDcnt,T.LMT	:CHECKED ENOUGH??
2062	031442	001002			BNE	333\$:NO
2063	031444				ESCAPE	SEG	:CHECK FOR FL:LOE, ELSE EXIT SEG
	031444	104010			EMT	C\$ESCAPE	
	031446	000116			.WORD	10001\$-	
2064	031450	005237	002136	333\$:	INC	CDcnt	:ACCOUNT FOR IT
2065							
2066	031454	005737	002130		TST	CHECK	:HEADER OR JUST DATA
2067	031460	001007			BNE	9\$:JUST DATA
2068	031462				ERRDF	36.,EM47,ERR10	:BAD DATA
	031462	104462			TRAP	T\$ERCODE	
	031464	000044			.WORD	36	
	031466	013264			.WORD	EM47	
	031470	015110			.WORD	ERR10	
2069	031472	005237	002130		INC	CHECK	:ACCOUNT FOR PRINT OF HEADER
2070	031476	000416			BR	6\$	
2071							
2072	031500			9\$:	PRINTB	#FRMT7,TMP1,GDDAT,BDDAT	
	031500	013746	002176		MOV	BDDAT,-(SP)	
	031504	013746	002174		MOV	GDDAT,-(SP)	
	031510	013746	002170		MOV	TMP1,-(SP)	
	031514	012746	016325		MOV	#FRMT7,-(SP)	
	031520	012746	000004		MOV	#4,-(SP)	
	031524	010600			MOV	SP,RO	
	031526	104014			EMT	C\$PNTB	
	031530	062706	000012		ADD	#12,SP	
2073	031534			6\$:	CKLOOP		
	031534	104006			EMT	C\$CLP1	
2074							
2075	031536	062737	000002 002172		ADD	#2,TMP2	:NEXT

TEST 29 - CHECK THROUGHPUT OF SILO

SEQ 0101

```

2076 031544 005237 002170      INC      TMP1      :NEXT
2077 031550 005237 002174      INC      GDDAT     :NEXT
2078 031554 023727 002170 000201  CMP      TMP1,#129. :DONE?
2079 031562 001315      BNE      4$
2080
2081 031564      ENDSEG      ;%%END OF SEGMENT%%
      031564      10001$:
      031564 104005      EMT      C$ESEG
2082
2083 031566      ENDSEG      ;%%END OF SEGMENT%%
      031566      10000$:
      031566 104005      EMT      C$ESEG
2084 031570      ENDTST      ;**END OF TEST**
      031570      L10062:
      031570 104001      EMT      C$ETST
2085
2086      .SBTTL  **TEST 30** - CHECK ZERO FILL ON WRITE
2087
2088 031572      BGNST      ;**START OF TEST**
2089
2090
2091
2092 031572      STARS
      ;:*****
      ;WHEN WRITING PARTIAL SECTORS (LESS THAN 128 WORDS) THE
      ;CONTROLLER WILL FILL IN THE REMAINING PORTION OF
      ;THE SECTOR WITH ZERO WORDS. CHECK THIS FEATURE
      ;WITH WORD COUNTS FROM 1 TO 127
      STARS
      ;:*****
2093
2094
2095
2096
2097 031572
2098
2099 031572 004737 021574      JSR      PC,HDHOME ;HEADS OVER TRACK 0
2100 031576      CKERFG      ;HEADS GO HOME OKAY
      031604 104032      EMT      C$EXIT
      031606 000442      .WORD    L10063-.
2101
2102 031610      BGNSEG      ;%%START OF SEGMENT%%
      031610 104004      EMT      C$BSEG
2103
2104 031612 012737 000001 002170      MOV      #1,TMP1   ;START WITH 1 WORD WRITE
2105 031620 012700 003260 35$:      MOV      #BUF,R0   ;WRITE BUFFER WITH 52525, WE'LL
2106 031624 012701 000200      MOV      #128,R1  ;WRITE 128 WORDS ALL THOUGH WE'RE
2107 031630 012720 052525 3$:      MOV      #52525,(R0)+ ;ONLY GOING TO TRANSFER < 128
2108 031634 005301      DEC      R1       ;DONE WITH BUFFER?
2109 031636 001374      BNE      3$       ;NO, GO BACK
2110 031640 013700 002170 33$:      MOV      TMP1,R0  ;GET TRANSFER WORD COUNT
2111 031644 005400      NEG      R0       ;NEGATE FOR RLMP
2112 031646 010077 150404      MOV      R0,@RLMP ;STORE WORD COUNT AWAY
2113 031652 012777 003260 150372      MOV      #BUF,@RLBA ;SET UP RLBA
2114 031660 005077 150370      CLR      @RLDA
2115 031664 004537 020674      JSR      R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
2116 031670 000012      WRITE     ;WRITE IT
2117 031672 004537 021514      JSR      R5,WTCRDY ;WAIT FOR WRITE TO FINISH
2118 031676      ESCAPE   SEG     ;CHECK FOR FL:LOE, ELSE EXIT SEG
      031676 104010      EMT      C$ESCAPE
      031700 000346      .WORD    10000$-.
2119

```

2120	031702	004537	020432		JSR	R5,CHERR		:CHECK CNTLR FOR ERRORS
2121	031706				ESCAPE	SEG		:CHECK FOR FL:LOE, ELSE EXIT SEG
	031706	104010			EMT	C\$ESCAPE		
	031710	000336			.WORD	10000\$-		
2122	031712				BGNSEG			:%%START OF SEGMENT%%
	031712	104004			EMT	C\$BSEG		
2123	031714	012700	003260		MOV	#BUF,R0		:WE'RE GOING TO OVERLAY BUFFER BEFORE
2124	031720	012701	000200		MOV	#128.,R1		:READING IT BACK.
2125	031724	012720	125252	18\$:	MOV	#125252,(R0)+		:OVERLAY IT WITH COMPLIMENT
2126	031730	005301			DEC	R1		:DONE?
2127	031732	001374			BNE	18\$:NO, KEEP GOING
2128								
2129	031734	012777	003260	150310	MOV	#BUF,@RLBA		:SET UP TO READ
2130	031742	012777	177600	150306	MOV	#-128.,@RLMP		:128 WORDS TO CHECK ZERO FILL
2131	031750	005077	150300		CLR	@RLDA		:SECTOR
2132	031754	004537	020674		JSR	R5,LDFUNC		:LOAD THE FUNCTION IN NEXT WORD
2133	031760	000014			READ			
2134	031762	004537	021514		JSR	R5,WTCRDY		:WAIT TIL WE FINISH THE READ
2135	031766				ESCAPE	SEG		:CHECK FOR FL:LOE, ELSE EXIT SEG
	031766	104010			EMT	C\$ESCAPE		
	031770	000234			.WORD	10001\$-		
2136								
2137	031772	004537	020432		JSR	R5,CHERR		:CHECK CNTLR FOR ERRORS
2138	031776	005737	002132		TST	T.CRC		:WAS ERROR A DCK??
2139	032002	001003			BNE	8\$:YES,SEE IF WE A DUMP
2140	032004			10\$:	ESCAPE	SEG		:CHECK FOR FL:LOE, ELSE EXIT SEG
	032004	104010			EMT	C\$ESCAPE		
	032006	000216			.WORD	10001\$-		
2141	032010	000404			BR	99\$:SKIP AROUND
2142	032012	005737	017204	8\$:	TST	T.DMP		:DO WE STILL WANT TO CHECK IT
2143	032016	001772			BEQ	10\$:NO
2144	032020				CKLOOP			:YES, CHECK FOR LOOP FIRST
	032020	104006			EMT	C\$CLP1		
2145	032022	005037	002136	99\$:	CLR	CDCNT		:CLEAR NUMBER WE'RE TO PRINT
2146	032026	005037	002130		CLR	CHECK		:ALLOW HEADER ON FIRST PRINT
2147	032032	013702	002170		MOV	TMP1,R2		:WORDS WRITTEN IN R2
2148	032036	012701	000200		MOV	#128.,R1		:CHECK 128 WORDS
2149								
2150	032042	012703	003260		MOV	#BUF,R3		:SET UP BUFFER BEGINNING
2151	032046	005037	002172		CLR	TMP2		:ZERO WORD COUNT
2152	032052	012737	052525	002174	MOV	#52525,GDDAT		:SET UP EXPECTED
2153	032060	011337	002176	4\$:	MOV	(R3),BDDAT		:GET WORD
2154	032064	023737	002176	002174	CMP	BDDAT,GDDAT		:IS WORD CORRECT?
2155	032072	001441			BEQ	12\$:YES, GO CHECK COUNTS AND REPEAT
2156								
2157	032074	023737	002136	017206	CMP	CDCNT,T.LMT		:CHECKED ENOUGH??
2158	032102	001002			BNE	333\$:NO
2159	032104				ESCAPE	SEG		:CHECK FOR FL:LOE, ELSE EXIT SEG
	032104	104010			EMT	C\$ESCAPE		
	032106	000116			.WORD	10001\$-		
2160	032110	005237	002136	333\$:	INC	CDCNT		:ACCOUNT FOR IT
2161								
2162	032114	005737	002130		TST	CHECK		:HEADER OR JUST DATA
2163	032120	001007			BNE	9\$:JUST DATA
2164	032122				ERRDF	37.,EM27,ERR12		
	032122	104462			TRAP	T\$ERCODE		
	032124	000045			.WORD	37		

```

2165 032126 012313          .WORD EM27
2166 032130 015234          .WORD ERR12
2167 032132 005237 002130  INC CHECK          ;ACCOUNT FOR PRINT OF HEADER
2168 032136 000417          BR 12$
2169 032140          9$: PRINTB #FRMT9,TMP1,R3,GDDAT,BDDAT
    032140 013746 002176  MOV BDDAT,-(SP)
    032144 013746 002174  MOV GDDAT,-(SP)
    032150 010346          MOV R3,-(SP)
    032152 013746 002170  MOV TMP1,-(SP)
    032156 012746 016520  MOV #FRMT9,-(SP)
    032162 012746 000005  MOV #5,-(SP)
    032166 010600          MOV SP,RO
    032170 104014          EMT C$PNTB
    032172 062706 000014  ADD #14,SP
2170 032176          12$: CKLOOP
    032176 104006          EMT C$CLP1
2171 032200 005723          6$: TST (R3)+
2172 032202 005237 002172  INC TMP2
2173 032206 005301          DEC R1          ;DONE ALL WORDS?
2174 032210 001405          BEQ 7$          ;EXIT TEST
2175 032212 005302          DEC R2          ;DONE CHECKING NON-ZERO WORDS
2176 032214 003321          BGT 4$          ;NO, BRANCH BACK
2177 032216 005037 002174  CLR GDDAT      ;YES, SET EXP'D AS ZERO
2178 032222 000716          BR 4$          ;BRANCH BACK
2179 032224          7$:          ;EXIT TEST
2180 032224          ENDSEG          ;%%END OF SEGMENT%%
    032224          10001$:
    032224 104005          EMT C$ESEG
2181 032226 005237 002170  INC TMP1
2182 032232 023727 002170 000200  CMP TMP1,#128.
2183 032240 001402          BEQ 34$
2184 032242 000137 031620  JMP 35$
2185 032246          34$:
2186 032246          ENDSEG          ;%%END OF SEGMENT%%
    032246          10000$:
    032246 104005          EMT C$ESEG
2187 032250          ENDTST          ;**END OF TEST**
    032250          L10063:
    032250 104001          EMT C$ETST
2188 032252          .SBTTL **TEST 31** - CHECK SECTOR BITS OF HEADER COMPARE
2189 032252          BGNTST          ;**START OF TEST**
2190
2191
2192
2193
2194
2195
2196 032252          STARS
2197          ;:*****
2198          ;TEST THAT ALL SECTOR BITS OF HEADER WORD CAN COMPARE
2199          ;UNIQUELY. WE TESTED THE HEADER COMPARE LOGIC EARLIER
2200          ;BUT THAT WAS NOT AN EXTENSIVE TEST OF THE SECTOR BITS.
2201          ;THE TEST PROCEDURE IS TO WRITE EACH SECTOR OF TRACK
2202          ;0 WITH THE SECTOR ADDRESS, THEN GO BACK AND READ
          ;EACH SECTOR. IF ANY SECTOR HAS ANY DATA THEN THAT
  
```



```

2203 ;WHICH WAS EXPECTED THEN WE HAVE AN ERROR
2204 ;ERROR PRINT OUT WILL GIVE SECTOR, EXPECTED AND RECEIVED
2205 032252 STARS
;:.....

2206
2207
2208
2209
2210
2211
2212
2213 032252 004737 021574 JSR PC,HDHOME ;HEADS OVER TRACK 0
2214 032256 CKERFG ;HEADS GO HOME OKAY
032264 104032 EMT C$EXIT
032266 000414 .WORD L10064-.

2215
2216 032270 BGNSEG ;%%START OF SEGMENT%%
032270 104004 EMT C$BSEG

2217
2218 032272 005037 002166 1$: CLR TMP0 ;CLEAR
2219
2220 032276 BGNSEG ;%%START OF SEGMENT%%
032276 104004 EMT C$BSEG

2221
2222 032300 012702 003260 199$: MOV #BUF,R2 ;WRITE A PATTERN FOR THE WRITE
2223 032304 012701 000200 MOV #128.,R1 ;ONE SECTOR'S WORTH
2224 032310 013722 002166 2$: MOV TMP0,(R2)+ ;WRITE IT
2225 032314 005301 DEC R1 ;DONE,
2226 032316 001374 BNE 2$ ;IF NOT, GO BACK
2227
2228 032320 012777 177600 147730 MOV #-128.,@RLMP ;ONE SECTOR WORD COUNT
2229 032326 012777 003260 147716 MOV #BUF,@RLBA ;WRITE FROM BUF
2230 032334 013777 002166 147712 MOV TMP0,@RLDA ;SECTOR
2231 032342 004537 020674 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
2232 032346 000012 WRITE
2233 032350 004537 021514 JSR R5,WTCRDY ;WAIT FOR WRITE TO FINISH
2234 032354 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
032354 104010 EMT C$ESCAPE
032356 000320 .WORD 10001$-.

2235 032360 005237 002166 INC TMP0 ;NEXT SECTOR
2236 032364 023727 002166 000050 CMP TMP0,#40. ;ALL DONE?
2237 032372 001342 BNE 199$ ;NO GO BACK
2238 032374 005037 002166 CLR TMP0 ;CLEAR
2239
2240 032400 BGNSEG ;%%START OF SEGMENT%%
032400 104004 EMT C$BSEG

2241
2242 032402 012702 003260 98$: MOV #BUF,R2 ;CLEAR THE BUFFER FIRST
2243 032406 012701 000200 MOV #128.,R1 ;128 WORDS
2244 032412 005022 3$: CLR (R2)+
2245 032414 005301 DEC R1
2246 032416 001375 BNE 3$
2247
2248 032420 013777 002166 147626 MOV TMP0,@RLDA ;GET SECTOR
2249 032426 012777 003260 147616 MOV #BUF,@RLBA ;SETUP BUS ADDRESS
2250
2251 032434 012777 177600 147614 MOV #-128.,@RLMP ;READ A SECTOR
  
```

```

2252 032442 004537 020674      JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
2253 032446 000014              READ
2254 032450 004537 021514      JSR      R5,WTCRDY
2255 032454              ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      032454 104010              EMT      C$ESCAPE
      032456 000216              .WORD   10002$-.

2256
2257 032460 004537 020432      JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
2258 032464 005737 002132      TST     T.CRC             ;WAS ERROR A DCK??
2259 032470 001003              BNE     8$                ;YES,SEE IF WE A DUMP
2260 032472 10$:  ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      032472 104010              EMT      C$ESCAPE
      032474 000200              .WORD   10002$-.

2261 032476 000404              BR      99$              ;SKIP AROUND
2262 032500 005737 017204      8$:  TST     T.DMP          ;DO WE STILL WANT TO CHECK IT
2263 032504 001772              BEQ     10$              ;NO
2264 032506              CKLOOP
      032506 104006              EMT      C$CLP1          ;YES, CHECK FOR LOOP FIRST

2265
2266              ;CHECK NOW TO SEE IF WE READ THE RIGHT SECTOR
2267
2268 032510 005037 002136      99$:  CLR     CDCNT          ;CLEAR NUMBER WE'RE TO PRINT
2269 032514 005037 002130      CLR     CHECK            ;ALLOW HEADER ON FIRST PRINT
2270 032520 013737 002166 002174  MOV     TMPO,GDDAT        ;EXPECTED DATA
2271 032526 012702 003260      MOV     #BUF,R2          ;BUFFER
2272 032532 012701 000200      MOV     #128.,R1         ;WORD COUNT
2273 032536 012237 002176      5$:  MOV     (R2)+,BDDAT      ;
2274 032542 023737 002176 002174  CMP     BDDAT,GDDAT
2275 032550 001440              BEQ     6$
2276
2277 032552 023737 002136 017206  CMP     CDCNT,T.LMT      ;CHECKED ENOUGH??
2278 032560 001002              BNE     333$             ;NO
2279 032562              ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      032562 104010              EMT      C$ESCAPE
      032564 000110              .WORD   10002$-.

2280 032566 005237 002136      333$: INC     CDCNT          ;ACCOUNT FOR IT
2281
2282 032572 005737 002130      TST     CHECK            ;HEADER OR JUST DATA
2283 032576 001007              BNE     9$                ;JUST DATA
2284 032600              ERRDF  38.,EM50,ERR11    ;
      032600 104462              TRAP    T$ERCODE
      032602 000046              .WORD   38
      032604 013313              .WORD   EM50
      032606 015162              .WORD   ERR11

2285 032610 005237 002130      INC     CHECK            ;ACCOUNT FOR PRINT OF HEADER
2286 032614 000416              BR      6$
2287
2288 032616      9$:  PRINTB #FRMT8, TMPO, GDDAT, BDDAT
      032616 013746 002176      MOV     BDDAT, -(SP)
      032622 013746 002174      MOV     GDDAT, -(SP)
      032626 013746 002166      MOV     TMPO, -(SP)
      032632 012746 016377      MOV     #FRMT8, -(SP)
      032636 012746 000004      MOV     #4, -(SP)
      032642 010600              MOV     SP, R0
      032644 104014              EMT     C$PNTB
      032646 062706 000012      ADD     #12, SP

2289 032652      6$:  CKLOOP
  
```

```

2290 032652 104006          EMT      C$CLP1
2291 032654 005301          DEC      R1          ;ALL OF SECTOR CHECKED?
2292 032656 001327          BNE      5$          ;GO BACK IF NOT
2293 032660 005237 002166    INC      TMOPO        ;NEXT SECTOR
2294 032664 023727 002166 000050  CMP      TMOPO,#40.   ;DONE?
2295 032672 001243          BNE      98$         ;NO, GO BACK
2296
2297 032674          ENDSEG          ;%%END OF SEGMENT%%
      032674          10002$:
      032674 104005          EMT      C$ESEG
2298
2299 032676          ENDSEG          ;%%END OF SEGMENT%%
      032676          10001$:
      032676 104005          EMT      C$ESEG
2300 032700          ENDSEG          ;%%END OF SEGMENT%%
      032700          10000$:
      032700 104005          EMT      C$ESEG
2301 032702          ENDTST          ;**END OF TEST**
      032702          L10064:
      032702 104001          EMT      C$ETST
2302
2303          .SBTTL  **TEST 32** - WRITE CHECK NPR INTEGRITY
2304
2305 032704          BGNTST          ;**START OF TEST**
2306
2307 032704          STARS
      ;:*****
2308          ;CHECK THAT NPR WILL NOT INTERFERE WITH THE OPERATION OF THE
2309          ;UNIBUS. WE SET UP LOCATION 4 TO HANDLE THE TRAP IF IT HAPPENS.
2310 032704          STARS
      ;:*****
2311
2312
2313 032704 004737 021574    JSR      PC,HDHOME    ;HEADS OVER TRACK 0
2314 032710          CKERFG          ;HEADS GO HOME OKAY
      032716 104032          EMT      C$EXIT
      032720 000372          .WORD    L10065-.
2315
2316 032722          BGNSEG          ;%%START OF SEGMENT%%
      032722 104004          EMT      C$BSEG
2317
2318 032724 012700 003260    MOV      #BUF,R0      ;SETUP AND WRITE
2319 032730 012701 000200    MOV      #128.,R1     ;128 WORDS
2320 032734 012720 125252    299$:  MOV      #125252,(R0)+ ;WRITE
2321 032740 005301          DEC      R1          ;DONE??
2322 032742 001374          BNE      299$
2323
2324 032744 012777 003260 147300  MOV      #BUF,@RLBA   ;LOAD BUS ADDRESS
2325 032752 012777 177600 147276  MOV      #-128.,@RLMP ;WORD COUNT
2326 032760 005077 147270    CLR      @RLDA        ;CLEAR DISK ADDRESS
2327 032764 004537 020674    JSR      R5,LDFUNC    ;LOAD THE FUNCTION IN NEXT WORD
2328 032770 000012          WRITE
2329 032772 004537 021514    JSR      R5,WTCRDY   ;WAIT FOR CONTROLLER READY
2330 032776          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      032776 104010          EMT      C$ESCAPE
      033000 000310          .WORD    10000$-.
  
```

```

2331 033002 004537 020432      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
2332 033006      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      033006 104010      EMT      C$ESCAPE
      033010 000300      .WORD   10000$-.
2333
2334
2335      ;VERIFY WRITE WITH READ BEFORE WRCHK
2336
2337 033012 005077 147236      CLR      @RLDA
2338 033016 012777 003260 147226      MOV      #BUF,@RLBA
2339 033024 012777 177600 147224      MOV      #-128.,@RLMP
2340 033032 004537 020674      JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
2341 033036 000014      READ
2342 033040 004537 021514      JSR      R5,WTCRDY
2343 033044      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      033044 104010      EMT      C$ESCAPE
      033046 000242      .WORD   10000$-.
2344 033050 004537 020432      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
2345 033054      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
      033054 104010      EMT      C$ESCAPE
      033056 000232      .WORD   10000$-.
2346
2347 033060      BGNSEG      ;%%START OF SEGMENT%%
      033060 104004      EMT      C$BSEG
2348
2349 033062      1$: SETVEC ERRVEC,#TRPHAN,#340      ;SET UP FOR TRAP
      033062 012746 000340      MOV      #340,-(SP)
      033066 012746 021566      MOV      #TRPHAN,-(SP)
      033072 013746 002140      MOV      ERRVEC,-(SP)
      033076 012746 000003      MOV      #3,-(SP)
      033102 104037      EMT      C$SVEC
      033104 062706 000010      ADD      #10,SP
2350 033110 005037 002150      CLR      TRPFLG      ;CLEAR TRAP OCCURANCE
2351 033114 012777 003260 147130      MOV      #BUF,@RLBA      ;BUS ADDRESS
2352 033122 005077 147126      CLR      @RLDA      ;LOAD DISK ADDRESS
2353 033126 012777 177600 147122      MOV      #-128.,@RLMP      ;WORD COUNT OF 128
2354 033134 005037 002174      CLR      GDDAT      ;SET UP CSR TO LOAD
2355 033140 013737 002142 002174      MOV      DRIVE,GDDAT      ;SET IN DRIVE
2356 033146 052737 000002 002174      BIS      #WRCHK,GDDAT      ;SET IN FUNCTION
2357 033154 004537 021202      JSR      R5,BEFORE      ;LOAD FOR ERROR PRINTOUT
2358 033160 013737 002174 002224      MOV      GDDAT,B.CS      ;SET IN COMMAND
2359 033166 052737 000201 002224      BIS      #201,B.CS      ;LOAD CRDY
2360 033174 042737 002000 002224      BIC      #OPI,B.CS      ;CLEAR (BIT 10)
2361 033202 013777 002174 147040      MOV      GDDAT,@RLCS      ;ISSUE WRITE CHECK
2362 033210 012701 000144      MOV      #100.,R1      ;WAIT FOR CRDY
2363 033214 032777 000200 147026 5$: BIT      #CRDY,@RLCS      ;NPR DONE
2364 033222 001013      BNE      6$      ;YES, 6$
2365 033224      WAITUS #20.      ;WAIT A WHILE
      033224 012700 000024      MOV      #20.,R0
      033230 104027      EMT      C$WTU
2366 033232 005301      DEC      R1      ;A WHILE UP
2367 033234 001367      BNE      5$      ;NO, GO BACK
2368
2369 033236 004537 021234      JSR      R5,AFTER
2370 033242      ERRDF 0.,CRTIM,ERR5      ;CONTROLLER TIMED OUT
      033242 104462      TRAP   T$ERCODE
      033244 000000      .WORD   0

```

```

2371 033246 007400          .WORD CRTIM
      033250 014664          .WORD ERR5
      033252 013700 002140 6$: CLRVEC          ;CLEAR VECTOR
      033252 013700 002140 MOV          ERRVEC,RO
      033256 104036          EMT          C$CVEC
2372 033260          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      033260 104010          EMT          C$ESCAPE
      033262 000024          .WORD 10001$-.
2373
2374 033264 005737 002150 TST          TRPFLG          ;DID TRAP OCCUR?
2375 033270 001406          BEQ          7$          ;NO
2376 033272 004537 021234 JSR          R5,AFTER
2377 033276          ERRSF 1.,EM57,ERRO          ;TRAP ON WRITE
      033276 104461          TRAP          T$ERCODE
      033300 000001          .WORD 1
      033302 013705          .WORD EM57
      033304 014452          .WORD ERRO
2378 033306          7$:
2379
2380
2381 033306          ENDSEG          ;%%END OF SEGMENT%%
      033306 104005          10001$: EMT          C$ESEG
2382 033310          ENDSEG          ;%%END OF SEGMENT%%
      033310 104005          10000$: EMT          C$ESEG
2383
2384 033312          ENDTST          ;**END OF TEST**
      033312 104001          L10065: EMT          C$ETST
2385
2386          .SBTTL **TEST 33** - WRITE CHECK FUNCTION
2387
2388 033314          BGNTST          ;**START OF TEST**
2389
2390 033314          STARS
      ;:*****
2391          ;CHECK OF WRITE CHECK LOGIC UNDER FLAG MODE
2392          ; WE WILL WRITE CHECK A FULL SECTOR (128 WORDS) FROM
2393          ;MEMORY (BUF). WE CHECK THAT NO ERRORS OCCUR.
2394 033314          STARS
      ;:*****
2395
2396
2397 033314 004737 021574 JSR          PC,HDHOME          ;HEADS OVER TRACK 0
2398 033320          CKERFG          ;HEADS GO HOME OKAY
      033326 104032          EMT          C$EXIT
      033330 000214          .WORD L10066-.
2399
2400 033332          BGNSEG          ;%%START OF SEGMENT%%
      033332 104004          EMT          C$BSEG
2401
2402 033334 012700 003260 MOV          #BUF,RO          ;SETUP AND WRITE
2403 033340 012701 000200 MOV          #128.,R1          ;128 WORDS
2404 033344 012720 125252 299$: MOV          #125252,(RO)+          ;WRITE
2405 033350 005301          DEC          R1          ;DONE??
2406 033352 001374          BNE          299$
  
```

```
2407
2408 033354 012777 003260 146670      MOV    #BUF,@RLBA      ;LOAD BUS ADDRESS
2409 033362 012777 177600 146666      MOV    #-128.,@RLMP   ;WORD COUNT
2410 033370 005077 146660                CLR    @RLDA           ;CLEAR DISK ADDRESS
2411 033374 004537 020674                JSR    R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
2412 033400 000012                WRITE
2413 033402 004537 021514                JSR    R5,WTCRDY     ;WAIT FOR CONTROLLER READY
2414 033406                ESCAPE SEG           ;CHECK FOR FL:LOE, ELSE EXIT SEG
      033406 104010                EMT    C$ESCAPE
      033410 000132                .WORD 10000$-
2415 033412 004537 020432                JSR    R5,CHERR      ;CHECK CNTLR FOR ERRORS
2416 033416                ESCAPE SEG           ;CHECK FOR FL:LOE, ELSE EXIT SEG
      033416 104010                EMT    C$ESCAPE
      033420 000122                .WORD 10000$-
2417 033422                BGNSEG                ;%%START OF SEGMENT%%
      033422 104004                EMT    C$BSEG
2418
2419                ;VERIFY WRITE WITH READ BEFORE WRCHK
2420
2421 033424 005077 146624                CLR    @RLDA
2422 033430 012777 003260 146614      MOV    #BUF,@RLBA
2423 033436 012777 177600 146612      MOV    #-128.,@RLMP
2424 033444 004537 020674                JSR    R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
2425 033450 000014                READ
2426 033452 004537 021514                JSR    R5,WTCRDY     ;CHECK FOR FL:LOE, ELSE EXIT SEG
2427 033456                ESCAPE SEG
      033456 104010                EMT    C$ESCAPE
      033460 000060                .WORD 10001$-
2428 033462 004537 020432                JSR    R5,CHERR      ;CHECK CNTLR FOR ERRORS
2429 033466                ESCAPE SEG           ;CHECK FOR FL:LOE, ELSE EXIT SEG
      033466 104010                EMT    C$ESCAPE
      033470 000050                .WORD 10001$-
2430
2431 033472                BGNSEG                ;%%START OF SEGMENT%%
      033472 104004                EMT    C$BSEG
2432
2433 033474                3$:
2434 033474 005077 146554                CLR    @RLDA
2435 033500 012777 177600 146550      MOV    #-128.,@RLMP   ;WORD COUNT
2436 033506 012777 003260 146536      MOV    #BUF,@RLBA    ;BUS ADDRESS
2437 033514 004537 020674                JSR    R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
2438 033520 000002                WRCHK                ;WRITE CHECK
2439
2440 033522 004537 021514                JSR    R5,WTCRDY     ;WAIT FOR CONTROLLER READY
2441 033526                ESCAPE SEG           ;CHECK FOR FL:LOE, ELSE EXIT SEG
      033526 104010                EMT    C$ESCAPE
      033530 000006                .WORD 10002$-
2442
2443
2444 033532 004537 020432                JSR    R5,CHERR      ;CHECK CNTLR FOR ERRORS
2445
2446 033536                ENDSEG                ;%%END OF SEGMENT%%
      033536                10002$:
      033536 104005                EMT    C$ESEG
2447 033540                ENDSEG                ;%%END OF SEGMENT%%
      033540                10001$:
      033540 104005                EMT    C$ESEG
```

```
2448 033542          ENDSEG          ;%%END OF SEGMENT%%
      033542          10000$:
2449 033544 104005    EMT          C$ESEG          ;**END OF TEST**
      033544          L10066:
      033544 104001    EMT          C$ETST

2450          .SBTTL  **TEST 34** - WRITE CHECK FUNCTION INTERRUPT
2451
2452
2453 033546          BGNST          ;**START OF TEST**
2454
2455 033546          STARS
      ;*****
      ;CHECK OF WRITE CHECK LOGIC UNDER INTERRUPT MODE
      ;WE WILL WRITE CHECK A FULL SECTOR (128 WORDS) FROM MEMORY (BUF).
      ;WE CHECK THAT NO ERRORS OCCUR. WE DO NOT CHECK RLDA OR RLBA
      ;INCREMENT AT THIS TIME.
      STARS
      ;*****

2461
2462
2463 033546 004737 021574      JSR          PC,HDHOME          ;HEADS OVER TRACK 0
2464 033552          CKERFG          ;HEADS GO HOME OKAY
      033560 104032      EMT          C$EXIT
      033562 000252      .WORD          L10067-.

2465
2466 033564          BGNSEG          ;%%START OF SEGMENT%%
      033564 104004      EMT          C$BSEG

2467
2468 033566 012700 003260      MOV          #BUF,R0          ;SETUP AND WRITE
2469 033572 012701 000200      MOV          #128.,R1          ;128 WORDS
2470 033576 012720 125252      299$: MOV          #125252,(R0)+      ;WRITE
2471 033602 005301          DEC          R1          ;DONE??
2472 033604 001374          BNE          299$

2473
2474 033606 012777 003260 146436      MOV          #BUF,@RLBA          ;LOAD BUS ADDRESS
2475 033614 012777 177600 146434      MOV          #-128.,@RLMP          ;WORD COUNT
2476 033622 005077 146426          CLR          @RLDA          ;CLEAR DISK ADDRESS
2477 033626 004537 020674          JSR          R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
2478 033632 000012          WRITE
2479 033634 004537 021514          JSR          R5,WTCRDY          ;WAIT FOR CONTROLLER READY
2480 033640          ESCAPE          SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      033640 104010      EMT          C$ESCAPE
      033642 000170      .WORD          10000$-.

2481 033644 004537 020432          JSR          R5,CHERR          ;CHECK CNTLR FOR ERROR
2482 033650          ESCAPE          SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      033650 104010      EMT          C$ESCAPE
      033652 000160      .WORD          10000$-.

2483          ;VERIFY WRITE WITH READ BEFORE WRCHK
2484
2485 033654 005077 146374          CLR          @RLDA
2486 033660 012777 003260 146364      MOV          #BUF,@RLBA
2487 033666 012777 177600 146362      MOV          #-128.,@RLMP
2488 033674 004537 020674          JSR          R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
2489 033700 000014          READ
2490 033702 004537 021514          JSR          R5,WTCRDY
2491 033706          ESCAPE          SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
```

```

033706 104010 EMT C$ESCAPE
033710 000122 .WORD 10000$-
2492 033712 004537 020432 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
2493 033716 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
033716 104010 EMT C$ESCAPE
033720 000112 .WORD 10000$-
2494
2495 033722 BGNSEG ;%%START OF SEGMENT%%
033722 104004 EMT C$BSEG
2496
2497
2498 033724 005037 002152 CLR INTFLG ;CLEAR INTERRUPT OCCURANCE FLAG
2499 033730 005077 146320 CLR @RLDA
2500 033734 012777 177600 146314 MOV #-128.,@RLMP ;SET UP WORD COUNT
2501 033742 012777 003260 146302 MOV #BUF,@RLBA ;SET UP BUS ADDRESS
2502
2503 033750 SETPRI #PRI00 ;PRIORITY TO 0
033750 012700 000000 MOV #PRI00,R0
033754 104041 EMT C$SPRI
2504 033756 004537 020674 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
2505 033762 000102 WRCHK!INTEN ;WRITE CHECK UNDER INTERRUPT
2506 033764 004537 021514 JSR R5,WTCRDY ;WAIT FOR INTERRUPT
2507 033770 ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
033770 104010 EMT C$ESCAPE
033772 000036 .WORD 10001$-
2508
2509 033774 SETPRI #PRI07 ;SET PRIORITY TO 7
033774 012700 000340 MOV #PRI07,R0
034000 104041 EMT C$SPRI
2510 034002 005737 002152 TST INTFLG ;DID INTERRUPT OCCUR?
2511 034006 001004 BNE 2$ ;YES-BRANCH NO-REPORT
2512
2513 034010 ERRDF 4.,EM60,ERRO ;WRITE DID NOT INTERRUPT
034010 104462 TRAP T$ERCODE
034012 000004 .WORD 4
034014 013745 .WORD EM60
034016 014452 .WORD ERRO
2514 034020 2$: ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
034020 104010 EMT C$ESCAPE
034022 000006 .WORD 10001$-
2515
2516 034024 004537 020432 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
2517
2518 034030 ENDSEG ;%%END OF SEGMENT%%
034030 10001$:
034030 104005 EMT C$ESEG
2519 034032 ENDSEG ;%%END OF SEGMENT%%
034032 10000$:
034032 104005 EMT C$ESEG
2520 034034 ENDTST ;**END OF TEST**
034034 L10067:
034034 104001 EMT C$ETST
2521
2522 .SBTTL **TEST 35** - PROPER INCREMENT OF RLBA ON WRITE CHECK
2523
2524 034036 BGNST ;**START OF TEST**
2525

```


2526
2527 034036

2528
2529
2530
2531
2532 034036

2533
2534
2535 034036 004737 021574
2536 034042 104032
034050 000256
034052

2537
2538 034054 104004
034054

2539
2540 034056 012700 003260
2541 034062 012701 000200
2542 034066 012720 125252
2543 034072 005301
2544 034074 001374

2545
2546 034076 012777 003260 146146
2547 034104 012777 177600 146144

2548 034112 005077 146136
2549 034116 004537 020674
2550 034122 000012
2551 034124 004537 021514
2552 034130 104010
034130 000174
034132

2553 034134 004537 020432
2554 034140 104010
034140 000164

2555
2556
2557 034144 005077 146104
2558 034150 012777 003260 146074
2559 034156 012777 177600 146072

2560 034164 004537 020674
2561 034170 000014
2562 034172 004537 021514
2563 034176 104010
034176 000126
034200

2564 034202 004537 020432
2565 034206 104010
034206 000116
034210

2566
2567 034212 104004
034212

2568

STARS

:CHECK THAT THE RLBA WILL INCREMENT PROPERLY AFTER THE
:WRITE CHECK WAS FINISHED THE RLBA SHOULD BE 128 WORDS (256 BYTES)
:CREATED. STARTING RLBA IS "BUF", ENDING SHOULD BE "BUF + 256."
:WE WILL MONITOR ALL ERRORS AND REPORT THEM ACCORDINGLY
STARS

JSR PC,HDHOME ;HEADS OVER TRACK 0
CKERFG ;HEADS GO HOME OKAY
EMT C\$EXIT
.WORD L10070-.
BGNSEG ;%%START OF SEGMENT%%
EMT C\$BSEG
MOV #BUF,R0 ;SETUP AND WRITE
MOV #128.,R1 ;128 WORDS
299\$: MOV #125252,(R0)+ ;WRITE
DEC R1 ;DONE??
BNE 299\$
MOV #BUF,@RLBA ;LOAD BUS ADDRESS
MOV #-128.,@RLMP ;WORD COUNT
CLR @RLDA ;CLEAR DISK ADDRESS
JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
WRITE
JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY
ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
EMT C\$ESCAPE
.WORD 10000\$-.
JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
EMT C\$ESCAPE
.WORD 10000\$-.
;VERIFY WRITE WITH READ BEFORE WRCHK
CLR @RLDA
MOV #BUF,@RLBA
MOV #-128.,@RLMP
JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
READ
JSR R5,WTCRDY
ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
EMT C\$ESCAPE
.WORD 10000\$-.
JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
EMT C\$ESCAPE
.WORD 10000\$-.
BGNSEG ;%%START OF SEGMENT%%
EMT C\$BSEG

```

2569 034214          3$:
2570 034214 005077 146034          CLR @RLDA
2571 034220 012777 003260 146024      MOV #BUF,@RLBA      ;SET UP BUS ADDRESS
2572 034226 012777 177600 146022      MOV #-128.,@RLMP   ;WORD COUNT
2573 034234 012737 003260 002174      MOV #BUF,GDDAT    ;FORM EXPECTED BUS ADDRESS
2574 034242 062737 000400 002174      ADD #256.,GDDAT   ;AFTER WRITE
2575
2576 034250 004537 020674          JSR R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
2577 034254 000002          WRCHK             ;WRITE CHECK
2578 034256 004537 021514          JSR R5,WTCRDY     ;WAIT FOR CONTROLLER READY
2579 034262          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      034262 104010          EMT C$ESCAPE
      034264 000040          .WORD 10001$-.
2580
2581 034266 004537 020432          JSR R5,CHERR      ;CHECK CNTLR FOR ERRORS
2582 034272          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      034272 104010          EMT C$ESCAPE
      034274 000030          .WORD 10001$-.
2583 034276 017737 145750 002176      MOV @RLBA,BDDAT   ;READ 'RLBA' FOR PRESENT ADDRESS
2584 034304 023737 002176 002174      CMP BDDAT,GDDAT  ;DID 'BA' INCREMENT PROPERLY?
2585 034312 001404          BEQ 2$           ;YES, CONTINUE
2586
2587 034314          ERRDF 5.,EM61,ERR4   ;BA DID NOT INCREMENT
      034314 104462          TRAP T$ERCODE
      034316 000005          .WORD 5
      034320 014003          .WORD EM61
      034322 014616          .WORD ERR4
2588
2589 034324          2$:
2590
2591 034324          ENDSEG              ;%%END OF SEGMENT%%
      034324          10001$:
      034324 104005          EMT C$ESEG
2592 034326          ENDSEG              ;%%END OF SEGMENT%%
      034326          10000$:
      034326 104005          EMT C$ESEG
2593 034330          ENDTST              ;**END OF TEST**
      034330          L10070:
      034330 104001          EMT C$ETST
2594
2595          .SBTTL **TEST 36** - PROPER INCREMENT OF RLDA ON WRITE CHECK
2596
2597 034332          BGNTST              ;**START OF TEST**
2598
2599 034332          STARS
      ;*****
2600          ;CHECK THAT THE SECTOR INCREMENTS AFTER THE WRITE CHECK WAS FINISHED.
2601          ;A FULL SECTOR WRITE CHECK THE RLDA SHOULD REFLECT AN INCREMENT
2602          ;OF THE SECOTR. "GDDAT" WAS THE EXPECTED RLDA.
2603 034332          STARS
      ;*****
2604
2605
2606 034332 004737 021574          JSR PC,HDHOME     ;HEADS OVER TRACK 0
2607 034336          CKERFG             ;HEADS GO HOME OKAY
      034344 104032          EMT C$EXIT
      034346 000254          .WORD L10071-.
  
```

```

2608
2609 034350          BGNSEG          ;%%START OF SEGMENT%%
      034350 104004  EMT          C$BSEG
2610
2611 034352 012700 003260          MOV      #BUF,R0          ;SETUP AND WRITE
2612 034356 012701 000200          MOV      #128.,R1         ;128 WORDS
2613 034362 012720 125252          299$: MOV      #125252,(R0)+ ;WRITE
2614 034366 005301          DEC      R1              ;DONE??
2615 034370 001374          BNE     299$
2616
2617 034372 012777 003260 145652  MOV      #BUF,@RLBA      ;LOAD BUS ADDRESS
2618 034400 012777 177600 145650  MOV      #-128.,@RLMP    ;WORD COUNT
2619 034406 005077 145642          CLR      @RLDA          ;CLEAR DISK ADDRESS
2620 034412 004537 020674          JSR     R5,LDFUNC       ;LOAD THE FUNCTION IN NEXT WORD
2621 034416 000012          WRITE
2622 034420 004537 021514          JSR     R5,WTCRDY      ;WAIT FOR CONTROLLER READY
2623 034424          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      034424 104010  EMT          C$ESCAPE
      034426 000172  .WORD 10000$-.
2624 034430 004537 020432          JSR     R5,CHERR       ;CHECK CNTLR FOR ERRORS
2625 034434          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      034434 104010  EMT          C$ESCAPE
      034436 000162  .WORD 10000$-.
2626          ;VERIFY WRITE WITH READ BEFORE WRCHK
2627
2628 034440 005077 145610          CLR      @RLDA
2629 034444 012777 003260 145600  MOV      #BUF,@RLBA
2630 034452 012777 177600 145576  MOV      #-128.,@RLMP
2631 034460 004537 020674          JSR     R5,LDFUNC       ;LOAD THE FUNCTION IN NEXT WORD
2632 034464 000014          READ
2633 034466 004537 021514          JSR     R5,WTCRDY
2634 034472          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      034472 104010  EMT          C$ESCAPE
      034474 000124  .WORD 10000$-.
2635 034476 004537 020432          JSR     R5,CHERR       ;CHECK CNTLR FOR ERRORS
2636 034502          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      034502 104010  EMT          C$ESCAPE
      034504 000114  .WORD 10000$-.
2637
2638 034506          BGNSEG          ;%%START OF SEGMENT%%
      034506 104004  EMT          C$BSEG
2639
2640 034510          3$:
2641 034510 005037 002174          CLR      GDDAT
2642 034514 013777 002174 145532  MOV      GDDAT,@RLDA    ;SETUP DISK ADDRESS
2643 034522 005237 002174          INC      GDDAT          ;CREATE EXPECTED SECTOR
2644 034526 012777 177600 145522  MOV      #-128.,@RLMP    ;WORD COUNT
2645 034534 012777 003260 145510  MOV      #BUF,@RLBA     ;SETUP BUS ADDRESS
2646
2647 034542 004537 020674          JSR     R5,LDFUNC       ;LOAD THE FUNCTION IN NEXT WORD
2648 034546 000002          WRCHK          ;WRITE CHECK
2649 034550 004537 021514          JSR     R5,WTCRDY      ;WAIT FOR CONTROLLER READY
2650 034554          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      034554 104010  EMT          C$ESCAPE
      034556 000040  .WORD 10001$-.
2651
2652 034560 004537 020432          JSR     R5,CHERR       ;CHECK CNTLR FOR ERRORS
  
```

```

2653 034564          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      034564 104010  EMT C$ESCAPE
      034566 000030  .WORD 10001$-.
2654
2655 034570 013737 002240 002176  MOV E.DA,BDDAT ;READ DISK ADDRESS
2656 034576 023737 002174 002176  CMP GDDAT,BDDAT ;DID SECTOR INCREMENT PROPERLY
2657 034604 001404          BEQ 2$          ;YES, BRANCH NO, REPORT ERROR
2658
2659 034606          ERRDF 6.,EM62,ERR4 ;DA DID NOT INCREMENT
      034606 104462  TRAP T$ERCODE
      034610 000006  .WORD 6
      034612 014060  .WORD EM62
      034614 014616  .WORD ERR4
2660
2661 034616          2$:
2662
2663 034616          ENDSEG          ;%%END OF SEGMENT%%
      034616 104005 10001$:
2664 034620          EMT C$ESEG
      034620 104005  ENDSEG          ;%%END OF SEGMENT%%
      034620 104005 10000$:
2665 034622          EMT C$ESEG
      034622          ENDTST          ;**END OF TEST**
      034622          L10071:
      034622 104001  EMT C$ETST
2666
2667
2668 .SBTTL **TEST 37** - MULTIPLE SECTOR WRITE CHECK
2669
2670 034624          BGNTST          ;**START OF TEST**
2671
2672 034624          STARS
2673          ;*****
2674          ;CHECK FOR MULTIPLE SECTOR WRITE CHECK. THIS TEST CHECKS
2675          ;THAT TWO SECTORS CAN BE SUCCESSFULLY CHECKED. WE LOAD
2676          ;A WORD COUNT OF 129 WORDS (ONE SECTOR + 1 WORD) STARTING AT
2677          ;SECTOR 0 THRU SECTOR 37 AND VERIFY THAT THE RLDA DOES
2678 034624          ;A DOUBLE INCREMENT EACH TIME.
2679          STARS
2680          ;*****
2681
2682 034624 004737 021574  JSR PC,HDHOME ;HEADS OVER TRACK 0
2683 034630          CKERFG          ;HEADS GO HOME OKAY
      034636 104032  EMT C$EXIT
      034640 000354  .WORD L10072-.
2684
2685 034642          BGNSEG          ;%%START OF SEGMENT%%
      034642 104004  EMT C$BSEG
2686
2687 034644 012737 000000 002166  MOV #0,TMP0
2688 034652 012737 000000 002170  MOV #0,TMP1
2689 034660 012700 003260          MOV #BUF,R0 ;SETUP AND WRITE
2690 034664 012701 000201          MOV #129,R1 ;129 WORDS
2691 034670 012720 125252 299$: MOV #125252,(R0)+ ;WRITE
2692 034674 005301          DEC R1 ;DONE??
  
```

```

2693 034676 001374          BNE      299$
2694
2695 034700 012777 003260 145344 1$:  MOV     #BUF,@RLBA      ;LOAD BUS ADDRESS
2696 034706 012777 177577 145342      MOV     #-129.,@RLMP    ;WORD COUNT
2697 034714 013737 002170 002174      MOV     TMP1,GDDAT
2698 034722 053737 002166 002174      BIS     TMPO,GDDAT
2699 034730 013777 002174 145316      MOV     GDDAT,@RLDA
2700 034736 004537 020674          JSR     R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
2701 034742 000012          WRITE
2702 034744 004537 021514          JSR     R5,WTCRDY     ;WAIT FOR CONTROLLER READY
2703 034750          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      034750 104010          EMT     C$ESCAPE
      034752 000240          .WORD  10000$-.
2704 034754 004537 020432          JSR     R5,CHERR      ;CHECK CNTLR FOR ERRORS
2705 034760          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      034760 104010          EMT     C$ESCAPE
      034762 000230          .WORD  10000$-.

2706
2707          ;VERIFY WRITE WITH READ BEFORE WRCHK
2708
2709 034764 013737 002170 002174      MOV     TMP1,GDDAT
2710 034772 053737 002166 002174      BIS     TMPO,GDDAT
2711 035000 013777 002174 145246      MOV     GDDAT,@RLDA
2712 035006 012777 003260 145236      MOV     #BUF,@RLBA
2713 035014 012777 177577 145234      MOV     #-129.,@RLMP
2714 035022 004537 020674          JSR     R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
2715 035026 000014          READ
2716 035030 004537 021514          JSR     R5,WTCRDY     ;CHECK FOR FL:LOE, ELSE EXIT SEG
2717 035034          ESCAPE  SEG
      035034 104010          EMT     C$ESCAPE
      035036 000154          .WORD  10000$-.
2718 035040 004537 020432          JSR     R5,CHERR      ;CHECK CNTLR FOR ERRORS
2719 035044          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      035044 104010          EMT     C$ESCAPE
      035046 000144          .WORD  10000$-.

2720
2721 035050          BGNSEG
      035050 104004          EMT     C$BSEG      ;%%START OF SEGMENT%%

2722
2723
2724 035052 013737 002170 002174      MOV     TMP1,GDDAT      ;GET CYLINDER
2725 035060 053737 002166 002174      BIS     TMPO,GDDAT      ;GET SECTOR
2726 035066 013777 002174 145160      MOV     GDDAT,@RLDA     ;SET DISK ADDRESS-SECTOR 0
2727 035074 062737 000002 002174      ADD     #2,GDDAT        ;SET EXPECTED + 2
2728 035102 012777 003260 145142      MOV     #BUF,@RLBA     ;SET BUS ADDRESS
2729 035110 012777 177577 145140      MOV     #-129.,@RLMP    ;WORD COUNT-SECTOR+1 WORD
2730
2731 035116 004537 020674          JSR     R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
2732 035122 000002          WRCHK
2733 035124 004537 021514          JSR     R5,WTCRDY     ;WRITE CHECK
2734 035130          ESCAPE  SEG          ;WAIT FOR CONTROLLER READY?
      035130 104010          EMT     C$ESCAPE     ;CHECK FOR FL:LOE, ELSE EXIT SEG
      035132 000042          .WORD  10001$-.

2735
2736 035134 004537 020432          JSR     R5,CHERR      ;CHECK CNTLR FOR ERRORS
2737 035140          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      035140 104010          EMT     C$ESCAPE
  
```

```

035142 000032 .WORD 10001$-.
2738
2739 035144 013737 002240 002176 MOV E,DA,BDDAT ;READ DISK ADDRESS
2740 035152 023737 002176 002174 CMP BDDAT,GDDAT ;IS DISK ADDRESS CORRECT
2741 035160 001404 BEQ 2$ ;YES, BRANCH NO, REPORT ERROR
2742
2743 035162 ERRDF 7.,EM63,ERR4 ;DISK ADDRESS NOT CORRECT
035162 104462 TRAP T$ERCODE
035164 000007 .WORD 7
035166 014135 .WORD EM63
035170 014616 .WORD ERR4
2744
2745 035172 2$: CKLOOP
035172 104006 EMT C$CLP1
2746
2747 035174 ENDSEG ;%%END OF SEGMENT%%
035174 10001$:
035174 104005 EMT C$ESEG
2748
2749 035176 005237 002166 INC TMO ;NEXT SECTOR
2750 035202 022737 000046 002166 CMP #46,TMO ;AT END?
2751 035210 001233 BNE 1$ ;NO, GO BACK
2752 035212 ENDSEG ;%%END OF SEGMENT%%
035212 10000$:
035212 104005 EMT C$ESEG
2753 035214 ENDTST ;**END OF TEST**
035214 L10072:
035214 104001 EMT C$ETST
2754 .SBTTL **TEST 38** - FORCE DCK WITH WRITE CHECK
2755
2756 035216 BGNTST ;**START OF TEST**
2757
2758 035216 STARS
:*****
2759 :FORCE A DCK WITH WRITE CHECK. THIS IS DONE BY WRITING
2760 :A SECTOR AND CHANGING A WORD IN MEMORY BEFORE WRITE CHECK
2761 :IS ISSUED..
2762 035216 STARS
:*****
2763
2764 035216 004737 021574 JSR PC,HDHOME ;HEADS OVER TRACK 0
2765 035222 CKERFG ;HEADS GO HOME OKAY
035230 104032 EMT C$EXIT
035232 000262 .WORD L10073-.
2766
2767 035234 BGNSEG ;%%START OF SEGMENT%%
035234 104004 EMT C$BSEG
2768
2769 035236 012700 003260 MOV #BUF,R0 ;SETUP AND WRITE
2770 035242 012701 000200 MOV #128,R1 ;128 WORDS
2771 035246 012720 125252 299$: MOV #125252,(R0)+ ;WRITE
2772 035252 005301 DEC R1 ;DONE??
2773 035254 001374 BNE 299$
2774
2775 035256 012777 003260 144766 MOV #BUF,@RLBA ;LOAD BUS ADDRESS
2776 035264 012777 177600 144764 MOV #-128,@RLMP ;WORD COUNT
2777 035272 005077 144756 CLR @RLDA ;CLEAR DISK ADDRESS
  
```

```

2778 035276 004537 020674      JSR    R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
2779 035302 000012              WRITE
2780 035304 004537 021514      JSR    R5,WTCRDY        ;WAIT FOR CONTROLLER READY
2781 035310              ESCAPE  SEG             ;CHECK FOR FL:LOE, ELSE EXIT SEG
      035310 104010          EMT    C$ESCAPE
      035312 000200          .WORD  10000$-.
2782 035314 004537 020432      JSR    R5,CHERR         ;CHECK CNTLR FOR ERRORS
2783 035320              ESCAPE  SEG             ;CHECK FOR FL:LOE, ELSE EXIT SEG
      035320 104010          EMT    C$ESCAPE
      035322 000170          .WORD  10000$-.
2784              ;VERIFY WRITE WITH READ BEFORE WRCHK
2785
2786 035324 005077 144724      CLR    @RLDA
2787 035330 012777 003260 144714  MOV    #BUF,@RLBA
2788 035336 012777 177600 144712  MOV    #-128.,@RLMP
2789 035344 004537 020674      JSR    R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
2790 035350 000014              READ
2791 035352 004537 021514      JSR    R5,WTCRDY
2792 035356              ESCAPE  SEG             ;CHECK FOR FL:LOE, ELSE EXIT SEG
      035356 104010          EMT    C$ESCAPE
      035360 000132          .WORD  10000$-.
2793 035362 004537 020432      JSR    R5,CHERR         ;CHECK CNTLR FOR ERRORS
2794 035366              ESCAPE  SEG             ;CHECK FOR FL:LOE, ELSE EXIT SEG
      035366 104010          EMT    C$ESCAPE
      035370 000122          .WORD  10000$-.
2795
2796 035372              BGNSEG
      035372 104004          EMT    C$BSEG          ;%%START OF SEGMENT%%
2797
2798
2799 035374 005037 003260      CLR    BUF
2800 035400 005077 144650      CLR    @RLDA
2801 035404 012777 003260 144640  MOV    #BUF,@RLBA        ;SETTING SECTOR 40 OF CYL. ADDR.
2802 035412 012777 177600 144636  MOV    #-128.,@RLMP      ;WORD COUNT
2803
2804 035420 004537 020674      JSR    R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
2805 035424 000002              WRCHK          ;WRITE CHECK
2806 035426 004537 021514      JSR    R5,WTCRDY        ;WAIT FOR CONTROLLER READY
2807 035432              ESCAPE  SEG             ;CHECK FOR FL:LOE, ELSE EXIT SEG
      035432 104010          EMT    C$ESCAPE
      035434 000054          .WORD  10001$-.
2808
2809 035436 013737 002234 002166  MOV    E.CS,TMPO        ;GET RLCS
2810 035444 042737 001777 002166  BIC    #1777,TMPO       ;SAVE ERROR BITS
2811 035452 022737 104000 002166  CMP    #BIT15:BIT11,TMPO ;DCK SET.
2812 035460 001402              BEQ    1$           ;YES, CONTINUE
2813 035462 004537 020432      JSR    R5,CHERR
2814 035466              1$: CKLOOP
      035466 104006          EMT    C$CLP1
2815
2816 035470 022737 104000 002166  CMP    #BIT15:BIT11,TMPO
2817 035476 001404              BEQ    2$
2818
2819 035500              ERRDF  23.,EM65,ERRO
      035500 104462          TRAP  T$ERCODE
      035502 000027          .WORD  23
      035504 014313          .WORD  EM65

```

```

2820 035506 014452          .WORD  ERRO          ;WHEN FORCED
2821 035510                2$:
2822
2823 035510                ENDSEG          ;%%END OF SEGMENT%%
      035510                10001$:
2824 035510 104005          EMT      C$ESEG          ;%%END OF SEGMENT%%
      035512                ENDSEG
      035512                10000$:
2825 035512 104005          EMT      C$ESEG          ;**END OF TEST**
      035514                ENDTST
      035514                L10073:
      035514 104001          EMT      C$ETST

2826
2827          .SBTTL  **TEST 39** - FORCE DCK WITH WRITE CHECK INTERRUPT
2828
2829 035516                BGNST          ;**START OF TEST**
2830
2831
2832 035516                STARS
      ;*****
2833          ;FORCE A DCK IN INTERRUPT MODE
2834 035516                STARS
      ;*****

2835
2836
2837 035516 004737 021574    JSR      PC,HDHOME        ;HEADS OVER TRACK 0
2838 035522                CKERFG          ;HEADS GO HOME OKAY
      035530 104032          EMT      C$EXIT
      035532 000322          .WORD  L10074-.

2839
2840 035534                BGNSEG          ;%%START OF SEGMENT%%
      035534 104004          EMT      C$BSEG

2841
2842 035536 012700 003260    MOV      #BUF,R0          ;SETUP AND WRITE
2843 035542 012701 000200    MOV      #128.,R1         ;128 WORDS
2844 035546 012720 125252    299$:  MOV      #125252,(R0)+ ;WRITE
2845 035552 005301          DEC      R1               ;DONE??
2846 035554 001374          BNE     299$

2847
2848 035556 012777 003260 144466  MOV      #BUF,@RLBA       ;LOAD BUS ADDRESS
2849 035564 012777 177600 144464  MOV      #-128.,@RLMP     ;WORD COUNT
2850 035572 005077 144456          CLR      @RLDA           ;CLEAR DISK ADDRESS
2851 035576 004537 020674          JSR      R5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
2852 035602 000012          WRITE
2853 035604 004537 021514          JSR      R5,WTCRDY        ;WAIT FOR CONTROLLER READY
2854 035610                ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      035610 104010          EMT      C$ESCAPE
      035612 000240          .WORD  10000$-.

2855 035614 004537 020432          JSR      R5,CHERR         ;CHECK CNTLR FOR ERRORS
2856 035620                ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      035620 104010          EMT      C$ESCAPE
      035622 000230          .WORD  10000$-.

2857          ;VERIFY WRITE WITH READ BEFORE WRCHK
2858
2859 035624 005077 144424          CLR      @RLDA
2860 035630 012777 003260 144414  MOV      #BUF,@RLBA
  
```



```

2861 035636 012777 177600 144412      MOV    #-128.,@RLMP
2862 035644 004537 020674              JSR    R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
2863 035650 000014                      READ
2864 035652 004537 021514              JSR    R5,WTCRDY
2865 035656                      ESCAPE SEG                ;CHECK FOR FL:LOE, ELSE EXIT SEG
    035656 104010                      EMT    C$ESCAPE
    035660 000172                      .WORD 10000$-
2866 035662 004537 020432              JSR    R5,CHERR          ;CHECK CNTLR FOR ERRORS
2867 035666                      ESCAPE SEG                ;CHECK FOR FL:LOE, ELSE EXIT SEG
    035666 104010                      EMT    C$ESCAPE
    035670 000162                      .WORD 10000$-
2868
2869 035672                      BGNSEG                    ;%%START OF SEGMENT%%
    035672 104004                      EMT    C$BSEG
2870
2871 035674                      SETPRI #PRI00
    035674 012700 000000              MOV    #PRI00,R0
    035700 104041                      EMT    C$SPRI
2872 035702 005037 002152              CLR    INTFLG            ;CLEAR INTERRUPT OCCURANCE FLAG
2873 035706 005037 003260              CLR    BUF
2874 035712 005077 144336              CLR    @RLDA
2875 035716 012777 003260 144326      MOV    #BUF,@RLBA        ;SETTING SECTOR 40 OF CYL. ADDR.
2876 035724 012777 177600 144324      MOV    #-128.,@RLMP      ;WORD COUNT
2877
2878 035732 004537 020674              JSR    R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
2879 035736 000102                      WRCHK!INTEN              ;WRITE CHECK
2880 035740 004537 021514              JSR    R5,WTCRDY          ;WAIT FOR CONTROLLER READY
2881 035744                      CKLOOP
    035744 104006                      EMT    C$CLP1
2882 035746                      SETPRI #PRI07
    035746 012700 000340              MOV    #PRI07,R0
    035752 104041                      EMT    C$SPRI
2883
2884 035754 005737 002152              TST    INTFLG            ;DID INTERRUPT OCCUR
2885 035760 001004                      BNE    2$                ;YES OKAY
2886
2887 035762                      ERRDF 24.,EM66,ERRO      ;NO INTERRUPT FROM DCK
    035762 104462                      TRAP  T$ERCODE
    035764 000030                      .WORD 24
    035766 014354                      .WORD EM66
    035770 014452                      .WORD ERRO
2888
2889 035772                      2$: ESCAPE SEG                ;CHECK FOR FL:LOE, ELSE EXIT SEG
    035772 104010                      EMT    C$ESCAPE
    035774 000054                      .WORD 10001$-
2890
2891
2892 035776 013737 002234 002166      MOV    E.CS,TMPO          ;GET RLCS
2893 036004 042737 001777 002166      BIC    #1777,TMPO         ;SAVE ERROR BITS
2894 036012 022737 104000 002166      CMP    #BIT15:BIT11,TMPO ;DCK SET.
2895 036020 001402                      BEQ    1$                ;YES, CONTINUE
2896
2897 036022 004537 020432              JSR    R5,CHERR
2898 036026                      1$: CKLOOP
    036026 104006                      EMT    C$CLP1
2899
2900 036030 022737 104000 002166      CMP    #BIT15:BIT11,TMPO

```

```

2901 036036 001404          BEQ      3$
2902 036040          ERRDF   25.,EM65,ERRO
      036040 104462          TRAP   T$ERCODE
      036042 000031          .WORD  25
      036044 014313          .WORD  EM65
      036046 014452          .WORD  ERRO
2903
2904 036050          3$:
2905
2906 036050          ENDSEG           ;%%END OF SEGMENT%%
      036050          10001$:
      036050 104005          EMT      C$ESEG           ;%%END OF SEGMENT%%
2907 036052          ENDSEG           ;%%END OF SEGMENT%%
      036052          10000$:
      036052 104005          EMT      C$ESEG
2908 036054          ENDTST           ;**END OF TEST**
      036054          L10074:
      036054 104001          EMT      C$ETST
2909
2910
2911          .SBTTL  **TEST 40** - CHECK ZERO FILL ON WRITE WITH WRITE CHECK
2912
2913 036056          BGNST           ;**START OF TEST**
2914
2915
2916
2917 036056          STARS
      ;*****
      ;WHEN WRITING PARTIAL SECTORS (LESS THAN 128 WORDS) THE
      ;CONTROLLER WILL FILL IN THE REMAINING PORTION OF
      ;THE SECTOR WITH ZERO WORDS. CHECK THIS FEATURE CAN BE WRITE CHECKED
      ;WITH WORD COUNTS FROM 1 TO 127
      STARS
      ;*****
2918
2919
2920
2921
2922 036056
2923
2924 036056 004737 021574          JSR      PC,HDHOME           ;HEADS OVER TRACK 0
2925 036062          CKERFG           ;HEADS GO HOME OKAY
      036070 104032          EMT      C$EXIT
      036072 000274          .WORD  L10075-.
2926
2927 036074          BGNSEG           ;%%START OF SEGMENT%%
      036074 104004          EMT      C$BSEG
2928
2929 036076 012737 000001 002170          MOV     #1,TMP1           ;START WITH 1 WORD WRITE
2930 036104 012700 003260          33$: MOV     #BUF,R0           ;WRITE BUFFER WITH 52525, WE'LL
2931 036110 012701 000200          MOV     #128.,R1         ;WRITE 128 WORDS ALL THOUGH WE'RE
2932 036114 012720 052525          3$:  MOV     #52525,(R0)+ ;ONLY GOING TO TRANSFER < 128
2933 036120 005301          DEC     R1               ;DONE WITH BUFFER?
2934 036122 001374          BNE     3$               ;NO, GO BACK
2935 036124 013700 002170          MOV     TMP1,R0          ;GET TRANSFER WORD COUNT
2936 036130 005400          NEG     R0               ;NEGATE FOR RLMP
2937 036132 010077 144120          MOV     R0,@RLMP        ;STORE WORD COUNT AWAY
2938 036136 012777 003260 144106          MOV     #BUF,@RLBA      ;SET UP RLBA
2939 036144 005077 144104          CLR     @RLDA
2940 036150 004537 020674          JSR     R5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
2941 036154 000012          WRITE  IT               ;WRITE IT
2942 036156 004537 021514          JSR     R5,WTCRDY       ;WAIT FOR WRITE TO FINISH
  
```

```

2943 036162          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      036162 104010  EMT C$ESCAPE
      036164 000200  .WORD 10000$-.
2944
2945 036166 004537 020432 JSR R5,CHERR          ;CHECK CNTLR FOR ERRORS
2946 036172          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      036172 104010  EMT C$ESCAPE
      036174 000170  .WORD 10000$-.
2947          ;VERIFY WRITE WITH READ BEFORE WRCHK
2948
2949 036176 005077 144052 CLR @RLDA
2950 036202 012777 003260 144042 MOV #BUF,@RLBA
2951 036210 013700 002170 MOV TMP1,R0
2952 036214 005400 NEG R0
2953 036216 010077 144034 MOV R0,@RLMP
2954 036222 004537 020674 JSR R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
2955 036226 000014 READ
2956 036230 004537 021514 JSR R5,WTCRDY
2957 036234          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      036234 104010  EMT C$ESCAPE
      036236 000126  .WORD 10000$-.
2958 036240 004537 020432 JSR R5,CHERR          ;CHECK CNTLR FOR ERRORS
2959 036244          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      036244 104010  EMT C$ESCAPE
      036246 000116  .WORD 10000$-.
2960
2961 036250          BGNSEG          ;%%START OF SEGMENT%%
      036250 104004  EMT C$BSEG
2962 036252 012777 003260 143772 MOV #BUF,@RLBA          ;SET UP TO READ
2963 036260 013700 002170 MOV TMP1,R0
2964 036264 005400 NEG R0
2965 036266 010077 143764 MOV R0,@RLMP
2966 036272 005077 143756 CLR @RLDA          ;SECTOR
2967 036276 004537 020674 JSR R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
2968 036302 000002 WRCHK
2969 036304 004537 021514 JSR R5,WTCRDY          ;WAIT TIL WE FINISH THE WRCHK
2970 036310          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      036310 104010  EMT C$ESCAPE
      036312 000034  .WORD 10001$-.
2971
2972 036314 004537 020432 JSR R5,CHERR          ;CHECK CNTLR FOR ERRORS
2973 036320 005737 002132 TST T.CRC          ;WAS ERROR A DCK??
2974 036324 001003 BNE 8$          ;YES, GIVE MOR INFO
2975 036326          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      036326 104010  EMT C$ESCAPE
      036330 000016  .WORD 10001$-.
2976 036332 000405 BR 99$          ;SKIP AROUND
2977 036334          CKLOOP          ;YES, CHECK FOR LOOP FIRST
      036334 104006  EMT C$CLP1
2978 036336          ERRDF 37.,EM64,ERR14
      036336 104462  TRAP T$ERCODE
      036340 000045  .WORD 37
      036342 014237  .WORD EM64
      036344 015356  .WORD ERR14
2979 036346          99$:          ;EXIT TEST
2980 036346          ENDSEG          ;%%END OF SEGMENT%%
      036346          10001$:

```

```

2981 036346 104005          EMT      C$ESEG
2982 036350 005237 002170    INC      TMP1
2983 036354 023727 002170 000200  CMP      TMP1,#128.
2984 036362 001250          BNE      33$
2985
2986 036364          ENDSEG          ;%%END OF SEGMENT%%
      036364          10000$:
      036364 104005    EMT      C$ESEG
2987 036366          ENDTST          ;**END OF TEST**
      036366          L10075:
      036366 104001    EMT      C$ETST
2988          .SBTTL  **TEST 41** - EXTENDED CHECK OF WRITE CHECK FUNCTION
2989
2990 036370          BGN:TST          ;**START OF TEST**
2991
2992 036370          STARS
      ;:*****
2993          ;CHECK OF WRITE CHECK LOGIC UNDER FLAG MODE
2994          ;THIS TEST IS DONE WITH ALL BIT PATTERNS
2995          ; WE WILL WRITE CHECK A FULL SECTOR (128 WORDS) FROM
2996          ;MEMORY (BUF). WE CHECK THAT NO ERRORS OCCUR.
2997 036370          STARS
      ;:*****
2998
2999
3000 036370 004737 021574    JSR      PC,HDHOME          ;HEADS OVER TRACK 0
3001 036374          CKERFG          ;HEADS GO HOME OKAY
      036402 104032    EMT      C$EXIT
      036404 000306    .WORD    L10076-.
3002
3003 036406 022737 000001 002126  CMP      #1,T.DRIVE          ;COMPARE TYPE OF DRIVE (RL01,RL02)
3004 036414 001003          BNE      22$                ;RL02, THEN BRANCH
3005 036416 012703 002522    MOV      #HDRTAB,R3          ;MOV HDRTAB TO R3
3006 036422 000402          BR       33$                ; THEN BRANCH
3007 036424 012703 002702    22$:    MOV      #HTAB,R3          ;MOV #HTAB TO R3
3008
3009 036430          33$:    BGNSEG          ;START OF SEGMENT
      036430 104004    EMT      C$BSEG
3010
3011 036432 012700 003260    298$:   MOV      #BUF,R0            ;SETUP AND WRITE
3012 036436 012701 000200    MOV      #128.,R1          ;128 WORDS
3013 036442 011302          MOV      (R3),R2
3014 036444 010220    299$:   MOV      R2,(R0)+          ;WRITE
3015 036446 005301          DEC      R1                ;DONE??
3016 036450 001375          BNE      299$
3017
3018 036452 012777 003260 143572  MOV      #BUF,@RLBA          ;LOAD BUS ADDRESS
3019 036460 012777 177600 143570  MOV      #-128.,@RLMP        ;WORD COUNT
3020 036466 005077 143562    CLR      @RLDA              ;CLEAR DISK ADDRESS
3021 036472 004537 020674    JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
3022 036476 000012          WRITE
3023 036500 004537 021514    JSR      R5,WTCRDY          ;WAIT FOR CONTROLLER READY
3024 036504          ESCAPE  SEG                ;CHECK FOR FL:LOE, ELSE EXIT SEG
      036504 104010    EMT      C$ESCAPE
      036506 000202    .WORD    10000$-.
3025 036510 004537 020432    JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
  
```

```

3026 036514          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      036514 104010  EMT      C$ESCAPE
      036516 000172  .WORD   10000$-.
3027 036520          BGNSEG          ;%%START OF SEGMENT%%
      036520 104004  EMT      C$BSEG
3028
3029          ;VERIFY WRITE WITH READ BEFORE WRCHK
3030
3031 036522 005077 143526          CLR      @RLDA
3032 036526 012777 003260 143516  MOV      #BUF,@RLBA
3033 036534 012777 177600 143514  MOV      #-128.,@RLMP
3034 036542 004537 020674          JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
3035 036546 000014          READ
3036 036550 004537 021514          JSR      R5,WTCRDY
3037 036554          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      036554 104010  EMT      C$ESCAPE
      036556 000076  .WORD   10001$-.
3038 036560 004537 020432          JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
3039 036564          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      036564 104010  EMT      C$ESCAPE
      036566 000066  .WORD   10001$-.
3040
3041 036570          BGNSEG          ;%%START OF SEGMENT%%
      036570 104004  EMT      C$BSEG
3042
3043 036572          3$:
3044 036572 005077 143456          CLR      @RLDA
3045 036576 012777 177600 143452  MOV      #-128.,@RLMP          ;WORD COUNT
3046 036604 012777 003260 143440  MOV      #BUF,@RLBA          ;BUS ADDRESS
3047 036612 004537 020674          JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
3048 036616 000002          WRCHK          ;WRITE CHECK
3049
3050 036620 004537 021514          JSR      R5,WTCRDY          ;WAIT FOR CONTROLLER READY
3051 036624          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      036624 104010  EMT      C$ESCAPE
      036626 000024  .WORD   10002$-.
3052
3053
3054 036630 004537 020432          JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
3055 036634 005737 002132          TST     T.CRC          ;WRITE CHECK ERROR??
3056 036640 001404          BEQ     4$          ;NO
3057
3058 036642          ERRHRD 410.,ERR15,EM70
      036642 104463  TRAP   T$ERCODE
      036644 000632  .WORD   410
      036646 015424  .WORD   ERR15
      036650 014426  .WORD   EM70
3059
3060 036652          4$:
3061
3062 036652          ENDSEG          ;%%END OF SEGMENT%%
      036652 10002$:
      036652 104005  EMT      C$ESEG
3063 036654          ENDSEG          ;%%END OF SEGMENT%%
      036654 10001$:
      036654 104005  EMT      C$ESEG
3064
  
```

```

3065 036656 005723          TST      (R3)+
3066 036660 022737 000001 002126  CMP      #1,T.DRIVE      ;RL01 OR RL02?
3067 036666 001003          BNE      65$             ;RL02 THEN BRANCH
3068 036670 020327 002700  CMP      R3,#HDREND      ;MOV #HDREND TO R3
3069 036674 000402          BR       67$             ;THEN BRANCH
3070
3071 036676 020327 003066 65$:    CMP      R3,#HEND      ;COM FOR LAST PATTERN
3072 036702 001402 67$:    BEQ      57$
3073 036704 000137 036432  JMP      298$
3074
3075
3076 036710          57$:    ENDSEG                ;END OF SEGMENT
      036710          10000$:
3077 036712          ENDTST
      036712          L10076:
      036712 104001  EMT      C$ESEG                ;**END OF TEST**
      EMT      C$ETST
3078
3079
3080          .SBTTL  **TEST 42** - EXTENDED CHECK OF WRITE CHECK FUNCTION
3081
3082 036714          BGNST                ;**START OF TEST**
3083
3084 036714          STARS
      :*****
3085          :CHECK OF WRITE CHECK LOGIC UNDER FLAG MODE
3086          :TEST IS DONE WITH ALL BIT PATTERNS
3087          : WE WILL WRITE CHECK A FULL SECTOR (128 WORDS) FROM
3088          :MEMORY (BUF). WE CHECK THAT NO ERRORS OCCUR.
3089 036714          STARS
      :*****
3090
3091
3092 036714 004737 021574  JSR      PC,HDHOME      ;HEADS OVER TRACK 0
3093 036720          CKERFG      ;HEADS GO HOME OKAY
      036726 104032  EMT      C$EXIT
      036730 000306  .WORD    L10077-.
3094
3095 036732 022737 000001 002126  CMP      #1,T.DRIVE      ;CHECK TYPE OF DRIVE
3096 036740 001003          BNE      22$             ;NOT RL01 THEN BRANCH
3097 036742 012703 002522  MOV      #HDRTAB,R3      ;MOV #HDRTAB TO R3
3098 036746 000402          BR       33$             ;THEN BRANCH
3099 036750 012703 002702 22$:    MOV      #HTAB,R3      ;MOV #HTAB TO R3 (RL02)
3100
3101 036754          33$:    BGNSEG                ;START OF SEGMENT
      036754 104004  EMT      C$BSEG
3102
3103 036756 012700 003260 298$:  MOV      #BUF,R0         ;SETUP AND WRITE
3104 036762 012701 000200  MOV      #128.,R1        ;128 WORDS
3105 036766 011302          MOV      (R3),R2        ;GET PATTERN
3106 036770 052702 100000  BIS      #BIT15,R2
3107 036774 010220 299$:  MOV      R2,(R0)+
3108 036776 005301          DEC      R1             ;DONE??
3109 037000 001375          BNE      299$
3110
3111 037002 012777 003260 143242  MOV      #BUF,@RLBA      ;LOAD BUS ADDRESS
3112 037010 012777 177600 143240  MOV      #-128.,@RLMP    ;WORD COUNT
  
```

```

3113 037016 005077 143232          CLR    @RLDA          ;CLEAR DISK ADDRESS
3114 037022 004537 020674          JSR    R5,LDFUNC     ;LOAD THE FUNCTION IN NEXT WORD
3115 037026 000012                   WRITE
3116 037030 004537 021514          JSR    R5,WTCRDY     ;WAIT FOR CONTROLLER READY
3117 037034                   ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      037034 104010                   EMT    C$ESCAPE
      037036 000176                   .WORD 10000$-
3118 037040 004537 020432          JSR    R5,CHERR      ;CHECK CNTLR FOR ERRORS
3119 037044                   ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      037044 104010                   EMT    C$ESCAPE
      037046 000166                   .WORD 10000$-
3120 037050                   BGNSEG
      037050 104004                   EMT    C$BSEG          ;%%START OF SEGMENT%%

3121
3122                               ;VERIFY WRITE WITH READ BEFORE WRCHK
3123
3124 037052 005077 143176          CLR    @RLDA
3125 037056 012777 003260 143166  MOV    #BUF,@RLBA
3126 037064 012777 177600 143164  MOV    #-128.,@RLMP
3127 037072 004537 020674          JSR    R5,LDFUNC     ;LOAD THE FUNCTION IN NEXT WORD
3128 037076 000014                   READ
3129 037100 004537 021514          JSR    R5,WTCRDY     ;CHECK FOR FL:LOE, ELSE EXIT SEG
3130 037104                   ESCAPE  SEG
      037104 104010                   EMT    C$ESCAPE
      037106 000076                   .WORD 10001$-
3131 037110 004537 020432          JSR    R5,CHERR      ;CHECK CNTLR FOR ERRORS
3132 037114                   ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      037114 104010                   EMT    C$ESCAPE
      037116 000066                   .WORD 10001$-
3133
3134 037120                   BGNSEG
      037120 104004                   EMT    C$BSEG          ;%%START OF SEGMENT%%

3135
3136 037122                               3$:
3137 037122 005077 143126          CLR    @RLDA
3138 037126 012777 177600 143122  MOV    #-128.,@RLMP ;WORD COUNT
3139 037134 012777 003260 143110  MOV    #BUF,@RLBA  ;BUS ADDRESS
3140 037142 004537 020674          JSR    R5,LDFUNC     ;LOAD THE FUNCTION IN NEXT WORD
3141 037146 000002                   WRCHK                ;WRITE CHECK
3142
3143 037150 004537 021514          JSR    R5,WTCRDY     ;WAIT FOR CONTROLLER READY
3144 037154                   ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      037154 104010                   EMT    C$ESCAPE
      037156 000024                   .WORD 10002$-
3145
3146
3147 037160 004537 020432          JSR    R5,CHERR      ;CHECK CNTLR FOR ERRORS
3148 037164 005737 002132          TST   T.CRC
3149 037170 001404                   BEQ   4$
3150
3151 037172                   ERRHRD 410.,ERR15,EM70
      037172 104463                   TRAP  T$ERCODE
      037174 000632                   .WORD 410
      037176 015424                   .WORD ERR15
      037200 014426                   .WORD EM70
3152
3153 037202                               4$:

```

```

3154
3155
3156 037202          ENDSEG          ;%%END OF SEGMENT%%
      037202          10002$:
3157 037202 104005  EMT C$ESEG
      037204          10001$:          ;%%END OF SEGMENT%%
      037204 104005  EMT C$ESEG
3158
3159 037206 005723  TST (R3)+
3160 037210 022737 000001 002126  CMP #1,T.DRIVE ;RL01 OR RL02?
3161 037216 001003  BNE 60$ ;RL02? THEN BRANCH
3162 037220 020327 002700  CMP R3,#HDREND ;LAST OF PATERN?
3163 037224 000402  BR 77$
3164 037226 020327 003066 60$: CMP R3,#HEND ;LAST OF PATTERN (RL02)
3165 037232 001251 77$: BNE 298$
3166
3167 037234          ENDSEG          ;%%END OF SEGMENT%%
      037234          10000$:
3168 037236          EMT C$ESEG
      037236          L10077:          ;**END OF TEST**
      037236 104001  EMT C$ETST
3169 .SBTTL **TEST 43** - READ WITHOUT HEADER COMPARE FUNCTION
3170
3171 037240
      STARS
      ;:*****
      ;TEST THAT READ WITHOUT HEADER VERIFICATION WORKS. THIS FUNCTION SHOULD
      ;READ AT THE NEXT SECTOR ENCOUNTERED. SET THE RLDA TO 0
      ;AND ISSUE THE FUNCTION IN FLAG MODE. UPON COMPLETION CHECK
      ;FOR ERRORS
      STARS
      ;:*****
3172
3173
3174
3175
3176 037240
      BGNST          ;**START OF TEST**
3177 037240
3178
3179
3180 037240 004737 021574 JSR PC,HDHOME ;HEADS OVER TRACK 0
3181 037244          CKERFG          ;HEADS GO HOME OKAY
      037252 104032  EMT C$EXIT
      037254 000052 .WORD L10100-.
3182
3183 037256          BGNSEG          ;%%START OF SEGMENT%%
      037256 104004  EMT C$BSEG
3184
3185
3186 037260 012777 177600 142770 MOV #-128,@RLMP ;SET UP WORD COUNT
3187 037266 012777 003260 142756 MOV #BUF,@RLBA ;SETUP BUS ADDRESS
3188 037274 012777 177777 142752 MOV #-1,@RLDA ;HEADER SHOULDN'T MATTER
3189 037302 004537 020674 JSR R5,LDFUNC ;LOAD THE FUNCTION IN NEXT WORD
3190 037306 000016 RDNHD ;READ DATA WITHOUT HEADER VERIFY
3191 037310 004537 021514 JSR R5,WTCRDY ;WAIT FOR IT TO FINISH
3192 037314          ESCAPE SEG ;CHECK FOR FL:LOE, ELSE EXIT SEG
      037314 104010  EMT C$ESCAPE
      037316 000006 .WORD 10000$-.
3193
3194 037320 004537 020432 JSR R5,CHERR ;CHECK CNTLR FOR ERRORS
3195
  
```



```

3196 037324          ENDSEG          ;%%END OF SEGMENT%%
      037324          10000$:
3197 037324 104005  EMT      C$ESEG
      037326          ENDTST          ;**END OF TEST**
      037326          L10100:
      037326 104001  EMT      C$ETST

3198
3199          .SBTTL  **TEST 44** - READ WITHOUT HEADER COMPARE FUNCTION INTERRUPT
3200
3201 037330          BGNTST          ;**START OF TEST**
3202
3203 037330          STARS
      :*****
3204          :TEST THAT READ WITHOUT HEADER VERIFICATION WORKS IN
3205          :INTERRUPT MODE.
3206 037330          STARS
      :*****

3207
3208 037330 004737 021574 JSR      R5,HDHOME      ;HEADS OVER TRACK 0
3209 037334          CKERFG          ;HEADS GO HOME OKAY
      037342 104032  EMT      C$EXIT
      037344 000114  .WORD    L10101-.

3210
3211 037346          BGNSEG          ;%%START OF SEGMENT%%
      037346 104004  EMT      C$BSEG

3212
3213 037350 005037 002152 CLR      INTFLG ;CLEAR INTERRUPT OCCURANCE FLAG
3214 037354 012777 177600 142674 MOV      #-128.,@RLMP ;SET UP WORD COUNT FOR ONE SECTOR
3215 037362 012777 003260 142662 MOV      #BUF,@RLBA ;SETUP BUFFER ADDRESS
3216 037370 012777 177777 142656 MOV      #-1,@RLDA ;DISK ADDRESS IS A DON'T CARE
3217 037376          SETPRI #PRI00
      037376 012700 000000 MOV      #PRI00,R0
      037402 104041  EMT      C$SPRI

3218 037404 004537 020674 JSR      R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
3219 037410 000116          RDNHD!INTEN ;INTERRUPT ENABLED
3220 037412 004537 021514 JSR      R5,WTCRDY ;WAIT FOR INTERRUPT
3221 037416          SETPRI #PRI07
      037416 012700 000340 MOV      #PRI07,R0
      037422 104041  EMT      C$SPRI

3222 037424          ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      037424 104010  EMT      C$ESCAPE
      037426 000030  .WORD    10000$-.

3223
3224 037430 005737 002152 TST      INTFLG          ;DID IT INTERRUPT
3225 037434 001004          BNE      1$              ;IF INTERRUPT GO TO 1$
3226
3227 037436          ERRDF 40.,EM40,ERRO ;NO INTERRUPT
      037436 104462  TRAP    T$ERCODE
      037440 000050  .WORD    40
      037442 012755  .WORD    EM40
      037444 014452  .WORD    ERRO

3228 037446          1$: ESCAPE SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      037446 104010  EMT      C$ESCAPE
      037450 000006  .WORD    10000$-.

3229
3230 037452 004537 020432 JSR      R5,CHERR          ;CHECK CNTLR FOR ERRORS
3231
  
```

TEST 44 - READ WITHOUT HEADER COMPARE FUNCTION INTERRUPT

SEQ 0129

```

3232 037456          ENDSEG          ;%%END OF SEGMENT%%
      037456          10000$:
3233 037456 104005   EMT          C$ESEG
      037460          ENDTST          ;**END OF TEST**
      037460          L10101:
      037460 104001   EMT          C$ETST

3234
3235          .SBTTL  **TEST 45** - CHECK RD W/O HDR CMP ACTUALLY READS
3236
3237 037462          BGNSTST          ;**START OF TEST**
3238
3239 037462          STARS
      ;*****
3240          ;CHECK THAT THE READ W/O HDR CMP FUNCTION ACTUALLY READS (INTO MEMORY)
3241          ;WE WILL WRITE A PATTERN INTO MEMORY AND THEN ISSUE
3242          ;A READ TO OVERLAY THAT PATTERN. AFTER THE READ
3243          ;WE CHECK TO SEE IF THE WRITTEN PATTERN HAS CHANGED.
3244          ;IF NOT WE ISSUE IT AGAIN AT THE SAME SECTION AFTER
3245          ;HAVING MODIFIED OUR PATTERN IN MEMORY (SINCE THERE IS
3246          ;ONE CHANCE THAT THE DISK COULD HAVE OUR PATTERN). AFTER
3247          ;THE SECOND READ WE CHECK THE BUFFER AGAIN. IF IT'S
3248          ;NOT CHANGED WE REPORT AN ERROR
3249 037462          STARS
      ;*****

3250
3251
3252 037462 004737 021574 JSR          PC,HDHOME          ;HEADS OVER TRACK 0
3253 037466          CKERFG          ;HEADS GO HOME OKAY
      037474 104032   EMT          C$EXIT
      037476 000160   .WORD        L10102-.

3254
3255 037500          BGNSEGE          ;%%START OF SEGMENT%%
      037500 104004   EMT          C$BSEG

3256
3257 037502 012737 024350 002166 MOV          #24350,TMPO          ;SET PATTERN TO WRITE
3258 037510 005037 002170 CLR          TMP1              ;CLEAR PASS INDICATOR
3259 037514 012700 003260 1$: MOV          #BUF,R0              ;SET UP BUFFER BEGINNING
3260 037520 012701 000200 MOV          #128.,R1
3261 037524 013720 002166 2$: MOV          TMPO,(R0)+          ;WRITE BUFFER
3262 037530 005301 DEC          R1              ;DONE??
3263 037532 001374 BNE          2$              ;NO, GO BACK
3264 037534 012777 000050 142512 MOV          #40.,@RLDA          ;LOAD DISK ADDRESS TO NONSENSE
3265 037542 012777 177600 142506 MOV          #-128.,@RLMP          ;SET WORD COUNT
3266 037550 012777 003260 142474 MOV          #BUF,@RLBA          ;LOAD BUS ADDRESS
3267 037556 012737 003260 002174 MOV          #BUF,GDDAT          ;FOR ERROR PRINT
3268
3269 037564 004537 020674 JSR          R5,LDFUNC          ;LOAD THE FUNCTION IN NEXT WORD
3270 037570 000016 RDNHD          ;READ W/O HDR CMP
3271 037572 004537 021514 JSR          R5,WTCRDY          ;WAIT FOR CONTROLLER READY
3272 037576          ESCAPE          SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
      037576 104010   EMT          C$ESCAPE
      037600 000054   .WORD        10000$-.

```

```

1
2 037602 004537 020432      JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
3 037606      ESCAPE SEG      ;CHECK FOR FL:LOE, ELSE EXIT SEG
  037606 104010      EMT      C$ESCAPE
  037610 000044      .WORD    10000$-.
4
5 037612 012702 003260      MOV      #BUF,R2      ;SET TO START COMPARING DATA
6 037616 022237 002166      4$:     CMP      (R2)+,TMP0 ;DID DATA CHANGE?
7 037622 001014      BNE     6$           ;YES, CHECK FOR END
8
9
10
11 037624 005737 002170      TST     TMP1          ;DATA DIDN'T CHANGE, CHECK
12 037630 001005      BNE     5$           ;IF 1ST OR 2ND TIME?
13
14 037632 005237 002170      INC     TMP1          ;INC PASS COUNT
15 037636 005137 002166      COM     TMP0          ;COMPLIMENT PATTERN
16 037642 000724      BR      1$           ;GO DO IT AGAIN
17
18 037644      5$:     ERRDF   20.,EM55,ERR9
  037644 104462      TRAP   T$ERCODE
  037646 000024      .WORD  20
  037650 013436      .WORD  EM55
  037652 015044      .WORD  ERR9
19
20 037654      6$:
21
22 037654      ENDSEG      ;%%END OF SEGMENT%%
  037654      10000$:
  037654 104005      EMT      C$ESEG
23 037656      ENDTST      ;**END OF TEST**
  037656      L10102:
  037656 104001      EMT      C$ETST
24
25      .SBTTL  **TEST 46** - CHECK RLBA INCREMENT WITH RD W/O HDR CMP
26
27 037660      BGNTST      ;**START OF TEST**
28
29 037660      STARS
  ;:*****
  ;CHECK THAT THE RLBA WILL INCREMENT WITH THE READ W/O HDR CMP
  ;THE RLBA SHOULD CONTAIN "BUF +256." AFTER A FULL SECTOR
  ;READ.
  ;STARS
  ;:*****
30
31
32
33 037660
34
35
36 037660 004737 021574      JSR      PC,HDHOME    ;HEADS OVER TRACK 0
37 037664      CKERFG      ;HEADS GO HOME OKAY
  037672 104032      EMT      C$EXIT
  037674 000120      .WORD    L10103-.
38
39 037676      BGNSEG      ;%%START OF SEGMENT%%
  037676 104004      EMT      C$BSEG
40
41 037700 012777 000050 142346      MOV      #40.,@RLDA
42 037706 012777 003260 142336      MOV      #BUF,@RLBA  ;SET UP BUS ADDRESS
  
```

```

43 037714 012777 177600 142334      MOV    #-128.,@RLMP      ;WORD COUNT
44 037722 012737 003260 002174      MOV    #BUF,GDDAT      ;FORM EXPECTED BUS ADDRESS
45 037730 062737 000400 002174      ADD    #256.,GDDAT     ;AFTER READ
46
47 037736 004537 020674                JSR    R5,LDFUNC        ;LOAD THE FUNCTION IN NEXT WORD
48 037742 000016                RDNHD                    ;READ W/O HDR CMP
49 037744 004537 021514                JSR    R5,WTCRDY        ;WAIT FOR CONTROLLER READY
50 037750                ESCAPE SEG              ;CHECK FOR FL:LOE, ELSE EXIT SEG
   037750 104010                EMT    C$ESCAPE
   037752 000040                .WORD 10000$-.
51
52 037754 004537 020432                JSR    R5,CHERR        ;CHECK CNTLR FOR ERRORS
53 037760                ESCAPE SEG              ;CHECK FOR FL:LOE, ELSE EXIT SEG
   037760 104010                EMT    C$ESCAPE
   037762 000030                .WORD 10000$-.
54 037764 013737 002236 002176      MOV    E.BA,BDDAT      ;READ 'RLBA' FOR PRESENT ADDRESS
55 037772 023737 002176 002174      CMP    BDDAT,GDDAT     ;DID 'BA' INCREMENT PROPERLY?
56 040000 001404                BEQ    1$              ;YES, CONTINUE
57
58 040002                ERRDF 21.,EM53,ERR4
   040002 104462                TRAP  T$ERCODE
   040004 000025                .WORD 21
   040006 013516                .WORD EM53
   040010 014616                .WORD ERR4
59
60 040012                1$:
61
62 040012                ENDSEG                  ;%%END OF SEGMENT%%
   040012                10000$:
   040012 104005                EMT    C$ESEG
63 040014                ENDTST                  ;**END OF TEST**
   040014                L10103:
   040014 104001                EMT    C$ETST
64
65
66
67
68
69
70
71                .SBTTL **TEST 47** - CHECK RLDA DOES INCREMENT WITH RD W/O HDR CMP
72
73 040016                BGNTST                  ;**START OF TEST**
74
75 040016                STARS
   ;:*****
76                ;CHECK THAT THE RLDA DOES INCREMENT BY ONE AFTER A
77                ;FULL SECTOR READ W/O HDR CMP
78                ;AFTER THE READ THE RLDA SHOULD STILL BE THE INITIAL RLDA + 1
79 040016                STARS
   ;:*****
80
81 040016 004737 021574                JSR    PC,HDHOME       ;HEADS OVER TRACK 0
82 040022                CKERFG                  ;HEADS GO HOME OKAY
   040030 104032                EMT    C$EXIT
   040032 000116                .WORD L10104-.
83

```

```

84 040034          BGNSEG          ;%%START OF SEGMENT%%
    040034 104004  EMT          C$BSEG
85
86
87 040036 012737 000050 002174  MOV      #40.,GDDAT      ;DA TO NONSENSE
88 040044 013777 002174 142202  MOV      GDDAT,@RLDA    ;SETUP DISK ADDRESS
89 040052 005237 002174          INC      GDDAT
90 040056 012777 177600 142172  MOV      #-128.,@RLMP   ;WORD COUNT
91 040064 012777 003260 142160  MOV      #BUF,@RLBA    ;SETUP BUS ADDRESS
92
93 040072 004537 020674          JSR      R5,LDFUNC      ;LOAD THE FUNCTION IN NEXT WORD
94 040076 000016          RDNHD                    ;READ WITHOUT HEADER COMPARE
95 040100 004537 021514          JSR      R5,WTCRDY     ;WAIT FOR CONTROLLER READY
96 040104          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
    040104 104010  EMT          C$ESCAPE
    040106 000040  .WORD      10000$-.
97
98 040110 004537 020432          JSR      R5,CHERR      ;CHECK CNTLR FOR ERRORS
99 040114          ESCAPE  SEG          ;CHECK FOR FL:LOE, ELSE EXIT SEG
    040114 104010  EMT          C$ESCAPE
    040116 000030  .WORD      10000$-.
100
101 040120 013737 002240 002176  MOV      E.DA,BDDAT     ;READ DISK ADDRESS
102 040126 023737 002174 002176  CMP      GDDAT,BDDAT   ;DID SECTOR INCREMENT PROPERLY
103 040134 001404          BEQ      1$           ;YES, BRANCH NO, REPORT ERROR
104
105 040136          ERRDF  22.,EM54,ERR4
    040136 104462  TRAP      T$ERCODE
    040140 000026  .WORD      22
    040142 013606  .WORD      EM54
    040144 014616  .WORD      ERR4
106
107 040146          1$:
108
109 040146          ENDSEG          ;%%END OF SEGMENT%%
    040146          10000$:
    040146 104005  EMT          C$ESEG
110 040150          ENDTST
    040150          L10104:
    040150 104001  EMT          C$ETST
111
112
113
114
115 040152          BGNMOD  HRDPRM
116
117 040152          BGNHRD
    040152 000030  .WORD  L10105-L$HARD/2
118
119 040154          GPRML  CNTYPE,CNT,1,YES
    040154 005130  .WORD  T$CODE
    040156 040234  .WORD  CNTYPE
    040160 000001  .WORD  1
120 040162          GPRMA  CSRMSG,CSR,0,160000,177776,YES
    040162 000031  .WORD  T$CODE
    040164 040241  .WORD  CSRMSG
    040166 160000  .WORD  T$LOLIM
  
```

121	040170	177776				.WORD	T\$HILIM
	040172					GPRML	DRTYPE,TYPDR,1,YES
	040172	003130				.WORD	T\$CODE
	040174	040266				.WORD	DRTYPE
	040176	000001				.WORD	1
122	040200					GPRMA	VECMMSG,VECT,0,0,776,YES
	040200	001031				.WORD	T\$CODE
	040202	040310				.WORD	VECMMSG
	040204	000000				.WORD	T\$LLOLIM
	040206	000776				.WORD	T\$HILIM
123	040210					GPRMD	BRMSG,PRIOR,0,340,0,7,YES
	040210	002032				.WORD	T\$CODE
	040212	040255				.WORD	BRMSG
	040214	000340				.WORD	340
	040216	000000				.WORD	T\$LLOLIM
	040220	000007				.WORD	T\$HILIM
124	040222					GPRMD	DRMSG,DRBT,0,03400,0,7,YES
	040222	004032				.WORD	T\$CODE
	040224	040317				.WORD	DRMSG
	040226	003400				.WORD	03400
	040230	000000				.WORD	T\$LLOLIM
	040232	000007				.WORD	T\$HILIM
125							
126	040234					ENDHRD	
						.EVEN	
	040234					L10105:	
127							
128	040234	122	114	061		CNTYPE: .ASCIZ	/RL11/
	040237	061	000				
129	040241	102	125	123		CSRMSG: .ASCIZ	/BUS ADDRESS/
	040244	040	101	104			
	040247	104	122	105			
	040252	123	123	000			
130	040255	102	122	040		BRMSG: .ASCIZ	/BR LEVEL/
	040260	114	105	126			
	040263	105	114	000			
131	040266	104	122	111		DRTYPE: .ASCIZ	/DRIVE TYPE = RLC1/
	040271	126	105	040			
	040274	124	131	120			
	040277	105	040	075			
	040302	040	122	114			
	040305	060	061	000			
132	040310	126	105	103		VECMMSG: .ASCIZ	/VECTOR/
	040313	124	117	122			
	040316	000					
133	040317	104	122	111		DRMSG: .ASCIZ	/DRIVE/
	040322	126	105	000			
134						.EVEN	
135							
136	040326					ENDMOD	
137							
138							
139	040326					BGNMOD	SFTPRM
140							
141	040326					BGNSFT	
	040326	000025				.WORD	L10106-L\$SOFT/2
142							

143	040330				GPRML	DMSG,DLT,1,YES
	040330	000130			.WORD	T\$CODE
	040332	040402			.WORD	DMSG
	040334	000001			.WORD	1
144	040336				XFERF	1\$
	040336	006044			.WORD	T\$CODE
145	040340				GPRMD	EMSG,ELT,D,177777,0,177777,YES
	040340	001052			.WORD	T\$CODE
	040342	040520			.WORD	EMSG
	040344	177777			.WORD	177777
	040346	000000			.WORD	T\$LLOLIM
	040350	177777			.WORD	T\$HILIM
146	040352			1\$:	GPRML	SMSG,SIZE,1,YES
	040352	002130			.WORD	T\$CODE
	040354	040426			.WORD	SMSG
	040356	000001			.WORD	1
147	040360				GPRML	CMSG,DMPCK,1,YES
	040360	003130			.WORD	T\$CODE
	040362	040437			.WORD	CMSG
	040364	000001			.WORD	1
148	040366				XFERF	2\$
	040366	006044			.WORD	T\$CODE
149	040370				GPRMD	LMSG,DLMT,D,177777,1,128,YES
	040370	004052			.WORD	T\$CODE
	040372	040463			.WORD	LMSG
	040374	177777			.WORD	177777
	040376	000001			.WORD	T\$LLOLIM
	040400	000200			.WORD	T\$HILIM
150	040402			2\$:		
151						
152						
153	040402				ENDSFT	
					.EVEN	
	040402			L10106:		
154						
155	040402	104	122	117	DMSG:	.ASCIZ /DROP ON ERROR LIMIT/
	040405	120	040	117		
	040410	116	040	105		
	040413	122	122	117		
	040416	122	040	114		
	040421	111	115	111		
	040424	124	000			
156	040426	101	125	124	SMSG:	.ASCIZ /AUTOSIZE/
	040431	117	123	111		
	040434	132	105	000		
157	040437	103	117	115	CMSG:	.ASCIZ /COMPARE DATA ON DCK/
	040442	120	101	122		
	040445	105	040	104		
	040450	101	124	101		
	040453	040	117	116		
	040456	040	104	103		
	040461	113	000			
158	040463	043	040	117	LMSG:	.ASCIZ /# OF WORDS IN ERROR REPORTED/
	040466	106	040	127		
	040471	117	122	104		
	040474	123	040	111		
	040477	116	040	105		

	040502	122	122	117	
	040505	122	040	122	
	040510	105	120	117	
	040513	122	124	105	
	040516	104	000		
159	040520	105	122	122	EMSG: .ASCIZ /ERROR LIMIT/
	040523	117	122	040	
	040526	114	111	115	
	040531	111	124	000	
160					
161	040534				ENDMOD
162					
163					
164	040534				LASTAD
	040534				.EVEN
165					L\$LAST::

1

.SBTTL DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP

```
35 071330 000000      .WORD 0      ;SPACE FOR USER POOL POINTER
36 071332 000000      .WORD 0      ;SIZE
37 071334 000000      .WORD 0      ;CHECKSUM (NOT CURRENTLY USED)
38 071336 000000      .WORD 0      ;SIZE OF H.W. PTAB. ALLOCATION
39          071342      END.SUPV=.*2
40          000200      .END 200
```

ASSEMBLY ROUTINES
SYMBOL TABLE

MACRO V03.01 9-FEB-79 19:01:51 PAGE 153-1

I 11

SEQ 0138

ABOFLA 041060 G	BPRIOR 002264	CURR.T 040544 G	C\$TPRI= 000013	EF08 = 000010 G
ABOPAS 040776 G	BRMSG 040255	CYLSK 002210	C\$UNBU= 000031	EF09 = 000011 G
ABO.FM 043340	BUF 003260	C\$AAD 053102	C\$WTM = 000026	EF10 = 000012 G
AFREG 007357	BVEC 002262	C\$AAE 053114	C\$WTU = 000027	EF11 = 000013 G
AFSI 040566 G	B\$AAB 047624	C\$AAK 054112	DAHS = 000020	EF12 = 000014 G
AFTER 021234	B\$AAF 047536	C\$AAL 054256	DATPAT 003070	EF13 = 000015 G
ALLOC 061500	B.BA 002226	C\$ABRT= 000021	DCKMES 007505	EF14 = 000016 G
ANS = 000012	B.CS 002224	C\$ADR = 000020	DECMG 060024	EF15 = 000017 G
APT.ER 042470	B.DA 002230	C\$AU = 000054	DEMES 007453	EF16 = 000020 G
ARLBA 007314	B.MP 002232	C\$BRK = 000022	DERFLG 002316	ELT = 000002
ARLCS 007307	CALBCC 002164	C\$BSEG= 000004	DERR = 040000	EMSG 040520
ARLDA 007322	CALLPC= 000022	C\$BSUB= 000002	DEV.CO 040546 G	EMT.TR 041064 G
ARLMP 007330	CALLPS= 000024	C\$BUFF= 000030	DIAGMC= 000000	EM1 010560
ASSEMB= 000011	CALLSP= 000026	C\$CEFG= 000046	DIAG.T 041066 G	EM10 011251
A\$AAV 045336	CALLTC= 000030	C\$CLEA= 000012	DLMT = 000010	EM100 010625
A\$AAW 045352	CAL.CL 066222	C\$CLP1= 000006	DLT = 000000	EM11 011316
A\$AAX 045364	CAL.TI 066260 G	C\$CVEC= 000036	DLTIMES 007512	EM12 011345
A\$AAY 045372	CDCNT 002136	C\$DCLN= 000044	DMPCK = 000006	EM13 011404
A\$AAZ 045406	CHECK 002130	C\$DODU= 000053	DMSG 040402	EM14 011436
A\$ABA 045416	CHERR 020432	C\$DRPT= 000024	DPDVD 070476 G	EM16 011520
BA16 = 000020	CHKLUP 047640	C\$DU = 000055	DPMUL 070364 G	EM17 011557
BA17 = 000040	CHKSTR 062042	C\$EDIT= 000002	DRBT = 000010	EM20 011617
BCCFBK 002162	CHKTTY 060130	C\$ERDF= 000002	DRDY = 000001	EM21 011674
BCSR 002260	CHK.MA 046000	C\$ERHR= 000003	DRIVE 002142	EM22 011752
BDDAT 002176	CHK.PC 053130	C\$ERSF= 000001	DRMSG 040317	EM23 012044
BEFORE 021202	CHK.SW 042170	C\$ERSO= 000004	DROP 017176	EM24 012124
BEREG 007336	CHRCNT 061362	C\$ESCA= 000010	DRPCOD 020322 G	EM25 012202
BGN.SU= 040534	CH.FLA 045506	C\$ESEG= 000005	DRST = 000010	EM26 012253
BINMSG 060010	CH.PAS 045524	C\$ESUB= 000003	DRTIM 007425	EM27 012313
BIT0 = 000001 G	CKERLT 020344	C\$ETST= 000001	DRTYPE 040266	EM30 012373
BIT00 = 000001 G	CLEAR. 047122	C\$EXIT= 000032	DSPCOD 017212 G	EM31 012433
BIT01 = 000002 G	CLKACC 040774 G	C\$GMAN= 000043	DS0 = 000000	EM32 012476
BIT02 = 000004 G	CLKBFR 066224	C\$GPHR= 000042	DS1 = 000400	EM33 012543
BIT03 = 000010 G	CLKCNT 040772 G	C\$GPRI= 000040	DS2 = 001000	EM34 012604
BIT04 = 000020 G	CLKJUM 066630 G	C\$GTIM= 000052	DS3 = 001400	EM35 012647
BIT05 = 000040 G	CLKRES 067632 G	C\$INIT= 000011	DUNIT. 041002 G	EM36 012714
BIT06 = 000100 G	CLKSER 067766 G	C\$INLP= 000020	DVC.FT 054062	EM4 011015
BIT07 = 000200 G	CLKSON 041032 G	C\$KWOFF= 000035	DWORD 002216	EM40 012755
BIT08 = 000400 G	CLK.SE 045602	C\$KWON= 000034	D\$AAG 054766	EM41 013020
BIT09 = 001000 G	CLNCOD 020226 G	C\$LOOP= 000100	D\$AAH 055004	EM42 013062
BIT1 = 000002 G	CLR.MA 046056	C\$MANI= 000051	D\$AAI 057552	EM43 013131
BIT10 = 002000 G	CMSG 040437	C\$MSG = 000023	D\$AAJ 057556	EM44 013175
BIT11 = 004000 G	CNT = 000012	C\$PNTB= 000014	D\$AAK 057574	EM45 013232
BIT12 = 010000 G	CNTYPE 040234	C\$PNTF= 000017	D\$AAL 057612	EM47 013264
BIT13 = 020000 G	CNT 064300	C\$PNTS= 000016	D\$AAM 057622	EM5 011054
BIT14 = 040000 G	COMMAN 040604 G	C\$PNTX= 000015	EF.CON= 000036 G	EM50 013313
BIT15 = 100000 G	COMMTA 064114	C\$POIN= 000040	EF.NEW= 000035 G	EM51 013353
BIT2 = 000004 G	COMP 007524	C\$QIO = 000377	EF.PWR= 000034 G	EM52 013405
BIT3 = 000010 G	CONT 017572	C\$RDBU= 000007	EF.RES= 000037 G	EM53 013516
BIT4 = 000020 G	CONTCL 067712 G	C\$REFG= 000050	EF.STA= 000040 G	EM54 013606
BIT5 = 000040 G	CONTIN 017440	C\$REQT= 000045	EF01 = 000001 G	EM55 013436
BIT6 = 000100 G	CRDY = 000200	C\$RESE= 000033	EF02 = 000002 G	EM56 013664
BIT7 = 000200 G	CRLF 060212	C\$REVI= 000002	EF03 = 000003 G	EM57 013705
BIT8 = 000400 G	CRTIM 007400	C\$RPT = 000025	EF04 = 000004 G	EM6 011120
BIT9 = 001000 G	CSR = 000000	C\$SEFG= 000047	EF05 = 000005 G	EM60 013745
BLD.HW 046222	CSRMSG 040241	C\$SPRI= 000041	EF06 = 000006 G	EM61 014003
BLOCK 063634	CURR.S 040542 G	C\$SVEC= 000037	EF07 = 000007 G	EM62 014060

ASSEMBLY ROUTINES
SYMBOL TABLE

MACRO V03.01 9-FEB-79 19:01:51 PAGE 153-2

J 11

SEQ 0139

EM63	014135	FLG.MA	045526	GSTMES	010155	I\$INIT=	000041	L\$HW	017160	G
EM64	014237	FNDFNC	002266	G\$EXCP=	000400	I\$MOD =	000041	L\$ICP	002104	G
EM65	014313	FORM.T	054430	G\$HILI=	000002	I\$MSG =	000041	L\$INIT	017352	G
EM66	014354	FREE	061736	G\$LOLI=	000001	I\$PWR =	000041	L\$LADP	002026	G
EM7	011174	FRMT1	016014	G\$NO =	000000	I\$RPT =	000041	L\$LAST	040534	G
EM70	014426	FRMT10	016623	G\$OFFS=	000400	I\$SEG =	000041	L\$MREV	002050	G
END	020140	FRMT11	016756	G\$OF SI=	000376	I\$SFT =	000041	L\$NAME	002000	G
END.OF	047110	FRMT13	017100	G\$PRMA=	000001	I\$SRV =	000041	L\$REPP	002066	G
END.SU=	071342	FRMT14	016447	G\$PRMD=	000002	I\$SUB =	000041	L\$REV	002010	G
ENVIRO	040606	FRMT15	017131	G\$PRML=	000000	I\$TST =	000041	L\$SOFT	040330	G
EOP.CH	070010	FRMT2	016053	G\$RADA=	000140	J\$JMP =	000167	L\$SPC	002062	G
EOP.FM	043354	FRMT2A	016072	G\$RADB=	000000	KBPTR	040644	L\$SPCP	002020	G
EOP.IN	045520	FRMT2B	016105	G\$RADD=	000040	KBUF	040646	L\$SPTP	002024	G
ERCOUN	002322	FRMT3	016134	G\$RADF=	000200	LDCSR	002154	L\$STA	002030	G
ERFLG	002274	FRMT4	016141	G\$RADL=	000120	LDFUNC	020674	L\$SW	017176	G
ERPOIN	002320	FRMT5	016177	G\$RADO=	000020	LF	007517	L\$TIML	002014	G
ERR =	100000	FRMT6	016250	G\$RADT=	000100	LINE.F	041062	L\$TIMU	002054	G
ERRFOR	054334	FRMT7	016325	G\$XFER=	000004	LINE1	015464	L\$TIM1	002052	G
ERRHAN	053134	FRMT8	016377	G\$YES =	000010	LINE2	015520	L\$TSTI	002100	G
ERRVEC	002140	FRMT9	016520	H\$CORED	045276	LINE3	015742	L\$UNIT	002012	G
ERR.HR	054072	FRMT98	017023	H\$COREQ	045206	LMSG	040463	L.CLK.	045132	
ERR.NU	040536	FRMT99	017075	H\$CORET	041022	LOAD.F	045522	L10000	014466	
ERR.SF	054076	F\$AU =	000015	H\$CRCME	007472	LOGMSG	060032	L10001	014500	
ERRO	014452	F\$BGN =	000040	H\$.ADR	040572	LOPIMN	002300	L10002	014542	
ERR1	014470	F\$CLEA=	000007	H\$.DEF	040564	LOPIMX	002276	L10003	014614	
ERR1FO	054420	F\$DU =	000016	H\$.DIA	040562	LPBFR	040642	L10004	014662	
ERR10	015110	F\$END =	000041	HDHOME	021574	LPCNTR	040640	L10005	014720	
ERR11	015162	F\$HARD=	000004	HDREND	002700	LPT.AD	045164	L10006	014762	
ERR12	015234	F\$HW =	000013	HDRLST	021130	LPT.RE	045160	L10007	014770	
ERR13	015310	F\$INIT=	000006	HDRTAB	002522	LSI.RE	045154	L10010	015042	
ERR14	015356	F\$JMP =	000050	HEND	003066	LUP	066126	L10011	015106	
ERR15	015424	F\$MOD =	000000	HERTZ.	045146	LUP.AD	053132	L10012	015160	
ERR2	014502	F\$MSG =	000011	HNFMES	007500	L\$APT	002036	L10013	015232	
ERR3	014544	F\$PWR =	000017	HOLDS\$P=	000020	L\$AUT	002074	L10014	015306	
ERR4	014616	F\$RPT =	000012	HPTCOD	017156	L\$CCP	002106	L10015	015354	
ERR5	014664	F\$SEG =	000003	HRDPRM	040152	L\$CLEA	020226	L10016	015422	
ERR6	014722	F\$SOFT=	000005	HTAB	002702	L\$CO	002032	L10017	015462	
ERR7	014764	F\$SRV =	000010	HW.ADR	040570	L\$DEPO	002011	L10020	017174	
ERR8	014772	F\$SUB =	000002	H\$AAB	064626	L\$DESC	002102	L10021	017212	
ERR9	015044	F\$SW =	000014	ININIT	041012	L\$DEVP	002064	L10022	020224	
ESC.PC	053126	F\$TEST=	000001	INITCO	017352	L\$DISP	017214	L10023	020320	
EV.CO	040540	GARBAG	061364	INITIA	060040	L\$DR	002112	L10024	020324	
E.BA	002236	GDDAT	002174	INIT.M	046124	L\$DRCT	002070	L10025	020332	
E.CS	002234	GETCHR	060070	INIT.R	040626	L\$DRS	002072	L10026	022316	
E.DA	002240	GETCMN	063454	INPUTA	060766	L\$DRST	002112	L10027	022462	
E.MP	002242	GETPAR	055146	INTEN =	000100	L\$DTP	002040	L10030	022612	
E.MP1	002244	GETSWI	062450	INTFLG	002152	L\$DU	020322	L10031	022746	
E.MP2	002246	GET.TW	062220	INTFOR	054264	L\$DUT	002076	L10032	023100	
FILL	060660	GLBDAT	002126	INTSRV	020326	L\$DVTY	002114	L10033	023236	
FILL.C	000204	GLBEQA	002126	INVAL.	045072	L\$EF	002056	L10034	023434	
FIRST	002206	GLBERR	014452	INVINT	054122	L\$EFLG	002034	L10035	023736	
FIX	021170	GLBSUB	020326	INV.SW	042124	L\$EXP1	002042	L10036	024126	
FLAGS	040600	GLBTXT	007260	IN.SUF	047074	L\$EXP2	002044	L10037	024324	
FLAG\$1	040602	GODRVR=	000202	I\$AU =	000041	L\$EXP3	002046	L10040	024476	
FLAGTA	064032	G\$BIT =	000002	I\$CLN =	000041	L\$HARD	040154	L10041	024674	
FLAG.I	045566	G\$STAT =	000004	I\$DU =	000041	L\$HPCP	002016	L10042	025074	
FLA.SE	064000	G\$TINT	010214	I\$HRD =	000041	L\$HPTP	002022	L10043	025244	

ASSEMBLY ROUTINES
SYMBOL TABLE

L10044	025346	NEXTAR	064216	PWR.UP	071314	SVCINS=	000000	T\$SEK1=	010001
L10045	025472	NOOP0 =	000000	P.CLK.	045140	SVCSUB=	177777	T\$SEK2=	010002
L10046	025666	NOPINT	007636	RDDINT	010304	SVCTAG=	000000	T\$SUBN=	000000
L10047	026022	NOPMES	007605	RDDMES	010253	SVCTST=	177777	T\$TAGL=	177777
L10050	026154	NOPWR	017400	RDHDR =	000010	SVHD	002222	T\$TAGN=	010107
L10051	026274	NORDY	007276	RDNHD =	000016	SWCHAN	045320	T\$TEMP=	000000
L10052	026454	NORES	007260	RDNINT	010455	SWITCH	064172	T\$TEST=	000057
L10053	027266	NO.CLK	045122	RDNMES	010421	SW.ADR	040574 G	T\$TSTM=	177777
L10054	027462	NO.FLA	064012	READ =	000014	SW.PTA	045304	T\$TSTS=	000001
L10055	027626	NO.LPT	061330	READ.P	066230 G	SYS.FT	054052	T\$\$CLE=	010023
L10056	027742	NO.PTA	045326	REGBAC	070720 G	S\$LSYM=	010000	T\$\$DU =	010024
L10057	030120	NR =	000000	REGSAV	070704 G	TEMP2	002200	T\$\$HAR=	010105
L10060	030520	NUMBIN	054454	REQN.P	040610 G	TEMP3	002202	T\$\$HW =	010020
L10061	031142	NUM.LA	054622	REQN.T	045502	TEMP4	002204	T\$\$INI=	010022
L10062	031570	NUM.NO	040576 G	REST	017516	TERMI	066216	T\$\$MSG=	010017
L10063	032250	NUM.UN	041204	RESTMS	020656	TERMLI	064020	T\$\$SEG=	010000
L10064	032702	NUNITS	047612	RE.SET	042272	TERMTA	060002	T\$\$SOF=	010106
L10065	033312	NXM =	020000	RHDINT	010031	TEST.M	045440	T\$\$SRV=	010025
L10066	033544	NXMMES	007460	RHDMES	007771	TIMFLG	040770 G	T\$\$SW =	010021
L10067	034034	NXT	017450	RHHS =	000100	TIMSRV	020334	T\$\$TES=	010104
L10070	034330	NXTFOR	064272	RLBA	002252	TIM.CO	040622 G	T.ANS	017210
L10071	034622	OCTMSG	060016	RLCS	002250	TIM.OP	054426	T.CNTL	002314
L10072	035214	OPI =	002000	RLDA	002254	TMPO	002166	T.CRC	002132
L10073	035514	OPIERR	007532	RLMP	002256	TMP1	002170	T.DMP	017204
L10074	036054	OPIMES	007465	RSTACK	070160 G	TMP2	002172	T.DRIV	002126
L10075	036366	OPIMN	002306	SAVEDO=	042470	TOO.MA	057762	T.LMT	017206
L10076	036712	OPIMX	002310	SEARCH	062166	TRPFLG	002150	T.SIZE	017202
L10077	037236	OSAPTS=	000000	SECMSK	002156	TRPHAN	021566	T1	022050 G
L10100	037326	OSAU =	000000	SEEK =	000006	TRYFNC	002272	T10	024130 G
L10101	037460	OSBGNR=	000000	SEGSTA	041034 G	TST.AB	047750	T11	024326 G
L10102	037656	OSBGNS=	000001	SEKINT	010123	TST.TO	042152	T12	024500 G
L10103	040014	OSDU =	000001	SEKMES	010072	TYPDR =	000006	T13	024676 G
L10104	040150	OSGNSW=	000001	SET.MA	045712	TYPEC	060356	T14	025076 G
L10105	040234	OSPOIN=	000001	SFTPRM	040326 G	TYPEPC	054252	T15	025246 G
L10106	040402	PARSES	063526	SHIFT	071016 G	TYPFLA	063674	T16	025350 G
MAJ.IN	040616 G	PAR.LA	057514	SIGN =	000004	TYPLIN	060254	T17	025474 G
MAJ.IC	066226	PASS.C	040550 G	SIMBCC	021302	TYPNUM	057636	T18	025670 G
MAJ.US	040620 G	PRINTC	061340	SIZE =	000004	TYPSTR	060274	T19	026024 G
MAN.TI	001244	PRINTF	064646	SKHOME	010511	TYP.ER	054102	T2	022320 G
MAP16	070734 G	PRIOR =	000004	SMSG	040426	TY.UNI	047114	T20	026156 G
MASK.B	047636	PRI00 =	000000 G	SPEC.U	045426	T\$ARGC=	000004	T21	026276 G
MASK.W	047634	PRI01 =	000040 G	SPTCOD	017174 G	T\$CODE=	004052	T22	026456 G
MAXCYL	002220	PRI02 =	000100 G	SPV.SE	000400	T\$ERCO=	000062	T23	027270 G
MAXSEC	002214	PRI03 =	000140 G	START	017456	T\$ERRN=	000026	T24	027464 G
MDHEDR	002000 G	PRI04 =	000200 G	STARTC	067706 G	T\$EXCP=	000000	T25	027630 G
MEM.SI	045174	PRI05 =	000240 G	START1	017420	T\$FLAG=	000040	T26	027744 G
MERLMT	017200	PRI06 =	000300 G	STHS =	000100	T\$HILI=	000200	T27	030122 G
MIN.IN	040612 G	PRI07 =	000340 G	STRCHR	060720	T\$LOLI=	000001	T28	030522 G
MIN.US	040614 G	PRNTST	061230	STRT.T	045504	T\$LSYM=	010000	T29	031144 G
MK =	000001	PRO.CM	045500	ST.SET	042336	T\$NEST=	177777	T3	022464 G
MODR	070276 G	PTAB.S	041020 G	SUNIT.	045510	T\$NSK0=	000000	T30	031572 G
MSCRLF	007521	PUTCHR	060044	SUPERV	043372	T\$NSK1=	000005	T31	032252 G
MSG.AD	040560 G	PWRFLG	002312	SUPFLA	041000 G	T\$NSK2=	000003	T32	032704 G
MSG.TY	040534 G	PWR.FA	071170 G	SUPV.T	041152 G	T\$NSK3=	000003	T33	033314 G
MUL	070232 G	PWR.FL	040624 G	SUP.PR	042110	T\$SAVL=	177777	T34	033546 G
MXSEC1	002212	PWR.MS	071316	SVCGBL=	000000	T\$SEGL=	177777	T35	034036 G
NEWPRI	067756 G	PWR.SA	071312	SVCHAN	050026	T\$SEK0=	010000	T36	034332 G

ASSEMBLY ROUTINES
SYMBOL TABLE

MACRO V03.01 9-FEB-79 19:01:51 PAGE 153-4

L 11

SEQ 0141

T37	034624 G	T7	023240 G	VECMSG	040310	XEQDIA	070044 G	XTIMST	066740
T38	035216 G	T8	023436 G	VECT =	000002	XEQSUB	070032 G	XXDP.D	045104
T39	035516 G	T9	023740 G	WCKINT	007730	XEQ.CL	047554	XXX	017500
T4	022614 G	UNITSI	002146	WCKMES	007670	XEQ.CM	045064	XSALWA=	000000
T40	036056 G	UNIT.D	040552 G	WHY	002134	XEQ.IN	047236	XSALS=	000040
T41	036370 G	UNI.MA	045430	WIDTH	055022	XEQ.LA	043326	XSOFFS=	000400
T42	036714 G	UOPIMN	002304	WRCHK =	000002	XEQ.OP	047330	XSTRUE=	000020
T43	037240 G	UOPIMX	002302	WRITE =	000012	XEQ.PR	042530	\$BREG	045600
T44	037330 G	USER.P	041014 G	WRLOCK	010537	XEQ.TE	047374	\$ENDAD	070016 G
T45	037462 G	USER.T	041016 G	WRTINT	010367	XMEM	002270	\$SAV2	071062 G
T46	037660 G	UUT	002144	WRTMES	010335	XPOLY	002160	\$SAV3	071076 G
T47	040016 G	VALID.	041254	WTCRDY	021514	XTIME	066716 G	\$SAV4	071114 G
T5	022750 G	VAL.LA	042074	WTRDRY	021454	XTIMEN	067542	\$SAV5	071134 G
T6	023102 G	VAL.SW	045540						

. ABS. 071340 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 20400 WORDS (80 PAGES)

DYNAMIC MEMORY AVAILABLE FOR 70 PAGES

CZRLHA.BIN,CZRLHA=#SVCRT/M,CZRLHA,DOCTOR