RL11,RLV11

CONTROLLER TEST PART 1

CZRLAB0

AH-E037B-MC

FICHE 1 OF 1

JAN 1979

digital

MADE IN USA

IDENTIFICATION

PRODUCT CODE: AC-E036B-MC

PRODUCT NAME: CZRLAB0 RL11/RLV11 CONTROLLER TEST PART 1

DATE CREATED: 11-OCT-78

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: D. DEKNIS

SEQ 0002

TABLE OF CONTENTS

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 1) IS A PDP-11 (LSI-11) BASED PROGRAM THAT WILL TEST THE CONTROLLER. IT STARTS BY TESTING BASIC INTERFACE LOGIC, REGISTER MANIPULATION AND FUNCTIONALITY WHICH INCLUDES NOOP, GET STATUS, READ HEADERS AND SEEK OPERATIONS. IT IS AIMED AT FULLY TESTING THE CONTROLLER IN THESE AREAS, BUT BY DEFAULT ALSO EXERCISES THE DRIVE. THE TEST COVERAGE OF THE PROGRAM IS EXTREMELY HIGH.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF MEMORY
CONSOLE DEVICE (LA30,LA36,VT50,ETC.)
RL11/RLV11 CONTROLLER(S)
1 - 8 RL01 DRIVES
1 - 8 RL01K CARTRIDGES WITH BAD SECTOR FILE
KW11P, KW11L (OPTIONAL)
LINEPRINTER(OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CZRLAB0 RL11/RLV11 CTLR 1
(FORMERLY MD-11-DZRLA-A)

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

THE RL01 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

CVRLA RLV11 RL01 DISKLESS TEST (RLV11 ONLY)

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RL01 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:

QUESTION	MEANING
L-CLK (L) N ?	IS THERE AN L-CLOCK?
P-CLK (L) N ?	" " " P-CLOCK?
50HZ (L) N ?	IS THE POWER 50 CYCLES (AS IN 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:

```
SIA/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 ONE 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 ENLESSLY 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 REISSUED.

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 AGAINS:

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 ON ERROR).

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

```
PRO/FLAGS:IER:LOE:HOE=0
```

THIS WILL DO THE FOLLOWING:

1. TURN ON THE IER (INHIBIT ERROR PRINTOUT) FLAG

2. TURN ON THE LOE FLAG

3. TURN OFF THE HOE FLAG

4. RESUME EXECUTION AT INSTRUCTION AFTER ERROR REPORT

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

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

1. START

2. RESTART

3. CONTINUE

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

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

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

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

1. START

2. RESTART

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

THE FULL PRINT-OUT FROM THE ABOVE DIALOGUE MIGHT LOOK LIKE THIS:

```
                                                    BY
                                                    WHOM
                                                    ENTERED:
.R DZRKXX                                           O
DZRKXX                                              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
CSR (O) ?                                           D,O
VECTOR (O) ?                                        D,O
BR LEVEL (O) ?                                      D,O
DRIVE (O) ? 0                                       D,O
UNIT 2                                              D
CSR (O) ?                                           D,O
VECTOR (O) ?                                        D,O
BR LEVEL (O) ?                                      D,O
DRIVE (O) ? 1                                       D,O
CHANGE SW (L) ? N                                   D,O
DZRKXX HARD ERR 00004 TST 003 SUB 002 PC:004130     D
ERR HLT                                             D
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                                                  O
DS-B>CON/FLAGS:HOE:IER:LOE=0                        D,O
CHANGE SW (L) ? N                                   D,O
DZRKXX EOP 1                                        D
DS-B>RESTART/PASS:1                                 D,O
CHANGE SW (L) ? N                                   D,O
--------------------
--------------------
--------------------
--------------------
--------------------
```

2.2 HOW TO CREATE A CHAINABLE FILE

THE DIAGNOSTIC AS RECEIVED FROM RELEASE ENGINEERING CANNOT BE RUN IN CHAIN MODE. THA 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*
****************************************

*HICORE 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:

	HOW ENTERED	LEGAL COMMANDS
1.	OPERATOR ENTERED 'RUN DIAG'	START PRINT DISPLAY FLAGS ZFLAGS
2.	DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSED	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 C) ERROR WAS ENCOUNTERED WITH HOE FLAG SET D) OPERATOR ENTERED CONTROL/C.

SEQ 0013

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 NUMBERICAL 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 SUMBITTED. THE DEFAULT IS NON-ENDING EXECUTION..B 'FLAG-LIST' IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR
IER	INHIBIT ERROR REPORTING
IBE	INHIBIT BASIC ERROR REPORTS
IXE	INHIBIT EXTENDED ERROR REPORTS
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER
PNT	PRINT NUMBER OF TES 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 ANSERS IF GIVEN BECOME THE NEW DEFAULTS. THE COMMAND MAY BE ISSUED WHEN COMAND 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 REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. DEFALT FOR PASS-CNT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART

2. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

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

COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS 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, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO-ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

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

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

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

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

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

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

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

```
# UNITS (D) ? 64

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

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

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

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

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

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 RESPONCE.

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) 330?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

BR LEVEL (O) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

DRIVE (O) 0?

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

2.6 SOFTWARE PARAMETERS

THE FOLLOWING QUESTIONS ARE ASKED IF REQUESTED ON A START, RESTART, OR CONTINUE. THEY ALLOW FLEXABILITY 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 ERRORS ARE ENCOUNTERED.

ANSWER Y OR N

ERROR LIMIT (D) 10?

NUMBER OF ERRORS ALLOWED BEFORE DROPPING UNIT.

ANSWER 1 TO 65K

AUTOSIZE (L) N?

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

ANSWER Y OR N

3.0 ERROR INFORMATION

ALL ERRORS ARE PRINTED VIA CONSOLE DEVICE. THE ERROR INCLUDES ERROR NUMBER, TYPE AND PROGRAM LOCATION. ERRORS INCLUDE REGISTERS BEFORE AND AT ERROR WITH RELEVENT DATA.

3.1 ERROR REPORTING

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

FOR SEEK FUNCTION

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

FOR GET STATUS FUNCTION

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

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

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

FOR READ HEADER FUNCTION

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

FOR GET STATUS FUNCTION

HAS DRIVE STATUS

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

6.0 TEST SUMMARIES

TEST 1 - RLCS ADDRESSABILITY

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

TEST 2 - RLBA ADDRESSABILITY

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

TEST 3 - RLDA ADDRESSABILITY

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

TEST 4 - RLMP ADDRESSABILITY

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

TEST 5 - READ WRITE OF RLCS

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

TEST 6 - READ WRITE OF RLBA

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

TEST 7 - READ WRITE OF RLDA

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

TEST 8 - BIS OF RLCS

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

TEST 9 - BIC OF RLCS

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

TEST 10 - BIS OF RLBA

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

TEST 11 - BIC OF RLBA

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

TEST 12 - BIS OF RLDA

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

TEST 13 - BIC OF RLDA

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

TEST 14 - BUS RESET OF RLCS

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

TEST 15 - BUS RESET OF RLBA

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

TEST 16 - BUS RESET OF RLDA

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

TEST 17 - UNIQUENESS OF RLCS

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

TEST 18 - UNIQUENESS OF RLBA

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

TEST 19 - UNIQUENESS OF RLDA

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

TEST 20 - UNIQUENESS OF RLMP

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

TEST 21 - NOOP FUNCTION

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

TEST 22 - TEST NOOP DOES NOTHING

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

TEST 23 - TEST OF INTERRUPT

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

TEST 24 - TEST PRIORITY BR LEVEL

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

TEST 25 - GET STATUS FUNCTION

THIS TEST WILL VERIFY THAT THE GET STATUS FUNCTION (2) WILL

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

TEST 26 - GET STATUS FUNCTION INTERRUPT

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

TEST 27 - GET STATUS FUNCTION GENERATES OPI

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

TEST 28 - OPI UNDER INTERRUPT

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

TEST 30 - READ HEADER FUNCTION INTERRUPT

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

TEST 31 - REPEATED RD HDRS YIELD SAME CYL AND HD

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

TEST 32 - CHECK OF HEADER CRC

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

TEST 33 - CHECK CONSECUTIVE HEADERS

THIS TEST WILL CHECK THAT HEADERS ARE CONSECUTIVE.

TEST 34 - SEEK FUNCTION

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

TEST 35 - CHECK DRIVE READY ON SEEK

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

TEST 36 - SEEK FUNCTION INTERRUPT

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

TEST 37 - TEST DIFFERENCE WORD TRANSMISSION

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

TEST 38 - VERIFY HEAD SELECT 0 VIA RD HEADER

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

TEST 39 - VERIFY HEAD SELECT 1 VIA RD HEADER

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

TEST 40 - VERIFY HEAD SELECT 0 VIA GET STATUS

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

TEST 41 - VERIFY HEAD SELECT 1 VIA GET STATUS

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

TEST 42 - TEST TIME AT WHICH DP WD GETS

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

TEST 43 - EXTENSIVE CHECK OF CRC

THIS TEST WILL MORE EXTENSIVELY CHECK THE CRC LOGIC BY POSITIONING AT DIFFERENT POINTS ON THE PACK AND CHECKING THAT THE HEADER CRC RECEIVED IS CORRECT.

TEST 44 - VERIFY GET STATUS WHILE DRDY IS LOW

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

```
     1                                  .ENABLE AMA
     2                                  .ENABLE ABS
     3                                  .NLIST  ME,CND,MD
     5
     6  000000                          SVC
     7          000000                  SVCINS=0
     8          000000                  SVCTAG=0
     9          002000                  .=2000
    10
    11
    12  002000                                  POINTER BGNSFT,BGNSW,BGNDU,BGNAU
    13
    14
    15  002000                          BGNMOD  MDHEDR
    16
    17  002000                                  HEADER  CZRLA,B,0,60,60,4,RL01
(4) 002000      103                             .ASCII /C/
(4) 002001      132                             .ASCII /Z/
(4) 002002      122                             .ASCII /R/
(4) 002003      114                             .ASCII /L/
(4) 002004      101                             .ASCII /A/
(6) 002005      000                             .BYTE   0
(6) 002006      000                             .BYTE   0
(5) 002007      000                             .BYTE   0
(4) 002010      102                             .ASCII  /B/
(4) 002011      060                             .ASCII  /0/
(4) 002012  000000                              .WORD   0
(4) 002014  000004                              .WORD   4
(4) 002016  023502                              .WORD   L$HARD
(4) 002020  023626                              .WORD   L$SOFT
(4) 002022  011340                              .WORD   L$HW
(4) 002024  011354                              .WORD   L$SW
(4) 002026  024114                              .WORD   L$LAST
(4) 002030  000000                              .WORD   0
(4) 002032  000000                              .WORD   0
(4) 002034  000000                              .WORD   0
(4) 002036  000000                              .WORD   0
(4) 002040  011364                              .WORD   L$DISPATCH
(4) 002042  000000                              .WORD   0
(4) 002044  000000                              .WORD   0
(4) 002046  000000                              .WORD   0
(4) 002050      002                             .BYTE   C$REVISION
(3) 002051      002                             .BYTE   C$EDIT
(4) 002052  000060                              .WORD   60
(4) 002054  000060                              .WORD   60
(4) 002056  000000                              .WORD   0
(5) 002060  000000                              .WORD   0
(4) 002062  000000                              .WORD   0
(4) 002064  002114                              .WORD   L$DVTYP
(4) 002066  000000                              .WORD   0
(4) 002070  002112                              .WORD   L$DR
(4) 002072  002112                              .WORD   L$DRST
(4) 002074  012424                              .WORD   L$AU
(4) 002076  012420                              .WORD   L$DU
(5) 002100  000014                              .WORD   14
(4) 002102  000000                              .WORD   0
```

```
(4)  002104  011514                        .WORD   L$INIT
(4)  002106  012352                        .WORD   L$CLEAN
 18
 19  002110                                 ENDMOD
 20
 21
 22
 23  002110                                 DEVREG
(5)  002110  000000                        .WORD   0
(2)  002112  000001                        .BLKW
 24
 25  002114                                 DEVTYP  <RL01>
(3)  002114  046122  030460      000        .ASCIZ  /RL01/
(2)          002122                        .EVEN
 26  002122                          BGNMOD  GLBEQAT
 27
 28  002122                                 EQUALS
 29          000001                  DRDY=BIT0               ;DRIVE READY (RLCS)
 30          000100                  INTEN=BIT6              ;INTERRUPT ENABLE (RLCS)
 31          100000                  ERR=BIT15               ;RL11 ERROR (RLCS)
 32          040000                  DERR=BIT14              ;RL01 DRIVE ERROR (RLCS)
 33          002000                  OPI=BIT10               ;OPERATION INCOMPLETE (RLCS)
 34          000200                  CRDY=BIT7               ;CONTROLLER READY (RLCS)
 35          000040                  BA17=BIT5               ;EXTENDED ADDRESS BIT 17 (RLCS)
 36          000020                  BA16=BIT4               ;EXTENDED ADDRESS BIT 16 (RLCS)
 37          020000                  NXM=BIT13               ;NON-EXISTANT MEMORY (RLCS)
 38          000000                  DS0=0                   ;DRIVE SELECT 0 (RLCS)
 39          000400                  DS1=BIT8                ;DRIVE SELECT 1 (RLCS)
 40          001000                  DS2=BIT9                ;DRIVE SELECT 2 (RLCS)
 41          001400                  DS3=BIT8!BIT9           ;DRIVE SELECT 3 (RLCS)
 42          000000                  NOOP0=0                 ;FUNCTION-NOOP(0)
 43          000016                  NOOP7=BIT1!BIT2!BIT3    ;FUNCTION-NOOP(7)
 44          000002                  WRCHK=BIT1              ;WRITE CHECK FUNCTION
 45          000004                  GSTAT=BIT2              ;GET STATUS FUNCTION
 46          000006                  SEEK=BIT2!BIT1          ;SEEK FUNCTION
 47          000010                  RDHDR=BIT3              ;READ HEADER FUNCTION
 48          000012                  WRITE=BIT3!BIT1         ;WRITE DATA FUNCTION
 49          000014                  READ=BIT3!BIT2          ;READ DATA FUNCTION
 50          000202                  GODRVR=BIT1!BIT7        ;CRDY AND DRDY
 51          000010                  DRST=BIT3               ;DRIVE RESET (RLDA)
 52          000002                  GSBIT=BIT1              ;GET STATUS BIT (RLDA)
 53          000001                  MK=BIT0                 ;MARKER BIT (RLDA)
 54          000004                  SIGN=BIT2               ;SIGN BIT (RLDA)
 55          000100                  RHHS=BIT6               ;HEAD SELECT IN READ HEADER
 56          000100                  STHS=BIT6               ;HEAD SELECT IN STATUS BACK
 57          000020                  DAHS=BIT4               ;HEAD SELECT IN SEEK
 58
 59                                  ;OFFSET FOR HARDWARE P-TABLE
 60
 61          000000                  CSR=0
 62          000002                  VECT=2
 63          000004                  PRIOR=4
 64          000006                  DRBT=6
 65          000010                  CNT=10
 66
 67                                  ;OFFSET FOR SOFTWARE P-TABLE
```

```
                                        DLT=0
                    000000              ELT=2
                    000002              SIZE=4
                    000004
                                        ENDMOD
002122
                                        BGNMOD  GLBDAT
002122
                                        .SBTTL  GLOBAL DATA
002122  000000      PWRFLG: .WORD   0
002124  000000      UUT:    .WORD   0
002126  000000      UNITST: .WORD   0
002130  000000      RLCS:   .WORD   0
002132  000000      RLBA:   .WORD   0
002134  000000      RLDA:   .WORD   0
002136  000000      RLMP:   .WORD   0
002140  000000      BCSR:   .WORD   0
002142  000000      BPRIOR: .WORD   0
002144  000000      BVEC:   .WORD   0
002146  000000      DRIVE:  .WORD   0           ;DRIVE UNDER TEST
002150  000000      B.CS:   .WORD   0
002152  000000      B.BA:   .WORD   0
002154  000000      B.DA:   .WORD   0
002156  000000      B.MP:   .WORD   0
002160  000000      DERFLG: .WORD
002162  000000      E.CS:   .WORD   0
002164  000000      E.BA:   .WORD   0
002166  000000      E.DA:   .WORD   0
002170  000000      E.MP:   .WORD   0
002172  000000      E.MP1:  .WORD   0
002174  000000      E.MP2:  .WORD   0
002176  000000      PFLG:   .WORD   0           ;PROCESSOR TYPE  0=UNIBUS  1=Q-BUS
002200  000000      TRPFLG: .WORD   0
002202  000000      INTFLG: .WORD   0           ;INTERRUPT OCCURANCE FLAG
002204  000000      LDCSR:  .WORD   0           ;LOCATION TO FORM RLCS
002206  000077      SECMSK: .WORD   77          ;MASK OUT SECTOR
002210  120001      XPOLY:  .WORD   120001
002212  000004      ERRVEC: .WORD   4
002214  000000      BCCFBK: .WORD   0           ;LOCATION USED BY "SIMBCC"
002216  000000      CALBCC: .WORD   0           ;LOCATION USED BY "SIMBCC"
002220  000000      TEMP2:  .WORD   0           ;LOCATION USED BY "SIMBCC"
002222  000000      TEMP3:  .WORD   0           ;LOCATION USED BY "SIMBCC"
002224  000000      TEMP4:  .WORD   0           ;LOCATION USED BY "SIMBCC"
002226  000000      TMP0:   .WORD   0
002230  000000      TMP1:   .WORD   0
002232  000000      TMP2:   .WORD   0
002234  000000      GDDAT:  .WORD   0
002236  000000      BDDAT:  .WORD   0
002240  000000      FIRST:  .WORD   0           ;FIRST SECTOR READ
002242  177700      CYLMSK: .WORD   177700      ;MASK CYLINDER AND HEAD SELECT
002244  000050      MXSEC1: .WORD   40.         ;MAX SECTOR ADDRESS +1
002246  000047      MAXSEC: .WORD   39.         ;MAX SECTOR ADDRESS
002250  000000      DWORD:  .WORD   0           ;DIFFERENCE WORD (SEEK)
002252  077600      MAXCYL: .WORD   77600       ;MAXIMUM CYLINDER ADDRESS
```

```
                                SVHD:   .WORD   0               ;SAVE CURRENT HEAD SELECT
                                WHY:    .WORD   0               ;REASON FOR DROP IN AUTOSIZE
                                T.CNTLR:        .WORD   0
                                TMPFNC: .WORD   0

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

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

                                        000000
                                        1                       ;WALKING 1
                                        2
                                        4
                                        10
                                        20
                                        40
                                        100
                                        200
                                        400
                                        1000
                                        2000
                                        4000
                                        10000
```

```
                                                20000
                                                40000
                                                100000
                                                177777          ;WALKING 0
                                                177776
                                                177775
                                                177773
                                                177767
                                                177757
                                                177737
                                                177677
                                                177577
                                                177377
                                                176777
                                                175777
                                                173777
                                                167777
                                                157777
                                                137777
                                                077777
                                                177777
                                ENDPAT:         000000

                                .SBTTL  PATTERNS FOR DIFFERENCE WORD

                                SKLST:  .WORD   BIT7
                                        .WORD   BIT8            ;SHIFTING 1
                                        .WORD   BIT9
                                        .WORD   BIT10
                                        .WORD   BIT11
                                        .WORD   BIT12
                                        .WORD   BIT13
                                        .WORD   BIT14
                                        .WORD   77600           ;SHIFTING 0
                                        .WORD   77400
                                        .WORD   76600
                                        .WORD   75600
                                        .WORD   73600
                                        .WORD   67600
                                        .WORD   57600
                                        .WORD   37600
                                        .WORD   77600
                                        .WORD   200
                                        .WORD   600             ;GROWING 1
                                        .WORD   1600
                                        .WORD   3600
                                        .WORD   7600
                                QUAMAX: .WORD   17600
                                HALMAX: .WORD   37600
                                        .WORD   77600
                                        .WORD   77400           ;GROWING 0
                                        .WORD   77000
                                        .WORD   76000
                                        .WORD   74000
                                        .WORD   70000
                                        .WORD   60000
```

```
  236  002572  040000                    .WORD   40000
  237  002574  000000                SKEND:  .WORD   00000
  238
  239                                    ;PATTERNS FOR TEST OF RLCS
  240
  241  002576  000000                CSPAT:  .WORD   0               ;SHIFTING 1
  242  002600  000002                        .WORD   BIT1
  243  002602  000004                        .WORD   BIT2
  244  002604  000010                        .WORD   BIT3
  245  002606  000020                        .WORD   BIT4
  246  002610  000040                        .WORD   BIT5
  247  002612  000100                        .WORD   BIT6
  248  002614  000400                        .WORD   BIT8
  249  002616  001000                        .WORD   BIT9
  250  002620  001576                        .WORD   1576            ;GROWING 0
  251  002622  001574                        .WORD   1574
  252  002624  001570                        .WORD   1570
  253  002626  001560                        .WORD   1560
  254  002630  001540                        .WORD   1540
  255  002632  001500                        .WORD   1500
  256  002634  001400                        .WORD   1400
  257  002636  001576                        .WORD   1576            ;SHIFT 0
  258  002640  001574                        .WORD   1574
  259  002642  001566                        .WORD   1566
  260  002644  001556                        .WORD   1556
  261  002646  001536                        .WORD   1536
  262  002650  001436                        .WORD   1436
  263  002652  001136                        .WORD   1136
  264  002654  000076                        .WORD   76
  265  002656  000006                        .WORD   6               ;GROWING 1
  266  002660  000016                        .WORD   16
  267  002662  000036                        .WORD   36
  268  002664  000076                        .WORD   76
  269  002666  000176                        .WORD   176             ;
  270  002670  000576                        .WORD   576
  271  002672  001576                        .WORD   1576
  272  002674  000000                CSEND:  .WORD   0
  273  002676  000000                ERPOINT:        .WORD   0
  274  002700  000074                ERCOUNT:        .BLKW   60.
  275  003070  000240                HDRBUF: .BLKW   160.
  276  003570                         ENDMOD
  277
  278  003570                         BGNMOD  GLBTXT
  279                                 .SBTTL  GLOBAL TEXT
  280
  284  003570  047516  041440  047117  NORES:  .ASCIZ  /NO CONTROLLER/
  285  003606  047516  042040  044522  NODRY:  .ASCIZ  /NO DRIVE CONNECTED/
  286  003631     040  051104  000126  DEMES:  .ASCIZ  / DRV/
  287  003636  047040  046530     000  NXMMES: .ASCIZ  / NXM/
  288  003643     040  050117  000111  OPIMES: .ASCIZ  / OPI/
  289  003650  044040  051103  000103  HCRCMES:.ASCIZ  / HCRC/
  290  003656  044040  043116     000  HNFMES: .ASCIZ  / HNF/
  291  003663     040  041504  000113  DCKMES: .ASCIZ  / DCK/
  292  003670  042040  052114     000  DLTMES: .ASCIZ  / DLT/
  293  003675     015  000012          MSCRLF: .ASCIZ  <15><12>
  294  003700  000015                  LF:     .ASCIZ  <15>
```

```
  295  003702  041440  046517  000120  COMP:    .ASCIZ  / COMP/
  296  003710  047506  041522  042105  OPIERR:  .ASCIZ  /FORCED OPI(GET STATUS) CAUSED OTHER ERRORS/
  297  003763     116  047517  020120  NOPMES:  .ASCIZ  /NOOP OPERATION-FLAG MODE/
  298  004014  047516  050117  047440  NOPINT:  .ASCIZ  /NOOP OPERATION-INTR. MODE/
  299  004046  051127  052111  020105  WCKMES:  .ASCIZ  /WRITE CHECK OPERATION-FLAG MODE/
  300  004106  051127  052111  020105  WCKINT:  .ASCIZ  /WRITE CHECK OPERATION-INTR. MODE/
  301  004147     122  040505  020104  RHDMES:  .ASCIZ  /READ HEADER OPERATION-FLAG MODE/
  302  004207     122  040505  020104  RHDINT:  .ASCIZ  /READ HEADER OPERATION-INTR. MODE/
  303  004250  042523  045505  047440  SEKMES:  .ASCIZ  /SEEK OPERATION-FLAG MODE/
  304  004301     123  042505  020113  SEKINT:  .ASCIZ  /SEEK OPERATION-INTR. MODE/
  305  004333     107  052105  051440  GSTMES:  .ASCIZ  /GET STATUS OPERATION-FLAG MODE/
  306  004372  042507  020124  052123  GSTINT:  .ASCIZ  /GET STATUS OPERATION-INTR MODE/
  307  004431     103  035123  000040  ARLCS:   .ASCIZ  /CS: /
  308  004436  041040  035101  000040  ARLBA:   .ASCIZ  / BA: /
  309  004444  042040  035101  000040  ARLDA:   .ASCIZ  / DA: /
  310  004452  046440  035120  000040  ARLMP:   .ASCIZ  / MP: /
  311  004460  042502  047506  042522  BEREG:   .ASCIZ  /BEFORE COMMAND: /
  312  004501     124  046511  020105  AFREG:   .ASCIZ  /TIME OF ERROR:  /
  313  004522  047503  052116  047522  CRTIM:   .ASCIZ  /CONTROLLER TIMED OUT/
  314  004547     104  044522  042526  DRTIM:   .ASCIZ  /DRIVE READY TIMED OUT/
  315  004575     103  047101  047040  EM1:     .ASCIZ  /CAN NOT ADDRESS RLCS/
  316  004622  040503  020116  047516  EM2:     .ASCIZ  /CAN NOT ADDRESS RLBA/
  317  004647     103  047101  047040  EM3:     .ASCIZ  /CAN NOT ADDRESS RLDA/
  318  004674  040503  020116  047516  EM4:     .ASCIZ  /CAN NOT ADDRESS RLMP/
  319  004721     122  041514  020123  EM5:     .ASCIZ  %RLCS READ/WRITE ERROR (BIT 0 DON'T CARE)%
  320  004772  046122  040502  051040  EM6:     .ASCIZ  %RLBA READ/WRITE ERROR%
  321  005020  046122  040504  051040  EM7:     .ASCIZ  %RLDA READ/WRITE ERROR%
  322  005046  050117  020111  047527  EM11:    .ASCIZ  /OPI WOULD NOT GENERATE INTERRUPT/
  323  005107     116  020117  047111  EM13:    .ASCIZ  /NO INTERRUPT FROM NOOP(0)/
  324  005141     116  047517  024120  EM14:    .ASCIZ  /NOOP(0) MODIFIED RLMP/
  325  005167     116  047517  024120  EM15:    .ASCIZ  /NOOP(0) MODIFIED RLBA/
  326  005215     116  047517  024120  EM16:    .ASCIZ  /NOOP(0) MODIFIED RLDA/
  327  005243     111  052116  051105  EM17:    .ASCIZ  /INTERRUPT PRIORITY FAILURE/
  328  005276  042507  020124  052123  EM30:    .ASCIZ  /GET STATUS WOULD NOT INTERRUPT/
  329  005335     107  052105  051440  EM30A:   .ASCIZ  /GET STATUS SHOULD NOT INTERRUPT/
  330  005375     122  046514  020120  EM32:    .ASCIZ  /RLMP CONTAINED WRONG STATUS/
  331  005431     117  044520  042040  EM33:    .ASCIZ  /OPI DID NOT SET-GSTAT WITHOUT GS BIT/
  332  005476  050117  020111  044504  EM34:    .ASCIZ  /OPI DID NOT SET-GSTAT WITHOUT GS AND MK BITS/
  333  005553     122  040505  020104  EM37:    .ASCIZ  /READ HEADER WOULD NOT INTERRUPT/
  334  005613     102  042101  041440  EM41:    .ASCIZ  /BAD CYLINDER OR HEAD SELECT IN REPEATED READ HEADER TEST/
  335  005704  040502  020104  042510  EM42:    .ASCIZ  /BAD HEADER CRC ON READ HEADER/
  336  005742  042523  052103  051117  EM43:    .ASCIZ  /SECTOR ADDRESS OUT OF SEQUENCE DURING CONSECUTIVE READ HEADERS/
  337  006041     127  044522  044524  EM44:    .ASCIZ  /WRITING RLMP MODIFIED RLCS/
  338  006074  051127  052111  047111  EM45:    .ASCIZ  /WRITING RLMP MODIFIED RLBA/
  339  006127     127  044522  044524  EM46:    .ASCIZ  /WRITING RLMP MODIFIED RLDA/
  340  006162  042523  045505  053440  EM47:    .ASCIZ  /SEEK WOULD NOT INTERRUPT/
  341  006213     104  044522  042526  EM52:    .ASCIZ  /DRIVE READY CAUSED EXTRANEOUS INTERRUPT/
  342  006263     102  042101  051440  EM54:    .ASCIZ  /BAD SEEK-TEST OF DIFFENCE WORD/
  343  006322  040502  020104  042510  EM55:    .ASCIZ  /BAD HEAD SELECT VIA RD HDR/
  344  006355     102  042101  044040  EM56:    .ASCIZ  /BAD HEAD SELECT VIA GET STATUS/
  345  006414  047514  042101  047111  EM57:    .ASCII  /LOADING RLDA BEFORE DRIVE READY ON SEEK/<15><12>
  346  006465     104  044522  042526           .ASCIZ  /DRIVE READY DID NOT SET/
  347  006515     102  052111  051440  EM61:    .ASCIZ  /BIT SET INSTRUCTION ON RLCS YIELDED WRONG RESULT/
  348  006576  044502  020124  046103  EM62:    .ASCIZ  /BIT CLEAR INSTRUCTION ON RLCS YIELDED WRONG RESULT/
  349  006661     102  052111  051440  EM63:    .ASCIZ  /BIT SET INSTRUCTION ON RLBA YIELDED WRONG RESULT/
  350  006742  044502  020124  046103  EM64:    .ASCIZ  /BIT CLEAR INSTRUCTION ON RLBA YIELDED WRONG RESULT/
```

```
                                                                L 3
  351  007025    102  052111  051440  EM65:   .ASCIZ  /BIT SET INSTRUCTION ON RLDA YIELDED WRONG RESULT/
  352  007106 044502  020124  046103  EM66:   .ASCIZ  /BIT CLEAR INSTRUCTION ON RLDA YIELDED WRONG RESULT/
  353  007171    102  051525  051040  EM67:   .ASCIZ  /BUS RESET DID NOT CLEAR RLCS/
  354  007226 052502  020123  042522  EM70:   .ASCIZ  /BUS RESET DID NOT CLEAR RLBA/
  355  007263    102  051525  051040  EM71:   .ASCIZ  /BUS RESET DID NOT CLEAR RLDA/
  356  007320 051127  052111  047111  EM72:   .ASCIZ  /WRITING RLCS MODIFIED RLBA/
  357  007353    127  044522  044524  EM73:   .ASCIZ  /WRITING RLCS MODIFIED RLDA/
  358  007406 051127  052111  047111  EM74:   .ASCIZ  /WRITING RLBA MODIFED RLCS/
  359  007440 051127  052111  047111  EM75:   .ASCIZ  /WRITING RLBA MODIFED RLDA/
  360  007472 051127  052111  047111  EM76:   .ASCIZ  /WRITING RLDA MODIFIED RLCS/
  361  007525    127  044522  044524  EM77:   .ASCIZ  /WRITING RLDA MODIFIED RLBA/
  362  007560 046122  051503  041440  EM101:  .ASCIZ  /RLCS CONTAINED FOLLOWING ERROR(S):  /
  363  007625 000170                  EM102:  .BLKB   120.
  364
  365          010016                 .EVEN
  366
  370  010016                                 ENDMOD
  371
  372                                 .SBTTL  GLOBAL ERRORS
  373
  374  010016                         BGNMOD  GLBERR
  375
  376  010016                         BGNMSG  ERR0
  377
  378  010016  004737  010342                JSR     PC,LINE1
  379  010022  004737  010376                JSR     PC,LINE2
  380
  381  010026  004537  012430                JSR     R5,CKERLT       ;CHECK ERROR LIMIT
  382  010032                                 ENDMSG
  (3)  010032                         L10000:
  (3)  010032  104023                        EMT     C$MSG
  383
  384  010034                         BGNMSG  ERR1
  385
  386  010034  004737  010342                JSR     PC,LINE1
  387
  388  010040  004537  012430                JSR     R5,CKERLT       ;CHECK ERROR LIMIT
  389  010044                                 ENDMSG
  (3)  010044                         L10001:
  (3)  010044  104023                        EMT     C$MSG
  390
  391  010046                         BGNMSG  ERR2
  392
  393  010046  004737  010342                JSR     PC,LINE1
  394  010052                                 PRINTB  #FRMT4,GDDAT,BDDAT
  (9)  010052  013746  002236                MOV     BDDAT,-(SP)
  (8)  010056  013746  002234                MOV     GDDAT,-(SP)
  (7)  010062  012746  011020                MOV     #FRMT4,-(SP)
  (6)  010066  012746  000003                MOV     #3,-(SP)
  (3)  010072  010600                        MOV     SP,R0
  (4)  010074  104014                        EMT     C$PNTB
  (4)  010076  062706  000010                ADD     #10,SP
  395
  396  010102  004537  012430                JSR     R5,CKERLT       ;CHECK ERROR LIMIT
  397  010106                                 ENDMSG
  (3)  010106                         L10002:
```

```
   (3)  010106  104023                    EMT     C$MSG
398
399  010110                        BGNMSG  ERR3
400
401  010110  004737  010342                JSR     PC,LINE1
402  010114  004737  010376                JSR     PC,LINE2
403  010120                                PRINTB  #FRMT5,TMP0,BDDAT,GDDAT
(10) 010120  013746  002234                MOV     GDDAT,-(SP)
(9)  010124  013746  002236                MOV     BDDAT,-(SP)
(8)  010130  013746  002226                MOV     TMP0,-(SP)
(7)  010134  012746  011056                MOV     #FRMT5,-(SP)
(6)  010140  012746  000004                MOV     #4,-(SP)
(3)  010144  010600                        MOV     SP,R0
(4)  010146  104014                        EMT     C$PNTB
(4)  010150  062706  000012                ADD     #12,SP
404
405  010154  004537  012430                JSR     R5,CKERLT       ;CHECK ERROR LIMIT
406  010160                                ENDMSG
(3)  010160                           L10003:
(3)  010160  104023                        EMT     C$MSG
407
408  010162                        BGNMSG  ERR4
409
410  010162  004737  010342                JSR     PC,LINE1
411  010166  004737  010376                JSR     PC,LINE2
412  010172                                PRINTB  #FRMT4,GDDAT,BDDAT
(9)  010172  013746  002236                MOV     BDDAT,-(SP)
(8)  010176  013746  002234                MOV     GDDAT,-(SP)
(7)  010202  012746  011020                MOV     #FRMT4,-(SP)
(6)  010206  012746  000003                MOV     #3,-(SP)
(3)  010212  010600                        MOV     SP,R0
(4)  010214  104014                        EMT     C$PNTB
(4)  010216  062706  000010                ADD     #10,SP
413
414  010222  004537  012430                JSR     R5,CKERLT       ;CHECK ERROR LIMIT
415  010226                                ENDMSG
(3)  010226                           L10004:
(3)  010226  104023                        EMT     C$MSG
416
417  010230                        BGNMSG  ERR5
418
419  010230  004737  010342                JSR     PC,LINE1
420
421  010234  004537  012430                JSR     R5,CKERLT       ;CHECK ERROR LIMIT
422  010240                                ENDMSG
(3)  010240                           L10005:
(3)  010240  104023                        EMT     C$MSG
423
424  010242                        BGNMSG  ERR6
425
426  010242  004737  010342                JSR     PC,LINE1
427  010246  004737  010620                JSR     PC,LINE3
428  010252  004737  010376                JSR     PC,LINE2
429
430
431  010256                        1$:     PRINTB  #FRMT99
```

```
(7)  010256  012746  011053                MOV     #FRMT99,-(SP)
(6)  010262  012746  000001                MOV     #1,-(SP)
(3)  010266  010600                        MOV     SP,R0
(4)  010270  104014                        EMT     C$PNTB
(4)  010272  062706  000004                ADD     #4,SP
432  010276  004537  012430                JSR     R5,CKERLT       ;CHECK ERROR LIMIT
433  010302                                ENDMSG
(3)  010302                          L10006:
(3)  010302  104023                        EMT     C$MSG
434
435  010304                          BGNMSG  ERR7
436
437  010304  004737  010342                JSR     PC,LINE1
438  010310                                PRINTB  #FRMT6,BDDAT
(8)  010310  013746  002236                MOV     BDDAT,-(SP)
(7)  010314  012746  011127                MOV     #FRMT6,-(SP)
(6)  010320  012746  000002                MOV     #2,-(SP)
(3)  010324  010600                        MOV     SP,R0
(4)  010326  104014                        EMT     C$PNTB
(4)  010330  062706  000006                ADD     #6,SP
439
440  010334  004537  012430                JSR     R5,CKERLT
441
442  010340                                ENDMSG
(3)  010340                          L10007:
(3)  010340  104023                        EMT     C$MSG
443
444  010342                          LINE1:  PRINTB  #FRMT1,RLCS,<B,DRIVE+1>
(9)  010342  005046                        CLR     -(SP)
(9)  010344  153716  002147                BISB    DRIVE+1,(SP)
(8)  010350  013746  002130                MOV     RLCS,-(SP)
(7)  010354  012746  010672                MOV     #FRMT1,-(SP)
(6)  010360  012746  000003                MOV     #3,-(SP)
(3)  010364  010600                        MOV     SP,R0
(4)  010366  104014                        EMT     C$PNTB
(4)  010370  062706  000010                ADD     #10,SP
445  010374  000207                        RTS     PC
446
447  010376                          LINE2:  PRINTB  #FRMT2,#BEREG,#ARLCS,B.CS,#ARLBA,B.BA
(12) 010376  013746  002152                MOV     B.BA,-(SP)
(11) 010402  012746  004436                MOV     #ARLBA,-(SP)
(10) 010406  013746  002150                MOV     B.CS,-(SP)
(9)  010412  012746  004431                MOV     #ARLCS,-(SP)
(8)  010416  012746  004460                MOV     #BEREG,-(SP)
(7)  010422  012746  010732                MOV     #FRMT2,-(SP)
(6)  010426  012746  000006                MOV     #6,-(SP)
(3)  010432  010600                        MOV     SP,R0
(4)  010434  104014                        EMT     C$PNTB
(4)  010436  062706  000016                ADD     #16,SP
448  010442                                PRINTB  #FRMT2A,#ARLDA,B.DA,#ARLMP,B.MP
(11) 010442  013746  002156                MOV     B.MP,-(SP)
(10) 010446  012746  004452                MOV     #ARLMP,-(SP)
(9)  010452  013746  002154                MOV     B.DA,-(SP)
(8)  010456  012746  004444                MOV     #ARLDA,-(SP)
(7)  010462  012746  010751                MOV     #FRMT2A,-(SP)
(6)  010466  012746  000005                MOV     #5,-(SP)
```

```
   (3)  010472  010600                        MOV     SP,R0
   (4)  010474  104014                        EMT     C$PNTB
   (4)  010476  062706  000014                ADD     #14,SP
   449  010502                                PRINTB  #FRMT2,#AFREG,#ARLCS,E.CS,#ARLBA,E.BA
   (12) 010502  013746  002164                MOV     E.BA,-(SP)
   (11) 010506  012746  004436                MOV     #ARLBA,-(SP)
   (10) 010512  013746  002162                MOV     E.CS,-(SP)
   (9)  010516  012746  004431                MOV     #ARLCS,-(SP)
   (8)  010522  012746  004501                MOV     #AFREG,-(SP)
   (7)  010526  012746  010732                MOV     #FRMT2,-(SP)
   (6)  010532  012746  000006                MOV     #6,-(SP)
   (3)  010536  010600                        MOV     SP,R0
   (4)  010540  104014                        EMT     C$PNTB
   (4)  010542  062706  000016                ADD     #16,SP
   450  010546                                PRINTB  #FRMT2B,#ARLDA,E.DA,#ARLMP,E.MP,E.MP1,E.MP2
   (13) 010546  013746  002174                MOV     E.MP2,-(SP)
   (12) 010552  013746  002172                MOV     E.MP1,-(SP)
   (11) 010556  013746  002170                MOV     E.MP,-(SP)
   (10) 010562  012746  004452                MOV     #ARLMP,-(SP)
   (9)  010566  013746  002166                MOV     E.DA,-(SP)
   (8)  010572  012746  004444                MOV     #ARLDA,-(SP)
   (7)  010576  012746  010764                MOV     #FRMT2B,-(SP)
   (6)  010602  012746  000007                MOV     #7,-(SP)
   (3)  010606  010600                        MOV     SP,R0
   (4)  010610  104014                        EMT     C$PNTB
   (4)  010612  062706  000020                ADD     #20,SP
   451  010616  000207                        RTS     PC
   452
   453  010620                        LINE3:  PRINTB  #FRMT3,#EM101
   (8)  010620  012746  007560                MOV     #EM101,-(SP)
   (7)  010624  012746  011013                MOV     #FRMT3,-(SP)
   (6)  010630  012746  000002                MOV     #2,-(SP)
   (3)  010634  010600                        MOV     SP,R0
   (4)  010636  104014                        EMT     C$PNTB
   (4)  010640  062706  000006                ADD     #6,SP
   454  010644                                PRINTB  #FRMT3,#EM102
   (8)  010644  012746  007625                MOV     #EM102,-(SP)
   (7)  010650  012746  011013                MOV     #FRMT3,-(SP)
   (6)  010654  012746  000002                MOV     #2,-(SP)
   (3)  010660  010600                        MOV     SP,R0
   (4)  010662  104014                        EMT     C$PNTB
   (4)  010664  062706  000006                ADD     #6,SP
   455  010670  000207                        RTS     PC
   456
   460
   461  010672  040445  047503  052116  FRMT1:  .ASCIZ  /%ACONTROLLER: %06%A  DRIVE: %01/
   462  010732  047045  052045  052045  FRMT2:  .ASCIZ  /%N%T%T%06%T%06/
   463  010751     045  022524  033117  FRMT2A: .ASCIZ  /%T%06%T%06/
   464  010764  052045  047445  022466  FRMT2B: .ASCIZ  /%T%06%T%06%A %06%A %06/
   465  011013     045  022516  000124  FRMT3:  .ASCIZ  /%N%T/
   466  011020  047045  040445  054105  FRMT4:  .ASCII  /%N%AEXP'D: %06%A REC'D: %06/
   467  011053     045  000116          FRMT99: .ASCIZ  /%N/
   468  011056  047045  040445  040514  FRMT5:  .ASCIZ  /%N%ALAST: %06%A PRES: %06%A EXP'D: %06%N/
   469  011127     045  022516  040501  FRMT6:  .ASCIZ  /%N%AAT PROCESSOR LEVEL %06%N/
   470  011164  040445  051105  047522  FRMT11: .ASCIZ  /%^ERROR LIMIT EXCEEDED-DROPPED%N/
   471  011225     045  022516  042101  FRMT12: .ASCIZ  /%N%ADRIVE DID NOT RECOVER FROM POWER FAILURE%N/
```

```
                                                                C 4
 472  011304  047045  052045  040445  FRMT13: .ASCIZ  /%N%T%A - WILL NOT TEST%N/
 473
 474          011336                  .EVEN
 475
 476
 480
 481
 482
 483  011336                                  ENDMOD
 484
 485  011336                          BGNMOD  HPTCODE
 486
 487  011336                          BGNHW
 (3)  011336  000005                          .WORD   L10010-L$HW/2
 488  011340  174400                          .WORD   174400          ;CSR
 489  011342  000160                          .WORD   160             ;VECTOR
 490  011344  000240                          .WORD   240             ;PRIORITY
 491  011346  000000                          .WORD   0               ;DRIVE (BITS 8,9,10)
 492  011350  000001                          .WORD   1               ;RL11 = 1, RLV11 = 0
 493
 494  011352                          ENDHW
 (3)  011352                          L10010:
 495
 496  011352                                  ENDMOD
 497
 498  011352                          BGNMOD  SPTCODE
 499
 500  011352                          BGNSW
 (3)  011352  000003                          .WORD   L10011-L$SW/2
 501
 502  011354  000000                  DROP:   .WORD   0
 503  011356  000012                  MERLMT: .WORD   10.
 504  011360  000000                  T.SIZE: .WORD   0
 505
 506  011362                          ENDSW
 (3)  011362                          L10011:
 507
 508  011362                                  ENDMOD
 509
 510  011362                          BGNMOD  DSPCODE
 511
 512  011362                          DISPATCH        44
 (4)  011362  000054                          .WORD   44
 (6)  011364  013702                          .WORD   T1
 (6)  011366  013776                          .WORD   T2
 (6)  011370  014072                          .WORD   T3
 (6)  011372  014166                          .WORD   T4
 (6)  011374  014262                          .WORD   T5
 (6)  011376  014402                          .WORD   T6
 (6)  011400  014504                          .WORD   T7
 (6)  011402  014572                          .WORD   T8
 (6)  011404  014716                          .WORD   T9
 (6)  011406  015042                          .WORD   T10
 (6)  011410  015146                          .WORD   T11
 (6)  011412  015246                          .WORD   T12
 (6)  011414  015336                          .WORD   T13
```

```
(6)  011416  015436                          .WORD   T14
(6)  011420  015546                          .WORD   T15
(6)  011422  015620                          .WORD   T16
(6)  011424  015656                          .WORD   T17
(6)  011426  016002                          .WORD   T18
(6)  011430  016142                          .WORD   T19
(6)  011432  016302                          .WORD   T20
(6)  011434  016506                          .WORD   T21
(6)  011436  016536                          .WORD   T22
(6)  011440  016742                          .WORD   T23
(6)  011442  017026                          .WORD   T24
(6)  011444  017172                          .WORD   T25
(6)  011446  017222                          .WORD   T26
(6)  011450  017374                          .WORD   T27
(6)  011452  017462                          .WORD   T28
(6)  011454  017610                          .WORD   T29
(6)  011456  017632                          .WORD   T30
(6)  011460  017712                          .WORD   T31
(6)  011462  020056                          .WORD   T32
(6)  011464  020214                          .WORD   T33
(6)  011466  020532                          .WORD   T34
(6)  011470  020570                          .WORD   T35
(6)  011472  020634                          .WORD   T36
(6)  011474  020760                          .WORD   T37
(6)  011476  021354                          .WORD   T38
(6)  011500  021506                          .WORD   T39
(6)  011502  021650                          .WORD   T40
(6)  011504  022010                          .WORD   T41
(6)  011506  022162                          .WORD   T42
(6)  011510  022510                          .WORD   T43
(6)  011512  023206                          .WORD   T44
513
514  011514                                  ENDMOD
515
516                                  .SBTTL  INITIALIZATION CODE
517  011514                          BGNMOD  INITCODE
518
519  011514                                  BGNINIT
520
521  011514                                  BRESET
(3)  011514  104033                          EMT     C$RESET
522  011516                                  READEF  #EF.PWR                 ;POWER UP?????
(3)  011516  012700  000034                  MOV     #EF.PWR,R0
(3)  011522  104050                          EMT     C$REFG
523  011524                                  BNCOMPLETE      NOPWR           ;NO,BRANCH
(2)  011524  103004                          BCC     NOPWR
524  011526  013737  002012  002122          MOV     L$UNIT,PWRFLG           ;YES, SET POWER FLAG
525  011534  000473                          BR      CONT                    ;GO TO CONTINUE POINT
526  011536                          NOPWR:  READEF  #EF.RESTART     ;RESTART?
(3)  011536  012700  000037                  MOV     #EF.RESTART,R0
(3)  011542  104050                          EMT     C$REFG
527  011544                                  BCOMPLETE START1
(2)  011544  103404                          BCS     START1
528  011546                                  READEF  #EF.START               ;START???
(3)  011546  012700  000040                  MOV     #EF.START,R0
(3)  011552  104050                          EMT     C$REFG
```

```
 529  011554                                  BNCOMPLETE      CONTINUE
(2)   011554  103010                          BCC     CONTINUE
 530  011556  012700  002700          START1: MOV     #ERCOUNT,R0
 531  011562  012701  000100                  MOV     #64.,R1
 532  011566  005020                  1$:     CLR     (R0)+
 533  011570  005301                          DEC     R1
 534  011572  001375                          BNE     1$
 535
 536  011574  000407                          BR      START
 537  011576                          CONTINUE:       READEF  #EF.CONTINUE    ;CONTINUE????
(3)   011576  012700  000036                  MOV     #EF.CONTINUE,R0
(3)   011602  104050                          EMT     C$REFG
 538  011604                                  BCOMPLETE       CONT
(2)   011604  103447                          BCS     CONT
 539
 540  011606  005737  002124          NXT:    TST     UUT                     ;DONE ALL UUT'S
 541  011612  001011                          BNE     XXX                     ;NO
 542  011614  012737  177777  002126  START:  MOV     #-1,UNITST
 543  011622  013737  002012  002124          MOV     L$UNIT,UUT
 544  011630  012737  002676  002676          MOV     #ERCOUNT-2,ERPOINT
 545
 546  011636  005237  002126          XXX:    INC     UNITST
 547  011642  062737  000002  002676          ADD     #2,ERPOINT
 548  011650  005337  002124                  DEC     UUT
 549  011654                          REST:   GPHARD  UNITST,R0
(3)   011654  013700  002126                  MOV     UNITST,R0
(3)   011660  104042                          EMT     C$GPHRD
 550  011662                                  BCOMPLETE       1$
(2)   011662  103406                          BCS     1$
 551  011664  005737  002122                  TST     PWRFLG                  ;POWER FLAG TO 0
 552  011670  001746                          BEQ     NXT                     ;YES, DONT DEC IT
 553  011672  005337  002122                  DEC     PWRFLG
 554  011676  000743                          BR      NXT                     ;GET NEXT ONE
 555  011700  012037  002140          1$:     MOV     (R0)+,BCSR
 556  011704  012037  002144                  MOV     (R0)+,BVEC
 557  011710  012037  002142                  MOV     (R0)+,BPRIOR
 558  011714  012037  002146                  MOV     (R0)+,DRIVE
 559  011720  012037  002260                  MOV     (R0)+,T.CNTLR   ;GET CONTROLLER TYPE
 560
 561  011724  013700  002140          CONT:   MOV     BCSR,R0
 562  011730  010037  002130                  MOV     R0,RLCS
 563  011734  062700  000002                  ADD     #2,R0
 564  011740  010037  002132                  MOV     R0,RLBA
 565  011744  062700  000002                  ADD     #2,R0
 566  011750  010037  002134                  MOV     R0,RLDA
 567  011754  062700  000002                  ADD     #2,R0
 568  011760  010037  002136                  MOV     R0,RLMP
 569  011764  005737  002122                  TST     PWRFLG
 570  011770  001064                          BNE     5$
 571  011772  005737  011360                  TST     T.SIZE                  ;DO WE WANT TO CHECK UNITS??
 572  011776  001461                          BEQ     5$                      ;NO
 573
 574  012000  005037  002200                  CLR     TRPFLG                  ;CLR OUT TRAP FLAG
 575  012004                                  SETVEC  ERRVEC,#TRPHAN,#340     ;SETUP VECTOR TO CATCH NON-EXIST
(7)   012004  012746  000340                  MOV     #'0,-(SP)
(6)   012010  012746  013544                  MOV     #TRPHAN,-(SP)
```

```
    (5) 012014 013746 002212                    MOV     ERRVEC,-(SP)
    (4) 012020 012746 000003                    MOV     #3,-(SP)
    (3) 012024 104037                           EMT     C$SVEC
    (2) 012026 062706 000010                    ADD     #10,SP
576 012032 005777 170072                        TST     @RLCS           ;ACCESS CONTROLLER
577 012036                                      CLRVEC  ERRVEC          ;RELEASE VECTOR
    (3) 012036 013700 002212                    MOV     ERRVEC,R0
    (3) 012042 104036                           EMT     C$CVEC
578 012044 005737 002200                        TST     TRPFLG          ;DID IT TRAP
579 012050 001404                               BEQ     7$              ;NO, CHECK IT'S DRIVE
580 012052 012737 003570 002256                 MOV     #NORES,WHY      ;SETUP ERR MESS
581 012060 000415                               BR      8$
582
583 012062 012777 000200 170040 7$:             MOV     #200,@RLCS      ;CONTROLLER READY
584 012070 053777 002146 170032                 BIS     DRIVE,@RLCS     ;SELECT DRIVE
585 012076 032777 000001 170024                 BIT     #1,@RLCS        ;DRIVE THERE
586 012104 001016                               BNE     5$              ;YES
587 012106 012737 003606 002256                 MOV     #NODRY,WHY      ;SETUP ERR MESS
588 012114                              8$:     PRINTB  #FRMT13,WHY
    (8) 012114 013746 002256                    MOV     WHY,-(SP)
    (7) 012120 012746 011304                    MOV     #FRMT13,-(SP)
    (6) 012124 012746 000002                    MOV     #2,-(SP)
    (3) 012130 010600                           MOV     SP,R0
    (4) 012132 104014                           EMT     C$PNTB
    (4) 012134 062706 000006                    ADD     #6,SP
589 012140 000434                               BR      6$
590
591 012142 005737 002122                5$:     TST     PWRFLG          ;RECENT POWER FAILURE????
592 012146 001457                               BEQ     END             ;NO
593
594                                     ;THERE WAS A RECENT POWER FAILURE, THEREFORE WE WILL WAIT
595                                     ;SIXTY SECONDS FOR THE DRIVE TO COME READY
596
597 012150 012701 000074                        MOV     #60.,R1         ;SIXTY SECOND TIMEOUT
598 012154 012777 000200 167746                 MOV     #200,@RLCS      ;SET CRDY
599 012162 053777 002146 167740                 BIS     DRIVE,@RLCS     ;SET IN DRIVE SELECT
600 012170 032777 000001 167732 2$:             BIT     #DRDY,@RLCS     ;DRIVE READY???
601 012176 001023                               BNE     3$              ;YES, THEN START TEST
602
603 012200                                      WAITMS  #10.            ;WAIT A SECOND
    (3) 012200 012700 000012                    MOV     #10.,R0
    (3) 012204 104026                           EMT     C$WTM
604
605 012206 005301                               DEC     R1              ;SIXTY SECONDS GONE BY
606 012210 001367                               BNE     2$              ;NO, GO BACK
607
608 012212                                      PRINTB  #FRMT12         ;DROPPING DRIVE
    (7) 012212 012746 011225                    MOV     #FRMT12,-(SP)
    (6) 012216 012746 000001                    MOV     #1,-(SP)
    (3) 012222 010600                           MOV     SP,R0
    (4) 012224 104014                           EMT     C$PNTB
    (4) 012226 062706 000004                    ADD     #4,SP
609 012232 004737 010342                6$:     JSR     PC,LINE1        ;GIVE DRIVE INFO
610 012236                                      DODU    UNITST          ;TELL SUPERVISOR TO DROP IT
    (3) 012236 013700 002126                    MOV     UNITST,R0
    (3) 012242 104053                           EMT     C$DODU
```

```
                                                    DOCLN                                   ;FORCE AN ABORT
(3)  012244  104044                                 EMT     C$DCLN
                                                    ENDINIT
                                                    ENDMOD
                                                    CLNCODE
                                                    BGNCLN
                                                    SETPRI  #PRI07
(3)  012352  012700  000340                         MOV     #PRI07,R0
(3)  012356  104041                                 EMT     C$SPRI
                                                    TST     PWRFLG                          ;TREAT POWER FAILURE
                                                    BEQ     2$
                                                    DEC     PWRFLG
```

```
012246  012777  000013  167660  3$:   MOV     #13,@RLDA                       ;SETUP DR RST
012254  012777  000204  167646        MOV     #204,@RLCS                      ;GS FUNC
012262  053777  002146  167640        BIS     DRIVE,@RLCS                     ;SELECT DRIVE
012270  042777  000200  167632        BIC     #200,@RLCS                      ;ISSUE IT
012276  032777  000200  167624  4$:   BIT     #200,@RLCS                      ;WAIT FOR READY
012304  001774                        BEQ     4$

012306                        END:    SETVEC  BVEC,#INTSRV,#340
(7)  012306  012746  000340          MOV     #340,-(SP)
(6)  012312  012746  013552          MOV     #INTSRV,-(SP)
(5)  012316  013746  002144          MOV     BVEC,-(SP)
(4)  012322  012746  000003          MOV     #3,-(SP)
(3)  012326  104037                  EMT     C$SVEC
(2)  012330  062706  000010          ADD     #10,SP
012334  005037  002176                CLR     PFLG                    ;CLR PROCESSOR FLAG
012340                               READBUS                         ;Q-BUS
(3)  012340  104007                  EMT     C$RDBU
012342                               BNCOMPLETE      1$
(2)  012342  103002                  BCC     1$
012344  005237  002176                INC     PFLG                    ;NO, Q-BUS THEN
012350                        1$:
012350                               ENDINIT
(3)  012350                   L10012:
(3)  012350  104011                  EMT     C$INIT

012352                               ENDMOD

012352                        BGNMOD CLNCODE

012352                               BGNCLN

012352                               SETPRI  #PRI07
(3)  012352  012700  000340          MOV     #PRI07,R0
(3)  012356  104041                  EMT     C$SPRI

012360  032777  000200  167542  1$:   BIT     #CRDY,@RLCS
012366  001774                        BEQ     1$

012370  042777  000100  167532        BIC     #INTEN,@RLCS

012376                               CLRVEC  BVEC
(3)  012376  013700  002144          MOV     BVEC,R0
(3)  012402  104036                  EMT     C$CVEC

012404  005737  002122                TST     PWRFLG                  ;TREAT POWER FAILURE
012410  001402                        BEQ     2$

012412  005337  002122                DEC     PWRFLG

012416                        2$:
```

```
                                                                    H 4
 652  012416                          L10013:  ENDCLN
 (3)  012416
 (3)  012416  104012                          EMT     C$CLEAN
 653
 654  012420                                   ENDMOD
 655
 656
 657
 658  012420                          BGNMOD   DRPCODE
 659
 660  012420                                   BGNDU
 661
 662  012420  000240                           NOP
 663
 664  012422                                   ENDDU
 (3)  012422                          L10014:
 (3)  012422  104055                          EMT     C$DU
 665
 666  012424                          ENDMOD
 667
 668  012424                          BGNMOD   ADDCODE
 669
 670  012424                                   BGNAU
 671
 672  012424  000240                           NOP
 673
 674  012426                                   ENDAU
 (3)  012426                          L10015:
 (3)  012426  104054                          EMT     C$AU
 675
 676  012430                          ENDMOD
 677
 678
 679                                  .SBTTL  GLOBAL SUBROUTINES
 680
 681  012430                          BGNMOD  GLBSUB
 682
 683  012430                          CKERLT: INLOOP
 (3)  012430  104020                          EMT     C$INLP
 684  012432                                   BCOMPLETE       99$
 (2)  012432  103427                          BCS     99$
 685  012434  005737  011354                   TST     DROP
 686  012440  001424                           BEQ     99$
 687  012442  005277  170230                   INC     @ERPOINT
 688  012446  027737  170224  011356          CMP     @ERPOINT,MERLMT
 689  012454  002416                           BLT     99$
 690
 691  012456                                   PRINTF  #FRMT11
 (7)  012456  012746  011164                   MOV     #FRMT11,-(SP)
 (6)  012462  012746  000001                   MOV     #1,-(SP)
 (3)  012466  010600                           MOV     SP,R0
 (4)  012470  104017                           EMT     C$PNTF
 (4)  012472  062706  000004                   ADD     #4,SP
 692  012476  004737  010342                   JSR     PC,LINE1
 693  012502                                   DODU    UNITST          ;DROP THE UNIT
 (3)  012502  013700  002126                   MOV     UNITST,R0
```

```
(3)  012506  104053                      EMT     C$DODU
694  012510                              DOCLN
(3)  012510  104044                      EMT     C$DCLN
695  012512                          99$:
696  012512  000205                      RTS     R5
697
698
699
700                                  .SBTTL  ROUTINE TO CHECK FOR CONTROLLER ERRORS
701
702                                  ;*********************************************************************
703                                  ;*THIS ROUTINE WILL CHECK RLCS FOR ERRORS AND PRINT THEM
704                                  ;*ACCORDINGLY.  IT WILL MERGE THE ERROR PRINTOUT WITH THE TEST
705                                  ;*ERROR MESSAGE.
706                                  ;*
707                                  ;*EXAMPLE:   RLCS CONTAINED FOLLOWING ERROR(S):
708                                  ;*                 DRV   OPI   HCRC   HNF
709                                  ;*           SEEK UNDER INTERRUPT
710                                  ;*
711                                  ;*
712                                  ;*
713                                  ;*ROUTINE USES R0,R1 AND PICKS HEADER FROM R3
714                                  ;*
715                                  ;*      CALL    JSR     R5,CHERR
716                                  ;*
717                                  ;*
718                                  ;*
719
720  012514  005037  002160          CHERR:  CLR     DERFLG          ;CLEAR OUT DRIVE ERROR FLAG
721  012520  032737  176000  002162          BIT     #176000,E.CS    ;ANY ERRORS SET
722  012526  001001                          BNE     199$            ;IF YES, INVESTIGATE
723  012530  000205                          RTS     R5              ;NO, EXIT
724  012532  023727  002262  000004  199$:   CMP     TMPFNC,#GSTAT   ;FUNCTION-NOP, RESET, GETSTATUS
725  012540  002401                          BLT     98$             ;YES, GO CHECK IF ONLY DRIVE ERROR
726  012542  000414                          BR      1$              ;YES SERVICE ERROR
727  012544  023727  002262  000002  98$:   CMP     TMPFNC,#WRCHK
728  012552  001410                          BEQ     1$
729  012554  013700  002162                  MOV     E.CS,R0         ;GET E.CS
730  012560  042700  001777                  BIC     #1777,R0        ;
731  012564  022700  140000                  CMP     #140000,R0      ;DRIVE ERROR ALONE?
732  012570  001001                          BNE     1$              ;NO, GO SERVICE
733  012572  000205                  2$:     RTS     R5              ;YES, EXIT
734
735  012574  012701  007625          1$:     MOV     #EM102,R1       ;GET START OF STRING
736  012600  005737  002162                  TST     E.CS            ;IS COMPOSITE ERROR SET?(BETTER BE)
737  012604  100003                          BPL     99$             ;IT'S NOT SOMETHING IS WRONG
738  012606  004537  013260                  JSR     R5,FIX          ;YES, PUT "COMP" IN STRING
739  012612  003702                          COMP                    ;"COMP"
740  012614  032737  040000  002162  99$:   BIT     #DERR,E.CS      ;DRIVE ERROR SET?
741  012622  001405                          BEQ     3$              ;NO, CONTINUE
742  012624  005237  002160                  INC     DERFLG          ;SET DRV ERROR FLAG
743  012630  004537  013260                  JSR     R5,FIX          ;YES, PUT "DRV" INTO STRING
744  012634  003631                          DEMES                   ;"DRV"
745  012636  032737  020000  002162  3$:    BIT     #NXM,E.CS       ;NON-EXISTENT MEMORY ERROR?
746  012644  001403                          BEQ     4$              ;NO, CONTINUE
747  012646  004537  013260                  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'
                                                BIT     #BIT11,E.CS     ;HEADERCRC ERROR?
                                                BEQ     5$              ;NO, GO CHECK HEADER NOT FOUND
                                                JSR     R5,FIX          ;GO PUT 'HCRC' IN STRING
                                                HCRCMES                 ;'HCRC'
                                        5$:     BIT     #BIT12,E.CS     ;HEADER NOT FOUND?
                                                BEQ     8$              ;NO, GO PUT 'CRLF' IN STRING
                                                JSR     R5,FIX          ;PUT 'HNF' IN STRING
                                                HNFMES                  ;'HNF'
                                                BR      8$              ;PUT 'CRLF' IN STRING
                                        6$:     BIT     #BIT11,E.CS     ;DATA CRC ERROR?
                                                BEQ     7$              ;NO, GO CHECK DATA LATE
                                                JSR     R5,FIX          ;PUT 'DCK' IN STRING
                                                DCKMES                  ;'DCK'
                                        7$:     BIT     #BIT12,E.CS     ;DATA LATE ERROR?
                                                BEQ     8$              ;NO, GO PUT IN 'CRLF'
                                                JSR     R5,FIX          ;PUT 'DLT' IN STRING
                                                DLTMES                  ;'DLT'
                                        8$:     JSR     R5,FIX
                                                MSCRLF
                                                JSR     R5,FIX
                                        RESTMS: .WORD   0               ;HEADER FROM TEST
                                                CLRB    (R1)            ;PUT TERMINATOR IN

                                                ERRDF   300.,LF,ERR6
(3)  013002  104462                             TRAP    T$ERCODE
(5)  013004  000454                             .WORD   300
(5)  013006  003700                             .WORD   LF
(5)  013010  010242                             .WORD   ERR6

013012  000205                                  RTS     R5              ;EXIT ROUTINE

                                .SBTTL LOAD RLCS
                                ;*****************************************************************
                                ;* ROUTINE TO LOAD RLCS WITH FUNCTION TO BE PERFORMED
                                ;*      CALL:   JSR     R5,LDFUNC
                                ;*              .WORD           ;BITS TO BE LOADED, FUNCTION
                                ;*                              ;AND INTR ENABLE ONLY
                                ;*
                                ;

013014  012537  002204          LDFUNC: MOV     (R5)+,LDCSR     ;GET BITS TO LOAD
013020  005737  002160                  TST     DERFLG
013024  001424                          BEQ     98$
013026  013746  002150                  MOV     B.CS,-(SP)
013032  012777  000013  167074          MOV     #13,@RLDA
013040  012737  000004  002150          MOV     #GSTAT,B.CS
013046  053737  002146  002150          BIS     DRIVE,B.CS
013054  013777  002150  167046          MOV     B.CS,@RLCS
013062  012637  002150                  MOV     (SP)+,B.CS
013066  032777  000200  167034  99$:    BIT     #200,@RLCS
013074  001774                          BEQ     99$
```

```
                                                    K 4
800  013076  010346                98$:    MOV     R3,-(SP)        ;SAVE R3
801  013100  042737  177661  002204        BIC     #177661,LDCSR   ;CLEAR ALL BUT FUNC & INTR EN
802  013106  013737  002204  013232        MOV     LDCSR,FNDFNC    ;SAVE FUNCTION
803  013114  042737  000100  013232        BIC     #INTEN,FNDFNC   ;ONLY FUNCTION
804  013122  013737  013232  002262        MOV     FNDFNC,TMPFNC
805  013130  012703  013234                MOV     #HDRLST,R3      ;GET HEADER LIST
806  013134  006237  013232                ASR     FNDFNC          ;ALIGN TO RIGHT
807  013140  001404                        BEQ     2$
808  013142  022323                1$:     CMP     (R3)+,(R3)+     ;BUMP R3 BY 4
809  013144  005337  013232                DEC     FNDFNC          ;FOUND IT
810  013150  001374                        BNE     1$              ;NO,KEEP LOOKING
811  013152  032737  000100  002204  2$:   BIT     #INTEN,LDCSR    ;YES,DO WE WANT FLAG OR INTR
812  013160  001401                        BEQ     3$              ;FLAG BRANCH
813  013162  005723                        TST     (R3)+           ;INTR POINT TO THAT ONE
814  013164  011303                3$:     MOV     (R3),R3         ;SET HEADER
815  013166  010337  012776                MOV     R3,RESTMS       ;SET UP HEADER
816  013172  053737  002146  002204        BIS     DRIVE,LDCSR     ;SELECT DRIVE
817  013200  052737  000200  002204  4$:   BIS     #200,LDCSR      ;CONTROLLER READY
818  013206  013777  002204  166714        MOV     LDCSR,@RLCS
819  013214  004537  013272                JSR     R5,BEFORE
820  013220  042777  000200  166702  5$:   BIC     #200,@RLCS
821  013226  012603                        MOV     (SP)+,R3        ;RESTORE R3
822  013230  000205                        RTS     R5              ;EXIT
823
824  013232  000000                FNDFNC: .WORD   0
825
826  013234  003763                HDRLST: NOPMES
827  013236  004014                        NOPINT
828  013240  004046                        WCKMES
829  013242  004106                        WCKINT
830  013244  004333                OKHDR:  GSTMES
831  013246  004372                        GSTINT
832  013250  004250                        SEKMES
833  013252  004301                        SEKINT
834  013254  004147                        RHDMES
835  013256  004207                        RHDINT
836
837                                ;*****************************************************************
838                                ;*ROUTINE TO MOVE ASCII STRINGS
839                                ;*USES REGISTERS R1 - WHERE STRING IS BEING BUILT
840                                ;*
841                                ;*      CALL    JSR     R5,FIX
842                                ;*              .WORD           ;ADDRESS OF STRING TO MOVE
843
844  013260  012500                FIX:    MOV     (R5)+,R0        ;GET ADDRESS AND MOVE RETURN
845  013262  112021                1$:     MOVB    (R0)+,(R1)+     ;GET BYTE AND UPDATE
846  013264  001376                        BNE     1$              ;WATCH 0 BYTE TERMINATOR
847  013266  105741                        TSTB    -(R1)           ;BACK UP OVER ZERO BYTE
848  013270  000205                        RTS     R5              ;EXIT
849
850
851                                ;LOAD REGISTERS BEFORE FUNCTION
852                                ;CALL:  JSR     R5,BEFORE
853
854  013272  017737  166632  002150  BEFORE: MOV     @RLCS,B.CS      ;READ CS
855  013300  017737  166626  002152          MOV     @RLBA,B.BA      ;READ BA
```

```
                                                                L 4
856  013306  017737  166622  002154          MOV     @RLDA,B.DA      ;READ DA
857  013314  017737  166616  002156          MOV     @RLMP,B.MP      ;READ MP
858  013322  000205                          RTS     R5
859
860
861                                  ;LOAD REGISTERS AT ERROR
862                                  ;CALL:  JSR     R5,AFTER
863
864  013324  017737  166600  002162 AFTER:   MOV     @RLCS,E.CS      ;READ CS
865  013332  017737  166574  002164          MOV     @RLBA,E.BA      ;READ BA
866  013340  017737  166570  002166          MOV     @RLDA,E.DA      ;READ DA
867  013346  017737  166564  002170          MOV     @RLMP,E.MP      ;READ MP
868  013354  017737  166556  002172          MOV     @RLMP,E.MP1     ;READ MP
869  013362  017737  166550  002174          MOV     @RLMP,E.MP2     ;READ MP
870  013370  000205                          RTS     R5
871
872
873
874                                  .SBTTL  ROUTINE TO CALCULATE CRC
875
876                                  ;ROUTINE WILL CALCULATE A CRC-16 CRC ON A WORD OF
877                                  ;1-16 BITS IN LENGTH, RESULT IS RETURNED IN ''CALBCC''
878                                  ;
879                                  ;       CALL:   JSR     R5,SIMBCC
880                                  ;               .WORD           ;NUMBER OF BITS (1-16)
881                                  ;               .WORD           ;DATA FOR CRC CALCULATION
882                                  ;               .WORD           ;PREVIOUS OR STARTING CRC
883                                  ;                               (SHOULD BE ZEROED FOR START)
884                                  ;       ROUTINE USES R0,R1,R2
885
886  013372  010046          .       SIMBCC: MOV     R0,-(SP)        ;SAVE R0
887  013374  010146                          MOV     R1,-(SP)        ;SAVE R1
888  013376  010246                          MOV     R2,-(SP)        ;SAVE R2
889  013400  012537  002220                  MOV     (R5)+,TEMP2     ;GET NUMBER OF BITS
890  013404  012537  002222                  MOV     (R5)+,TEMP3     ;GET DATA FOR CRC CALCULATION
891  013410  012537  002224                  MOV     (R5)+,TEMP4     ;GET STARTING CRC
892  013414  005037  002214          1$:     CLR     BCCFBK          ;
893  013420  013700  002224                  MOV     TEMP4,R0        ;GET PRESENT CRC
894  013424  006037  002222                  ROR     TEMP3           ;ROTATE NEW DATA
895  013430  005500                          ADC     R0              ;MERGE NEW WITH OLD
896  013432  032700  000001                  BIT     #1,R0           ;BIT 0 SET
897  013436  001402                          BEQ     2$              ;IF NOT CONTINUE
898  013440  005137  002214                  COM     BCCFBK          ;
899  013444  013700  002210          2$:     MOV     XPOLY,R0        ;GET CRC POLYNOMIAL (CRC-16)
900  013450  005100                          COM     R0              ;COMPLIMENT POLYNOMIAL
901  013452  040037  002214                  BIC     R0,BCCFBK
902  013456  000241                          CLC                     ;CLEAR CARRY
903  013460  006037  002224                  ROR     TEMP4
904  013464  013700  002214                  MOV     BCCFBK,R0
905  013470  013701  002224                  MOV     TEMP4,R1
906  013474  010102                          MOV     R1,R2
907  013476  040100                          BIC     R1,R0
908  013500  043702  002214                  BIC     BCCFBK,R2
909  013504  050200                          BIS     R2,R0
910  013506  043737  002210  002224        BIC     XPOLY,TEMP4
911  013514  050037  002224                  BIS     R0,TEMP4
```

```
   912  013520  005337  002220                DEC     TEMP2
   913  013524  001333                        BNE     1$
   914  013526  013737  002224  002216        MOV     TEMP4,CALBCC
   915  013534  012602                        MOV     (SP)+,R2
   916  013536  012601                        MOV     (SP)+,R1
   917  013540  012600                        MOV     (SP)+,R0
   918  013542  000205                        RTS     R5              ;RETURN
   919
   920
   921
   922                              ;ROUTINE TO SET FLAG IF TRAP OCCURRED
   923                              ;''TRPHAN'' IS IN LOCATION 4.
   924
   925
   926  013544  005237  002200      TRPHAN: INC     TRPFLG          ;INDICATE TRAP
   927  013550  000002                      RTI                     ;RETURN
   928
   929  013552                      BGNSRV  .
   930
   931  013552  005237  002202      INTSRV: INC     INTFLG          ;INDICATE INTERRUPT
   932
   933  013556                      ENDSRV
   (3)  013556                      L10016:
   (2)  013556  000002                      RTI
   934
   935                              ;ROUTINE TO WAIT FOR DRIVE READY
   936  013560  010146              WTDRDY: MOV     R1,-(SP)        ;SAVE R1
   937  013562  012701  003720              MOV     #2000.,R1       ;TIME OUT OF 200 MILLISECONDS
   938  013566  032777  000001  166334  1$: BIT     #DRDY,@RLCS     ;DRIVE READY?
   939  013574  001011                      BNE     2$              ;YES, EXIT
   940
   941  013576                              WAITUS  #1              ;WAIT A WHILE
   (3)  013576  012700  000001              MOV     #1,R0
   (3)  013602  104027                      EMT     C$WTU
   942  013604  005301                      DEC     R1              ;CHECK IF TIME UP
   943  013606  001367                      BNE     1$              ;NO, GO CHECK DRIVE READY
   944
   945  013610                              ERRDF   200.,DRTIM,ERR5 ;DRIVE READY DID NOT SET
   (3)  013610  104462                      TRAP    T$ERCODE
   (5)  013612  000310                      .WORD   200
   (5)  013614  004547                      .WORD   DRTIM
   (5)  013616  010230                      .WORD   ERR5
   946
   947  013620  012601              2$:     MOV     (SP)+,R1        ;RESTORE
   948  013622  000205                      RTS     R5              ;EXIT
   949
   950                              ;ROUTINE TO WAIT FOR CONTROLLER READY
   951  013624  010146              WTCRDY: MOV     R1,-(SP)        ;SAVE R1
   952  013626  012701  017500              MOV     #8000.,R1       ;WAIT 800 MILLISECONDS
   953  013632  032777  000200  166270  1$: BIT     #CRDY,@RLCS     ;CONTROLLER READY
   954  013640  001014                      BNE     2$              ;YES, EXIT
   955  013642                              WAITUS  #1              ;WAIT A WHILE
   (3)  013642  012700  000001              MOV     #1,R0
   (3)  013646  104027                      EMT     C$WTU
   956  013650  005301                      DEC     R1              ;CHECK IF TIME UP
   957  013652  001367                      BNE     1$              ;NO GO BACK
```

```
                                                ;GET REGISTERS
013654  004537  013324          JSR     R5,AFTER
                                ERRDF   100.,CRTIM,ERR6 ;CONTROLLER TIMED OUT
(3) 013660  104462              TRAP    T$ERCODE
(5) 013662  000144              .WORD   100
(5) 013664  004522              .WORD   CRTIM
(5) 013666  010242              .WORD   ERR6
013670  000402                  BR      3$              ;EXIT
013672  004537  013324  2$:     JSR     R5,AFTER        ;GET REGISTERS
013676  012601          3$:     MOV     (SP)+,R1
013700  000205                  RTS     R5              ;EXIT
013702                          ENDMOD
                        .SBTTL  **TEST 1** - RLCS ADDRESSABILITY
013702                  BGNTST                          ;****START OF TEST****
013702                  STARS
(2)                     ;;**************************************************************
                        ;TEST TO SEE IF WE CAN ADDRESS THE CONTROL
                        ;AND STATUS REGISTER.  IF WE TRAP WE WILL REPORT
                        ;THE ERROR AND ABORT.  AFTER THIS TEST WE ONLY KNOW
                        ;THAT WE CAN ADDRESS THE REGISTER.
013702                  STARS
(2)                     ;;**************************************************************
013702  005037  002200  1$:     CLR     TRPFLG          ;CLEAR TRAP OCCURANCE
013706          2$:     SETVEC  ERRVEC,#TRPHAN,#340  ;SET TO CATCH TRAP
(7) 013706  012746  000340      MOV     #340,-(SP)
(6) 013712  012746  013544      MOV     #TRPHAN,-(SP)
(5) 013716  013746  002212      MOV     ERRVEC,-(SP)
(4) 013722  012746  000003      MOV     #3,-(SP)
(3) 013726  104037              EMT     C$SVEC
(2) 013730  062706  000010      ADD     #10,SP
013734  005777  166170          TST     @RLCS           ;ADDRESS RLCS
013740                          CLRVEC  ERRVEC          ;RELEASE TRAP VECTOR
(3) 013740  013700  002212      MOV     ERRVEC,R0
(3) 013744  104036              EMT     C$CVEC
013746  005737  002200          TST     TRPFLG          ;TRAP OCCURRED???
013752  001407                  BEQ     3$              ;NO, IKAY PROCEED
013754  013737  002130  002234  MOV     RLCS,GDDAT      ;SET UP ERROR DATA
013762                          ERRSF   0.,EM1,ERR1     ;BUS TIMEOUT IN ADDRESSING RLCS
(3) 013762  104461              TRAP    T$ERCODE
(5) 013764  000000              .WORD   0
(5) 013766  004575              .WORD   EM1
(5) 013770  010034              .WORD   ERR1
```

```
  996  013772
  (3)  013772  104006                    3$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
                                                  EMT     C$CLP1
  997  013774                            ENDTST                          ;****END OF TEST****
  (3)  013774                            L10017:
  (3)  013774  104001                             EMT     C$ETST
  998
  999
 1000                                    .SBTTL  **TEST 2** - RLBA ADDRESSABILITY
 1001
 1002  013776                            BGNTST                          ;****START OF TEST****
 1003
 1004
 1005  013776                            STARS
  (2)                                    ;;**************************************************************
 1006                                    ;TEST TO SEE IF WE CAN ADDRESS THE BUS ADDRESS
 1007                                    ;REGISTER.  IF WE TRAP WE WILL REPORT THE ERROR
 1008                                    ;AND ABORT.  AFTER THIS TEST WE ONLY KNOW THAT
 1009                                    ;WE CAN ADDRESS THE REGISTER.
 1010  013776                            STARS
  (2)                                    ;;**************************************************************
 1011
 1012  013776  005037  002200            1$:     CLR     TRPFLG          ;CLEAR TRAP OCCURANCE
 1013  014002                            2$:     SETVEC  ERRVEC,#TRPHAN,#340  ;SET TO CATCH TRAP
  (7)  014002  012746  000340                     MOV     #340,-(SP)
  (6)  014006  012746  013544                     MOV     #TRPHAN,-(SP)
  (5)  014012  013746  002212                     MOV     ERRVEC,-(SP)
  (4)  014016  012746  000003                     MOV     #3,-(SP)
  (3)  014022  104037                             EMT     C$SVEC
  (2)  014024  062706  000010                     ADD     #10,SP
 1014
 1015  014030  005777  166076                     TST     @RLBA           ;ADDRESS RLBA
 1016  014034                                     CLRVEC  ERRVEC          ;RELEASE TRAP VECTOR
  (3)  014034  013700  002212                     MOV     ERRVEC,R0
  (3)  014040  104036                             EMT     C$CVEC
 1017  014042  005737  002200                     TST     TRPFLG          ;TRAP OCCURRED???
 1018  014046  001407                             BEQ     3$              ;NO, CONTINUE
 1019  014050  013737  002132  002234             MOV     RLBA,GDDAT      ;SETUP ERROR DATA
 1020
 1021  014056                                     ERRSF   1.,EM2,ERR1     ;BUS TIMEOUT IN ADDRESSING RLBA
  (3)  014056  104461                             TRAP    T$ERCODE
  (5)  014060  000001                             .WORD   1
  (5)  014062  004622                             .WORD   EM2
  (5)  014064  010034                             .WORD   ERR1
 1022  014066                            3$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  014066  104006                             EMT     C$CLP1
 1023  014070                            ENDTST                          ;****END OF TEST****
  (3)  014070                            L10020:
  (3)  014070  104001                             EMT     C$ETST
 1024
 1025
 1026                                    .SBTTL  **TEST 3** - RLDA ADDRESSABILITY
 1027
 1028  014072                            BGNTST                          ;****START OF TEST****
 1029  014072                            STARS
  (2)                                    ;;**************************************************************
 1030                                    ;TEST TO SEE IF WE CAN ADDRESS THE DISK ADDRESS
```

```
                                ;REGISTER IF WE TRAP WE WILL REPORT THE ERROR
                                ;AND ABORT.  AFTER THIS TEST WE ONLY KNOW THAT
                                ;WE CAN ADDRESS THE REGISTER.
014072                          STARS
 (2)                            ;;*****************************************************************
                                1$:     CLR     TRPFLG          ;CLEAR TRAP OCCURANCE
014076                          2$:     SETVEC  ERRVEC,#TRPHAN,#340  ;SET TO CATCH TRAP
 (7)  014076  012746  000340            MOV     #340,-(SP)
 (6)  014102  012746  013544            MOV     #TRPHAN,-(SP)
 (5)  014106  013746  002212            MOV     ERRVEC,-(SP)
 (4)  014112  012746  000003            MOV     #3,-(SP)
 (3)  014116  104037                    EMT     C$SVEC
 (2)  014120  062706  000010            ADD     #10,SP

 014124  005777  166004                 TST     @RLDA           ;ADDRESS RLDA
 014130                                 CLRVEC  ERRVEC          ;RELEASE TRAP VECTOR
 (3)  014130  013700  002212            MOV     ERRVEC,R0
 (3)  014134  104036                    EMT     C$CVEC
 014136  005737  002200                 TST     TRPFLG          ;TRAP OCCURRED???
 014142  001407                         BEQ     3$              ;NO, CONTINUE

 014144  013737  002134  002234         MOV     RLDA,GDDAT      ;SETUP ERROR INFO
 014152                                 ERRSF   2.,EM3,ERR1     ;BUS TIMEOUT IN ADDRESSING RLDA
 (3)  014152  104461                    TRAP    T$ERCODE
 (5)  014154  000002                    .WORD   2
 (5)  014156  004647                    .WORD   EM3
 (5)  014160  010034                    .WORD   ERR1
 014162                         3$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  014162  104006                    EMT     C$CLP1
 014164                         ENDTST                          ;****END OF TEST****
 (3)  014164                    L10021:
 (3)  014164  104001                    EMT     C$ETST


                                .SBTTL  **TEST 4** - RLMP ADDRESSABILITY

 014166                         BGNTST                          ;****START OF TEST****
 014166                         STARS
 (2)                            ;;*****************************************************************
                                ;TEST TO SEE IF WE CAN ADDRESS THE MULTIPURPOSE
                                ;REGISTER.  IF WE TRAP WE WILL REPORT THE ERROR AND
                                ;ABORT.  AFTER THIS TEST WE ONLY KNOW THAT WE CAN
                                ;ADDRESS THE REGISTER.
 014166                         STARS
 (2)                            ;;*****************************************************************

 014166  005037  002200         1$:     CLR     TRPFLG          ;CLEAR TRAP OCCURANCE
 014172                         2$:     SETVEC  ERRVEC,#TRPHAN,#340  ;SET UP TO CATCH TRAP
 (7)  014172  012746  000340            MOV     #340,-(SP)
 (6)  014176  012746  013544            MOV     #TRPHAN,-(SP)
 (5)  014202  013746  002212            MOV     ERRVEC,-(SP)
 (4)  014206  012746  000003            MOV     #3,-(SP)
 (3)  014212  104037                    EMT     C$SVEC
```

```
  (2)  014214  062706  000010                    ADD     #10,SP
1064
1065   014220  005777  165712                    TST     @RLMP           ;ADDRESS RLMP
1066   014224                                    CLRVEC  ERRVEC          ;RELEASE TRAP VECTOR
  (3)  014224  013700  002212                    MOV     ERRVEC,R0
  (3)  014230  104036                            EMT     C$CVEC
1067   014232  005737  002200                    TST     TRPFLG          ;TRAP OCCURRED???
1068   014236  001407                            BEQ     3$              ;NO, CONTINUE
1069   014240  013737  002136  002234            MOV     RLMP,GDDAT      ;SET UP ERROR INFO
1070
1071   014246                                    ERRSF   3.,EM4,ERR1     ;BUS TIMEOUT IN ADDRESSING RLMP
  (3)  014246  104461                            TRAP    T$ERCODE
  (5)  014250  000003                            .WORD   3
  (5)  014252  004674                            .WORD   EM4
  (5)  014254  010034                            .WORD   ERR1
1072   014256                          3$:       CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  014256  104006                            EMT     C$CLP1
1073   014260                          ENDTST                          ;****END OF TEST****
  (3)  014260                          L10022:
  (3)  014260  104001                            EMT     C$ETST
1074
1075
1076                                    .SBTTL  **TEST 5** - READ WRITE OF RLCS
1077
1078   014262                          BGNTST                          ;****START OF TEST****
1079
1080
1081
1082   014262                          STARS
  (2)                                   ;;**********************************************************************
1083                                    ;TEST THAT WE CAN WRITE/READ BITS 8,9 AND BITS 6-1
1084                                    ;OF THE CONTROL AND STATUS REGISTER.  BITS 15-10 AND 0
1085                                    ;ARE DON'T CARE BITS AT THIS TIME AND BIT 7
1086                                    ;(CONTROLLER READY) IS ALWAYS WRITTEN TO A ONE.
1087   014262                          STARS
  (2)                                   ;;**********************************************************************
1088
1089
1090   014262  012703  002576                    MOV     #CSPAT,R3       ;SET UP TABLE POINTER OF PATTERNS
1091
1092   014266                          BGNSEG                          ;****START OF SEGMENT****
  (3)  014266  104004                            EMT     C$BSEG
1093
1094   014270                          CSTEST:
1095   014270  011337  002234                    MOV     (R3),GDDAT      ;GET PATTERN INTO GDDAT
1096   014274  052737  000200  002234            BIS     #200,GDDAT      ;INSURE GO IS SET
1097   014302  013777  002234  165620            MOV     GDDAT,@RLCS     ;LOAD RLCS (CONTROL AND STATUS)
1098   014310  032777  040000  165612            BIT     #DERR,@RLCS     ;IF DRIVE ERROR PRESENT
1099   014316  001403                            BEQ     99$             ;THEN EXPECT DRIVE AND
1100   014320  052737  140000  002234            BIS     #ERR!DERR,GDDAT ;COMPOSITE ERROR
1101   014326  017737  165576  002236  99$:      MOV     @RLCS,BDDAT     ;READ RLCS BACK
1102   014334  042737  000001  002236            BIC     #DRDY,BDDAT     ;IGNORE DRIVE READY
1103   014342  023737  002234  002236            CMP     GDDAT,BDDAT     ;DID WE READ WHAT WE LOADED
1104   014350  001404                            BEQ     1$              ;YES, THEN BRANCH
1105
1106   014352                                    ERRDF   4.,EM5,ERR2     ;WRONG DATA IN RLCS
```

```
  (3)  014352  104462                    TRAP    T$ERCODE
  (5)  014354  000004                    .WORD   4
  (5)  014356  004721                    .WORD   EM5
  (5)  014360  010046                    .WORD   ERR2
1107   014362                    1$:     ESCAPE  SEG             ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
  (3)  014362  104010                    EMT     C$ESCAPE
  (3)  014364  000012                    .WORD   10000$-.
1108
1109
1110   014366  005723                    TST     (R3)+           ;BUMP FOR NEXT PATTERN
1111   014370  020327  002674            CMP     R3,#CSEND       ;CHECK FOR END
1112   014374  001335                    BNE     CSTEST          ;NOT END, LOAD NEXT PATTERN
1113
1114   014376                    ENDSEG                          ;****END OF SEGMENT****
  (3)  014376                    10000$:
  (3)  014376  104005                    EMT     C$ESEG
1115   014400                    ENDTST                          ;****END OF TEST****
  (3)  014400                    L10023:
  (3)  014400  104001                    EMT     C$ETST
1116
1117
1118                             .SBTTL  **TEST 6** - READ WRITE OF RLBA
1119
1120   014402                    BGNTST                          ;****START OF TEST****
1121
1122   014402                    STARS
  (2)                            ;;*****************************************************************
1123                             ;TEST THAT WE CAN WRITE/READ BITS IS THRU 1 OF THE
1124                             ;BUS ADDRESS REGISTER.  FOUR PATTERNS ARE USED: GROWING 1, SHIFTING 1,
1125                             ;GROWING 0 AND SHIFTING 0.  BIT 0 IS ALSO LOADED BUT
1126                             ;SHOULD ALWAYS COME BACK AS 0
1127   014402                    STARS
  (2)                            ;;*****************************************************************
1128
1129
1130   014402  012703  002264            MOV     #BEGPAT,R3      ;GET START OF PATTERN LIST
1131   014406                    BGNSEG                          ;****START OF SEGMENT****
  (3)  014406  104004                    EMT     C$BSEG
1132   014410                    BATEST:
1133   014410  011337  002234            MOV     (R3),GDDAT      ;GET PATTERN TO SEND
1134   014414  005737  002260            TST     T.CNTLR         ;RL11??
1135   014420  001403                    BEQ     2$              ;NO,
1136   014422  042737  000001  002234    BIC     #BIT0,GDDAT     ;KEEP RLBA EVEN (UNIBUS)
1137   014430  013777  002234  165474    2$:     MOV     GDDAT,@RLBA     ;LOAD PATTERN TO BUS ADDRESS
1138   014436  017737  165470  002236    MOV     @RLBA,BDDAT     ;READ IT BACK
1139   014444  023737  002234  002236    CMP     GDDAT,BDDAT     ;IS IT CORRECT?
1140   014452  001404                    BEQ     1$              ;IF SO, BRANCH
1141
1142   014454                            ERRDF   5.,EM6,ERR2     ;DATA WRONG IN RLBA
  (3)  014454  104462                    TRAP    T$ERCODE
  (5)  014456  000005                    .WORD   5
  (5)  014460  004772                    .WORD   EM6
  (5)  014462  010046                    .WORD   ERR2
1143   014464                    1$:     ESCAPE  SEG             ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
  (3)  014464  104010                    EMT     C$ESCAPE
  (3)  014466  000012                    .WORD   10000$-.
```

```
1144
1145
1146  014470  005723                          TST     (R3)+           ;BUMP FOR NEXT PATTERN
1147  014472  020327  002472                  CMP     R3,#ENDPAT      ;CHECK FOR END
1148  014476  001344                          BNE     BATEST          ;NOT END, BRANCH FOR NEXT
1149
1150  014500                        ENDSEG                            ;****END OF SEGMENT****
 (3)  014500                        10000$:
 (3)  014500  104005                          EMT     C$ESEG
1151  014502                        ENDTST                            ;****END OF TEST****
 (3)  014502                        L10024:
 (3)  014502  104001                          EMT     C$ETST
1152
1153
1154                               .SBTTL  **TEST 7** - READ WRITE OF RLDA
1155
1156  014504                       BGNTST                            ;****START OF TEST****
1157
1158  014504                       STARS
 (2)                               ;;*****************************************************************
1159                               ;TEST THAT WE CAN WRITE/READ THE DISK ADDRESS REGISTER
1160                               ;ALL BIT POSITIONS ARE WRITTEN USING FOUR PATTERNS:
1161                               ;GROWING 1, SHIFTING 1, GROWING 0 AND SHIFTING 0
1162  014504                       STARS
 (2)                               ;;*****************************************************************
1163
1164
1165  014504  012703  002264                  MOV     #BEGPAT,R3      ;SET UP POINTER TO PATTERN LIST
1166  014510                       BGNSEG                            ;****START OF SEGMENT****
 (3)  014510  104004                          EMT     C$BSEG
1167  014512                       DATEST:
1168  014512  011337  002234                  MOV     (R3),GDDAT      ;GET PATTERN
1169  014516  013777  002234  165410          MOV     GDDAT,@RLDA     ;LOAD PATTERN IN DA
1170
1171  014524  017737  165404  002236          MOV     @RLDA,BDDAT     ;READ PATTERN BACK
1172  014532  023737  002234  002236          CMP     GDDAT,BDDAT     ;IS IT CORRECT?
1173  014540  001404                          BEQ     1$              ;BRANCH IF CORRECT
1174
1175  014542                                  ERRDF   6.,EM7,ERR2     ;WRONG DATA IN RLDA
 (3)  014542  104462                          TRAP    T$ERCODE
 (5)  014544  000006                          .WORD   6
 (5)  014546  005020                          .WORD   EM7
 (5)  014550  010046                          .WORD   ERR2
1176  014552                       1$:        ESCAPE  SEG             ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
 (3)  014552  104010                          EMT     C$ESCAPE
 (3)  014554  000012                          .WORD   10000$-.
1177
1178
1179  014556  005723                          TST     (R3)+           ;BUMP POINTER
1180  014560  020327  002472                  CMP     R3,#ENDPAT      ;AT END OF PATTERNS?
1181  014564  001352                          BNE     DATEST          ;NO, BRANCH BACK
1182
1183  014566                        ENDSEG                            ;****END OF SEGMENT****
 (3)  014566                        10000$:
 (3)  014566  104005                          EMT     C$ESEG
1184  014570                        ENDTST                            ;****END OF TEST****
```

```
  (3)  014570                          L10025:
  (3)  014570  104001                          EMT     C$ETST
1185
1186
1187                                   .SBTTL  **TEST 8** - BIS OF RLCS
1188
1189  014572                           BGNTST                          ;****START OF TEST****
1190  014572                           STARS
  (2)                                  ;;*****************************************************************
1191                                   ;TEST THAT WE CAN USE THE 'BIS'' INSTRUCTION ON THE CONTROL
1192                                   ;AND STATUS REGISTER.  BITS 8,9 AND 6-1 ARE TESTED TO
1193                                   ;SET INDIVIDUALLY AS WELL AS COLLECTIVELY WITHOUT DESTROYING
1194                                   ;ANY PREVIOUS DATA PATTERN
1195  014572                           STARS
  (2)                                  ;;*****************************************************************
1196
1197
1198  014572  012703  002576                   MOV     #CSPAT,R3       ;GET BEGINNING OF LIST
1199  014576                           BGNSEG                          ;****START OF SEGMENT****
  (3)  014576  104004                        EMT     C$BSEG
1200  014600                           1$:
1201  014600  012777  000200  165322           MOV     #CRDY,@RLCS     ;INSURE GO IS THERE
1202  014606  011337  002234                   MOV     (R3),GDDAT      ;SET UP EXPECTED RLCS
1203  014612  052737  000200  002234           BIS     #CRDY,GDDAT     ;IN GDDAT
1204  014620  051377  165304                   BIS     (R3),@RLCS      ;BIT SET PATTERN IN RLCS
1205  014624  032777  040000  165276           BIT     #DERR,@RLCS     ;IF ERROR BIT SET THEN
1206  014632  001403                           BEQ     99$             ;EXPECT IT ON THE READ
1207  014634  052737  140000  002234           BIS     #ERR!DERR,GDDAT ;BACK
1208  014642  017737  165262  002236  99$:     MOV     @RLCS,BDDAT     ;READ RLCS TO CHECK 'BIS''
1209  014650  042737  000001  002236           BIC     #DRDY,BDDAT     ;CLEAR OUT DRIVE READY
1210  014656  023737  002236  002234           CMP     BDDAT,GDDAT     ;DID BIS WORK?
1211  014664  001404                           BEQ     2$              ;BRANCH IF OKAY
1212
1213  014666                                   ERRDF   7.,EM61,ERR2    ;WRONG DATA IN RLCS
  (3)  014666  104462                        TRAP    T$ERCODE
  (5)  014670  000007                        .WORD   7
  (5)  014672  006515                        .WORD   EM61
  (5)  014674  010046                        .WORD   ERR2
1214  014676                           2$:     ESCAPE  SEG             ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
  (3)  014676  104010                        EMT     C$ESCAPE
  (3)  014700  000012                        .WORD   10000$-.
1215                                                                   ;BIT OR CLEARED OTHER BIT
1216
1217  014702  005723                           TST     (R3)+           ;GET NEXT PATTERN
1218  014704  022703  002674                   CMP     #CSEND,R3       ;AT END OF LIST
1219  014710  001333                           BNE     1$              ;NO GO BACK FOR TEST OF
1220                                                                   ;NEXT PATTERN
1221  014712                           ENDSEG                          ;****END OF SEGMENT****
  (3)  014712                        10000$:
  (3)  014712  104005                        EMT     C$ESEG
1222  014714                           ENDTST                          ;****END OF TEST****
  (3)  014714                        L10026:
  (3)  014714  104001                        EMT     C$ETST
1223
1224
1225                                   .SBTTL  **TEST 9** - BIC OF RLCS
```

```
1226
1227  014716                          BGNTST                          ;****START OF TEST****
1228
1229  014716                          STARS
 (2)                                  ;;*****************************************************************
1230                                  ;TEST THAT THE 'BIC'' INSTRUCTION WILL WORK ON THE
1231                                  ;CONTROL AND STATUS REGISTER.  BITS 8-9 AND 6-1 ARE
1232                                  ;TESTED.
1233  014716                          STARS
 (2)                                  ;;*****************************************************************
1234
1235
1236  014716  012703  002576                  MOV     #CSPAT,R3       ;GET BEGINNING OF PATTERNS
1237  014722                          BGNSEG                          ;****START OF SEGMENT****
 (3)  014722  104004                          EMT     C$BSEG
1238  014724                          1$:
1239  014724  012777  001776  165176          MOV     #1776,@RLCS     ;SET ALL SETTABLE BITS
1240  014732  012737  001776  002234          MOV     #1776,GDDAT     ;SET UP EXPECT DATA IN
1241  014740  041337  002234                  BIC     (R3),GDDAT      ;GDDAT
1242  014744  041377  165160                  BIC     (R3),@RLCS      ;CLEAR BITS IN RLCS VIA 'BIC''
1243  014750  032777  040000  165152          BIT     #DERR,@RLCS     ;IF DRIVE ERROR BIT SET
1244  014756  001403                          BEQ     99$             ;EXPECT IT SET WHEN WE
1245  014760  052737  140000  002234          BIS     #ERR!DERR,GDDAT ;READ IT BACK
1246  014766  017737  165136  002236  99$:     MOV     @RLCS,BDDAT     ;MOVE RLCS TO BDDAT FOR COMPARE
1247  014774  042737  000001  002236          BIC     #DRDY,BDDAT     ;CLEAR DRIVE READY
1248  015002  023737  002236  002234          CMP     BDDAT,GDDAT     ;DID 'BIC'' WORK PROPERLY
1249  015010  001404                          BEQ     2$              ;BRANCH IF OKAY
1250
1251  015012                                  ERRDF   8.,EM62,ERR2    ;WRONG DATA IN RLCS
 (3)  015012  104462                          TRAP    T$ERCODE
 (5)  015014  000010                          .WORD   8
 (5)  015016  006576                          .WORD   EM62
 (5)  015020  010046                          .WORD   ERR2
1252  015022                          2$:     ESCAPE  SEG             ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
 (3)  015022  104010                          EMT     C$ESCAPE
 (3)  015024  000012                          .WORD   10000$-.
1253
1254  015026  005723                          TST     (R3)+           ;GET NEXT PATTERN
1255  015030  020327  002674                  CMP     R3,#CSEND       ;AT END OF LIST
1256  015034  001333                          BNE     1$              ;NO, GO BACK WITH NEXT PATTERN
1257  015036                          ENDSEG                          ;****END OF SEGMENT****
 (3)  015036                          10000$:
 (3)  015036  104005                          EMT     C$ESEG
1258  015040                          ENDTST                          ;****END OF TEST****
 (3)  015040                          L10027:
 (3)  015040  104001                          EMT     C$ETST
1259
1260
1261                                  .SBTTL  **TEST 10** - BIS OF RLBA
1262
1263  015042                          BGNTST                          ;****START OF TEST****
1264
1265  015042                          STARS
 (2)                                  ;;*****************************************************************
1266                                  ;TEST THAT THE 'BIS'' INSTRUCTION WILL WORK ON THE BUS
1267                                  ;ADDRESS REGISTER.  BITS 15-0 ARE LOADED, ONLY BITS 15-1
```

```
                                ;ARE EXPECTED BACK.  FOUR PATTERNS ARE USED: GROWING 1, SHIFTING 1,
                                ;GROWING 0, AND SHIFTING 0.
015042                          STARS
 (2)                            ;;*****************************************************************
                                
                                        MOV     #BEGPAT,R3      ;GET START OF LIST
015046                          BGNSEG                          ;****START OF SEGMENT****
 (3) 015046 104004                      EMT     C$BSEG
015050                          1$:
015050 005077 165056                    CLR     @RLBA           ;CLEAR 'BA''
015054 011337 002234                    MOV     (R3),GDDAT      ;SET EXPECTED
015060 005737 002260                    TST     T.CNTLR         ;RL11
015064 001403                           BEQ     3$              ;NO
015066 042737 000001 002234             BIC     #1,GDDAT        ;BIT 0 CAN'T SET IN RLBA (UNIBUS)
015074 051377 165032            3$:     BIS     (R3),@RLBA      ;BIS RLBA WITH PATTERN
015100 017737 165026 002236             MOV     @RLBA,BDDAT     ;READ 'BA''
015106 023737 002236 002234             CMP     BDDAT,GDDAT     ;DID RLBA LOAD PROPERLY?
015114 001404                           BEQ     2$              ;BRANCH IF YES

015116                                  ERRDF   9.,EM63,ERR2    ;WRONG DATA IN RLBA
 (3) 015116 104462                      TRAP    T$ERCODE
 (5) 015120 000011                      .WORD   9
 (5) 015122 006661                      .WORD   EM63
 (5) 015124 010046                      .WORD   ERR2
015126                          2$:     ESCAPE  SEG             ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
 (3) 015126 104010                      EMT     C$ESCAPE
 (3) 015130 000012                      .WORD   10000$-.

015132 005723                           TST     (R3)+           ;GET NEXT PATTERN
015134 020327 002472                    CMP     R3,#ENDPAT      ;DID WE COMPLETE LIST
015140 001343                           BNE     1$              ;NO, GO BACK FOR NEXT.
015142                          ENDSEG                          ;****END OF SEGMENT****
 (3) 015142                     10000$:
 (3) 015142 104005                      EMT     C$ESEG
015144                          ENDTST                          ;****END OF TEST****
 (3) 015144                     L10030:
 (3) 015144 104001                      EMT     C$ETST


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

015146                          BGNTST                          ;****START OF TEST****

015146                          STARS
 (2)                            ;;*****************************************************************
                                ;TEST THAT THE 'BIC'' INSTRUCTION WILL WORK ON THE BUS
                                ;ADDRESS REGISTER.  BITS 15-1 ARE TESTED WITH 4 PATTERNS
                                ;GROWING 1, SHIFTING 1, GROWING 0 AND SHIFTING 0.
015146                          STARS
 (2)                            ;;*****************************************************************


015146 012703 002264                    MOV     #BEGPAT,R3      ;GET START OF LIST
015152                          BGNSEG                          ;****START OF SEGMENT****
 (3) 015152 104004                      EMT     C$BSEG
```

```
1309  015154                                1$:
1310  015154  012777  177776  164750              MOV     #-2,@RLBA       ;SET RLBA TO ALL 1'S (BIT 0=0)
1311  015162  012737  177776  002234              MOV     #-2,GDDAT       ;SET UP EXPECTED RESULTS
1312  015170  041337  002234                    BIC     (R3),GDDAT      ;IN GDDAT
1313  015174  041377  164732                    BIC     (R3),@RLBA      ;BIC RLBA
1314  015200  017737  164726  002236            MOV     @RLBA,BDDAT     ;READ RLBA
1315  015206  023737  002236  002234            CMP     BDDAT,GDDAT     ;BIC WORK OKAY?
1316  015214  001404                            BEQ     2$              ;IF YES BRANCH
1317
1318  015216                                    ERRDF   10.,EM64,ERR2   ;WRONG DATA IN RLBA
  (3) 015216  104462                            TRAP    T$ERCODE
  (5) 015220  000012                            .WORD   10
  (5) 015222  006742                            .WORD   EM64
  (5) 015224  010046                            .WORD   ERR2
1319  015226                                2$: ESCAPE  SEG             ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
  (3) 015226  104010                            EMT     C$ESCAPE
  (3) 015230  000012                            .WORD   10000$-.
1320
1321  015232  005723                            TST     (R3)+           ;GET NEXT PATTERN
1322  015234  020327  002472                    CMP     R3,#ENDPAT      ;HAVE WE COMPLETED LIST
1323  015240  001345                            BNE     1$              ;NO, GO BACK FOR NEXT
1324  015242                                ENDSEG                      ;****END OF SEGMENT****
  (3) 015242                                10000$:
  (3) 015242  104005                            EMT     C$ESEG
1325  015244                                ENDTST                      ;****END OF TEST****
  (3) 015244                                L10031:
  (3) 015244  104001                            EMT     C$ETST
1326
1327
1328                                        .SBTTL  **TEST 12** - BIS OF RLDA
1329
1330  015246                                BGNTST                      ;****START OF TEST****
1331
1332  015246                                STARS
  (2)                                       ;;*****************************************************************
1333                                        ;TEST THAT THE 'BIS" INSTRUCTION WILL WORK ON THE DISK ADDRESS
1334                                        ;REGISTER.  BITS 15-0 ARE TESTED WITH 4 PATTERNS, GROWING 1,
1335                                        ;SHIFTING 1, GROWING 0, AND SHIFTING 0.
1336  015246                                STARS
  (2)                                       ;;*****************************************************************
1337
1338
1339  015246  012703  002264                    MOV     #BEGPAT,R3      ;GET START OF LIST
1340  015252                                BGNSEG                      ;****START OF SEGMENT****
  (3) 015252  104004                            EMT     C$BSEG
1341  015254                                1$:
1342  015254  005077  164654                    CLR     @RLDA           ;CLEAR 'DA''
1343  015260  011337  002234                    MOV     (R3),GDDAT      ;SET EXPECTED
1344  015264  051377  164644                    BIS     (R3),@RLDA      ;BIS RLDA
1345  015270  017737  164640  002236            MOV     @RLDA,BDDAT     ;READ RLDA
1346  015276  023737  002236  002234            CMP     BDDAT,GDDAT     ;IS RLDA CORRECT
1347  015304  001404                            BEQ     2$              ;IF OKAY BRANCH
1348
1349  015306                                    ERRDF   11.,EM65,ERR2   ;WRONG DATA IN RLDA
  (3) 015306  104462                            TRAP    T$ERCODE
  (5) 015310  000013                            .WORD   11
```

```
  (5)  015312  007025                        .WORD   EM65
  (5)  015314  010046                        .WORD   ERR2
1350   015316                        2$:     ESCAPE  SEG             ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
  (3)  015316  104010                        EMT     C$ESCAPE
  (3)  015320  000012                        .WORD   10000$-.
1351
1352   015322  005723                        TST     (R3)+           ;GET NEXT PATTERN
1353   015324  020327  002472                CMP     R3,#ENDPAT      ;HAVE WE FINISHED?
1354   015330  001351                        BNE     1$              ;NO GO BACK
1355   015332                        ENDSEG                          ;****END OF SEGMENT****
  (3)  015332                        10000$:
  (3)  015332  104005                        EMT     C$ESEG
1356   015334                        ENDTST                          ;****END OF TEST****
  (3)  015334                        L10032:
  (3)  015334  104001                        EMT     C$ETST
1357
1358
1359                                .SBTTL  **TEST 13** - BIC OF RLDA
1360
1361   015336                       BGNTST                          ;****START OF TEST****
1362
1363   015336                       STARS
  (2)                               ;;*****************************************************************
1364                                ;TEST THAT THE 'BIC'' INSTRUCTION WORKS ON THE DISK
1365                                ;ADDRESS REGISTER.  ALL BITS ARE TESTED WITH FOUR
1366                                ;PATTERNS: GROWING 1, SHIFTING 1, GROWING 0 AND SHIFTING 0
1367   015336                       STARS
  (2)                               ;;*****************************************************************
1368
1369
1370   015336  012703  002264                MOV     #BEGPAT,R3      ;GET START OF LIST
1371   015342                       BGNSEG                          ;****START OF SEGMENT****
  (3)  015342  104004                        EMT     C$BSEG
1372   015344                       1$:
1373   015344  012777  177777  164562        MOV     #-1,@RLDA       ;SET RLDA TO ALL 1'S
1374   015352  012737  177777  002234        MOV     #-1,GDDAT       ;SET EXPECTED DATA
1375   015360  041337  002234                BIC     (R3),GDDAT      ;SET EXPECTED DATA
1376   015364  041377  164544                BIC     (R3),@RLDA      ;'BIC'' RLDA
1377   015370  017737  164540  002236        MOV     @RLDA,BDDAT     ;READ RLDA
1378   015376  023737  002234  002236        CMP     GDDAT,BDDAT     ;DID 'BIC'' WORK?
1379   015404  001404                        BEQ     2$              ;IF IT DID BRANCH
1380
1381   015406                               ERRDF   12.,EM66,ERR2   ;WRONG DATA IN RLDA
  (3)  015406  104462                        TRAP    T$ERCODE
  (5)  015410  000014                        .WORD   12
  (5)  015412  007106                        .WORD   EM66
  (5)  015414  010046                        .WORD   ERR2
1382   015416                       2$:     ESCAPE  SEG             ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
  (3)  015416  104010                        EMT     C$ESCAPE
  (3)  015420  000012                        .WORD   10000$-.
1383
1384   015422  005723                        TST     (R3)+           ;GET NEXT PATTERN
1385   015424  020327  002472                CMP     R3,#ENDPAT      ;DONE?
1386   015430  001345                        BNE     1$              ;NO GO BACK
1387   015432                       ENDSEG                          ;****END OF SEGMENT****
  (3)  015432                        10000$:
```

```
  (3)  015432  104005                        EMT     C$ESEG
1388   015434                        ENDTST                          ;****END OF TEST****
  (3)  015434                        L10033:
  (3)  015434  104001                        EMT     C$ETST
1389
1390
1391                                 .SBTTL  **TEST 14** - BUS RESET OF RLCS
1392
1393   015436                        BGNTST                          ;****START OF TEST****
1394
1395   015436                        STARS
  (2)                                ;;*****************************************************************
1396                                 ;TEST THAT A BUS RESET WILL CLEAR THE PROPER BITS
1397                                 ;OF THE CONTROL AND STATUS REGISTER.  THOSE BITS ARE
1398                                 ;BITS 6-1,8,9,10,11,12,13,15.  BIT 15 WILL CLEAR ONLY
1399                                 ;IF BIT 14 (DRIVE ERROR IS NOT SET).  BIT 0 (DRIVE READY)
1400                                 ;IS A DON'T CARE.  IF AT THE START UP THIS TEST BIT
1401                                 ;14 (DRIVE ERROR) IS SET WE WILL INSIST IF IS THERE AFTER
1402                                 ;THE 'RESET' ALONG WITH BIT 15 (COMPOSITE ERROR).  BITS
1403                                 ;15-10 ARE NOT WRITEABLE.
1404   015436                        STARS
  (2)                                ;;*****************************************************************
1405
1406
1407   015436                              SETPRI  #PRI07          ;PRIORITY TO SEVEN
  (3)  015436  012700  000340              MOV     #PRI07,R0
  (3)  015442  104041                       EMT     C$SPRI
1408   015444  012777  000377  164456       MOV     #377,@RLCS      ;LOAD ALL RLCS LOADABLE BITS
1409   015452  012737  000200  002234       MOV     #CRDY,GDDAT     ;SETUP EXPECTED
1410   015460  032777  040000  164442       BIT     #DERR,@RLCS     ;DRIVE ERR SET?
1411   015466  001403                       BEQ     1$              ;IF NOT DON'T EXPECT IT
1412   015470  052737  140000  002234       BIS     #DERR!ERR,GDDAT  ;IT'S SET, INIT BETTER NOT CLR
1413   015476  012700  000100        1$:    MOV     #100,R0         ;SET UP A WAIT LOOP
1414   015502                              BRESET                  ;BUS RESET
  (3)  015502  104033                       EMT     C$RESET
1415   015504  005300                2$:    DEC     R0              ;WAIT IN CASE OF DRIVE ERROR
1416   015506  001376                       BNE     2$              ;
1417   015510  017737  164414  002236       MOV     @RLCS,BDDAT     ;READ RLCS
1418   015516  042737  000001  002236       BIC     #DRDY,BDDAT     ;CLEAR OUT DRDY - DON'T CARE
1419   015524  023737  002236  002234       CMP     BDDAT,GDDAT     ;DID INIT WORK
1420   015532  001404                       BEQ     3$              ;YES, BRANCH
1421
1422   015534                              ERRDF   13.,EM67,ERR2   ;WRONG DATA IN RLCS
  (3)  015534  104462                       TRAP    T$ERCODE
  (5)  015536  000015                       .WORD   13
  (5)  015540  007171                       .WORD   EM67
  (5)  015542  010046                       .WORD   ERR2
1423   015544                        3$:
1424   015544                        ENDTST                          ;****END OF TEST****
  (3)  015544                        L10034:
  (3)  015544  104001                       EMT     C$ETST
1425
1426
1427                                 .SBTTL  **TEST 15** - BUS RESET OF RLBA
1428
1429   015546                        BGNTST                          ;****START OF TEST****
```

```
1430
1431  015546                          STARS
 (2)                                  ;;*****************************************************************
1432                                  ;TEST THAT A BUS RESET WILL CLEAR THE ENTIRE
1433                                  ;BUS ADDRESS REGISTER.  THE BUS ADDRESS IS LOADED WITH 177776
1434                                  ;AND IS EXPECTED TO BE ZERO AFTER THE RESET
1435  015546                          STARS
 (2)                                  ;;*****************************************************************
1436
1437
1438  015546  012777  177776  164356          MOV     #-2,@RLBA       ;SET BA TO ALL 1'S
1439  015554  005737  002260                  TST     T.CNTLR         ;RL11??
1440  015560  001403                          BEQ     2$              ;NO
1441  015562  052777  000001  164342          BIS     #1,@RLBA
1442  015570  005037  002234          2$:     CLR     GDDAT           ;CLEAR EXPECTED DATA
1443  015574                                  BRESET                  ;ISSUE BUS INIT
 (3)  015574  104033                          EMT     C$RESET
1444  015576  017737  164330  002236          MOV     @RLBA,BDDAT     ;READ RLBA
1445  015604  001404                          BEQ     1$              ;IF CLEAR BRANCH
1446
1447  015606                                  ERRDF   14.,EM70,ERR2   ;WRONG DATA IN RLBA
 (3)  015606  104462                          TRAP    T$ERCODE
 (5)  015610  000016                          .WORD   14
 (5)  015612  007226                          .WORD   EM70
 (5)  015614  010046                          .WORD   ERR2
1448  015616                          1$:
1449
1450  015616                          ENDTST                          ;****END OF TEST****
 (3)  015616                          L10035:
 (3)  015616  104001                          EMT     C$ETST
1451
1452
1453                                  .SBTTL  **TEST 16** - BUS RESET OF RLDA
1454
1455  015620                          BGNTST                          ;****START OF TEST****
1456
1457  015620                          STARS
 (2)                                  ;;*****************************************************************
1458                                  ;TEST THAT A BUS RESET WILL CLEAR THE ENTIRE
1459                                  ;DISK ADDRESS REGISTER.  THE DISK ADDRESS IS LOADED WITH 177777
1460                                  ;AND IS EXPECTED TO BE ZERO AFTER THE RESET.
1461  015620                          STARS
 (2)                                  ;;*****************************************************************
1462
1463
1464  015620  012777  177777  164306          MOV     #-1,@RLDA       ;SET DA TO ALL 1'S
1465  015626  005037  002234                  CLR     GDDAT           ;CLEAR EXPECTED
1466  015632                                  BRESET                  ;ISSUE BUS INIT
 (3)  015632  104033                          EMT     C$RESET
1467  015634  017737  164274  002236          MOV     @RLDA,BDDAT     ;READ RLDA
1468  015642  001404                          BEQ     1$              ;IF CLEAR BRANCH
1469
1470  015644                                  ERRDF   15.,EM71,ERR2   ;WRONG DATA IN RLDA
 (3)  015644  104462                          TRAP    T$ERCODE
 (5)  015646  000017                          .WORD   15
 (5)  015650  007263                          .WORD   EM71
```

```
  (5)  015652  010046                        .WORD   ERR2
1471   015654                          1$:
1472
1473   015654                          ENDTST                          ;****END OF TEST****
  (3)  015654                          L10036:
  (3)  015654  104001                          EMT     C$ETST
1474
1475
1476                                   .SBTTL  **TEST 17** - UNIQUENESS OF RLCS
1477
1478   015656                          BGNTST                          ;****START OF TEST****
1479
1480   015656                          STARS
  (2)                                  ;;*****************************************************************
1481                                   ;TEST THE UNIQUENESS OF THE CONTROL AND STATUS
1482                                   ;REGISTER.  THE RLBA AND RLDA ARE PRELOADED WITH
1483                                   ;177776 AND 177777 RESPECTIVELY.  THE RLCS IS THEN
1484                                   ;LOADED TO INSURE THAT NEITHER THE RLBA OR RLDA
1485                                   ;ARE MODIFIED BY THE WRITING OF THE RLCS.
1486   015656                          STARS
  (2)                                  ;;*****************************************************************
1487
1488
1489   015656  012737  000201  002204          MOV     #DRDY!CRDY,LDCSR        ;SET DRIVE AND CONTROLLER READY
1490   015664  012777  177776  164240          MOV     #-2,@RLBA               ;SET RLBA TO ALL 1'S
1491   015672  012777  177777  164234          MOV     #-1,@RLDA               ;SET RLDA TO ALL 1'S
1492   015700  013777  002204  164222          MOV     LDCSR,@RLCS             ;WRITE RLCS
1493
1494                                   ;CHECK THAT RLBA REMAINED UNEFFECTED
1495
1496   015706  022777  177776  164216          CMP     #-2,@RLBA       ;RLBA OKAY?
1497   015714  001412                          BEQ     1$              ;YES, GO CHECK DA
1498
1499   015716  012737  177776  002234          MOV     #-2,GDDAT       ;SET UP EXPECTED
1500   015724  017737  164202  002236          MOV     @RLBA,BDDAT     ;READ RLBA
1501
1502   015732                                  ERRDF   16.,EM72,ERR2   ;CS MODIFIED BA
  (3)  015732  104462                          TRAP    T$ERCODE
  (5)  015734  000020                          .WORD   16
  (5)  015736  007320                          .WORD   EM72
  (5)  015740  010046                          .WORD   ERR2
1503   015742                          1$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  015742  104006                          EMT     C$CLP1
1504
1505   015744  022777  177777  164162          CMP     #-1,@RLDA       ;RLDA OKAY?
1506   015752  001412                          BEQ     2$              ;YES, CONTINUE
1507
1508   015754  012737  177777  002234          MOV     #-1,GDDAT       ;SET UP EXPECTED
1509   015762  017737  164146  002236          MOV     @RLDA,BDDAT     ;READ DA
1510
1511   015770                                  ERRDF   17.,EM73,ERR2   ;CS MODIFIED DA
  (3)  015770  104462                          TRAP    T$ERCODE
  (5)  015772  000021                          .WORD   17
  (5)  015774  007353                          .WORD   EM73
  (5)  015776  010046                          .WORD   ERR2
1512   016000                          2$:
```

```
1513
1514
1515  016000                          ENDTST                          ;****END OF TEST****
 (3)  016000                          L10037:
 (3)  016000  104001                          EMT     C$ETST
1516
1517
1518                                  .SBTTL  **TEST 18** - UNIQUENESS OF RLBA
1519
1520  016002                          BGNTST                          ;****START OF TEST****
1521  016002                          STARS
 (2)                                  ;;*****************************************************************
1522                                  ;TEST THE UNIQUENESS OF THE BUS ADDRESS REGISTER.  THE
1523                                  ;RLCS AND RLDA ARE LOADED WITH XXX20X AND 177777
1524                                  ;RESPECTIVELY.  THE RLBA IS THEN WRITTEN TO INSURE
1525                                  ;THAT NEITHER THE RLCS OR RLDA ARE MODIFIED
1526                                  ;BY WRITING THE RLBA.
1527  016002                          STARS
 (2)                                  ;;*****************************************************************
1528
1529
1530  016002  012737  000200  002234          MOV     #CRDY,GDDAT     ;CONTROLLER READY
1531  016010  032777  040000  164112          BIT     #DERR,@RLCS     ;IF DRIVE ERROR IS
1532  016016  001403                          BEQ     99$             ;SET THEN EXPECT IT
1533  016020  052737  140000  002234          BIS     #ERR!DERR,GDDAT ;SET WHEN WE READ IT.
1534  016026  013777  002234  164074  99$:    MOV     GDDAT,@RLCS     ;LOAD RLCS
1535  016034  012777  177777  164072          MOV     #-1,@RLDA       ;LOAD RLDA
1536  016042  005077  164064                  CLR     @RLBA           ;CLEAR RLBA
1537
1538                                  ;CHECK IF RLCS IS OKAY
1539
1540  016046  017737  164056  002236          MOV     @RLCS,BDDAT     ;READ RLCS
1541  016054  042737  000001  002236          BIC     #DRDY,BDDAT     ;IGNORE DRIVE READY
1542  016062  023737  002236  002234          CMP     BDDAT,GDDAT     ;CS OK?
1543  016070  001404                          BEQ     1$              ;YES, GO CHECK DA
1544
1545  016072                                  ERRDF   18.,EM74,ERR2   ;BA MODIFIED CS
 (3)  016072  104462                          TRAP    T$ERCODE
 (5)  016074  000022                          .WORD   18
 (5)  016076  007406                          .WORD   EM74
 (5)  016100  010046                          .WORD   ERR2
1546  016102                          1$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  016102  104006                          EMT     C$CLP1
1547
1548  016104  022777  177777  164022          CMP     #-1,@RLDA       ;IS RLDA OKAY?
1549
1550  016112  001412                          BEQ     2$              ;IF OKAY BRANCH
1551
1552  016114  012737  177777  002234          MOV     #-1,GDDAT       ;SET UP EXPECTED
1553  016122  017737  164006  002236          MOV     @RLDA,BDDAT     ;READ RLDA
1554
1555  016130                                  ERRDF   19.,EM75,ERR2   ;BA MODIFIED DA
 (3)  016130  104462                          TRAP    T$ERCODE
 (5)  016132  000023                          .WORD   19
 (5)  016134  007440                          .WORD   EM75
 (5)  016136  010046                          .WORD   ERR2
```

```
                                2$:
                                ENDTST                          ;****END OF TEST****
                                L10040:
(3)  016140  104001                     EMT     C$ETST

                                .SBTTL  **TEST 19** - UNIQUENESS OF RLDA

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

                                STARS
(2)                             ;;**************************************************************************
                                ;TEST THE UNIQUENESS OF THE DISK ADDRESS REGISTER.  THE RLCS
                                ;AND RLBA ARE LOADED WITH XXX20X AND 177776
                                ;RESPECTIVELY.  THE RLDA IS THEN WRITTEN TO INSURE
                                ;THAT NEITHER THE RLCS OR THE RLBA ARE MODIFIED
                                ;BY WRITING THE RLDA.
                                STARS
(2)                             ;;**************************************************************************

016142  012737  000200  002234          MOV     #CRDY,GDDAT     ;CONTROLLER READY
016150  032777  040000  163752          BIT     #DERR,@RLCS     ;IF DRIVE ERROR SET
016156  001403                          BEQ     99$             ;THEN EXPECT IT LATER
016160  052737  140000  002234          BIS     #ERR!DERR,GDDAT
016166  013777  002234  163734  99$:    MOV     GDDAT,@RLCS     ;LOAD CS
016174  012777  177776  163730          MOV     #-2,@RLBA       ;LOAD BA WITH ALL 1'S
016202  005077  163726                  CLR     @RLDA           ;CLEAR RLDA

                                ;CHECK IF RLCS IS OKAY

016206  017737  163716  002236          MOV     @RLCS,BDDAT     ;READ RLCS
016214  042737  000001  002236          BIC     #DRDY,BDDAT     ;IGNORE DRIVE READY
016222  023737  002234  002236          CMP     GDDAT,BDDAT     ;RLCS OKAY?
016230  001404                          BEQ     1$              ;YES, THEN BRANCH

016232                                  ERRDF   20.,EM76,ERR2   ;DA MODIFIED CS
(3)  016232  104462                     TRAP    T$ERCODE
(5)  016234  000024                     .WORD   20
(5)  016236  007472                     .WORD   EM76
(5)  016240  010046                     .WORD   ERR2
016242                          1$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
(3)  016242  104006                     EMT     C$CLP1

016244  022777  177776  163660          CMP     #-2,@RLBA       ;IS RLBA OKAY?
016252  001412                          BEQ     2$              ;BRANCH IF OKAY

016254  012737  177776  002234          MOV     #-2,GDDAT       ;SET UP EXPECTED
016262  017737  163644  002236          MOV     @RLBA,BDDAT     ;READ RLBA

016270                                  ERRDF   21.,EM77,ERR2   ;DA MODIFIED BA
(3)  016270  104462                     TRAP    T$ERCODE
(5)  016272  000025                     .WORD   21
(5)  016274  007525                     .WORD   EM77
(5)  016276  010046                     .WORD   ERR2
```

```
1599  016300                          2$:
1600
1601
1602  016300                          ENDTST                          ;****END OF TEST****
 (3)  016300                          L10041:
 (3)  016300  104001                          EMT     C$ETST
1603
1604                                  .SBTTL  **TEST 20** - UNIQUENESS OF RLMP
1605
1606  016302                          BGNTST                          ;****START OF TEST****
1607
1608
1609  016302                          STARS
 (2)                                  ;;**************************************************************
1610                                  ;TEST THE UNIQUENESS OF THE MULTI-PURPOSE REGISTER
1611                                  ;WE WILL WRITE THE RLCS, RLBA, AND THE RLDA, THEN THE
1612                                  ;RLMP IS WRITTEN. WE THEN GO BACK AN VERIFY THE CONTENTS
1613                                  ;OF THE RLCS, RLBA, RLDA.
1614  016302                          STARS
 (2)                                  ;;**************************************************************
1615
1616
1617  016302  012737  000200  002234          MOV     #CRDY,GDDAT     ;CONTROLLER READY
1618  016310  032777  040000  163612          BIT     #DERR,@RLCS     ;IF DRIVE ERROR SET
1619  016316  001403                          BEQ     99$             ;THE EXPECT IT LATER
1620  016320  052737  140000  002234          BIS     #ERR!DERR,GDDAT
1621  016326  013777  002234  163574  99$:    MOV     GDDAT,@RLCS     ;LOAD CS
1622  016334  012777  177776  163570          MOV     #-2,@RLBA       ;LOAD BA WITH ALL 1'S
1623  016342  012777  177777  163564          MOV     #-1,@RLDA       ;LOAD RLDA
1624  016350  005077  163562                  CLR     @RLMP           ;WRITE RLMP
1625
1626                                  ;CHECK IF RLCS IS OKAY
1627
1628  016354  017737  163550  002236          MOV     @RLCS,BDDAT     ;READ RLCS
1629  016362  042737  000001  002236          BIC     #DRDY,BDDAT     ;IGNORE DRIVE READY
1630  016370  023737  002234  002236          CMP     GDDAT,BDDAT     ;RLCS OKAY?
1631  016376  001404                          BEQ     1$              ;YES, THEN BRANCH
1632
1633  016400                                  ERRDF   201.,EM44,ERR2  ;MP MODIFIED CS
 (3)  016400  104462                          TRAP    T$ERCODE
 (5)  016402  000311                          .WORD   201
 (5)  016404  006041                          .WORD   EM44
 (5)  016406  010046                          .WORD   ERR2
1634  016410                          1$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  016410  104006                          EMT     C$CLP1
1635
1636  016412  022777  177776  163512          CMP     #-2,@RLBA       ;IS RLBA OKAY?
1637  016420  001412                          BEQ     2$              ;BRANCH IF OKAY
1638
1639  016422  012737  177776  002234          MOV     #-2,GDDAT       ;SET UP EXPECTED
1640  016430  017737  163476  002236          MOV     @RLBA,BDDAT     ;READ RLBA
1641
1642  016436                                  ERRDF   211.,EM45,ERR2  ;MP MODIFIED BA
 (3)  016436  104462                          TRAP    T$ERCODE
 (5)  016440  000323                          .WORD   211
 (5)  016442  006074                          .WORD   EM45
```

```
  (5)  016444  010046                          .WORD   ERR2
1643   016446                        2$:       CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  016446  104006                          EMT     C$CLP1
1644   016450  022777  177777  163456          CMP     #-1,@RLDA       ;DISK ADDRESS OKAY
1645   016456  001412                          BEQ     3$              ;YES, CONTINUE
1646
1647   016460  017737  163450  002236          MOV     @RLDA,BDDAT     ;SET UP BAD
1648   016466  012737  177777  002234          MOV     #-1,GDDAT       ;SET UP EXPECTED
1649
1650   016474                                   ERRDF   212.,EM46,ERR2  ;MP MODIFIED DA
  (3)  016474  104462                          TRAP    T$ERCODE
  (5)  016476  000324                          .WORD   212
  (5)  016500  006127                          .WORD   EM46
  (5)  016502  010046                          .WORD   ERR2
1651
1652   016504                        3$:
1653
1654
1655   016504                        ENDTST                          ;****END OF TEST****
  (3)  016504                        L10042:
  (3)  016504  104001                          EMT     C$ETST
1656
1657                                 .SBTTL  **TEST 21** - NOOP FUNCTION(RL11 ONLY)
1658
1659   016506                        BGNTST                          ;****START OF TEST****
1660
1661
1662
1663   016506                        STARS
  (2)                                ;;*****************************************************************
1664                                 ;TEST THAT NOOP WILL FUNCTION.  WE WILL ISSUE THE
1665                                 ;NOOP AND WAIT FOR CONTROLLER READY TO SET.  A
1666                                 ;TIMEOUT OF 200 MILLISECS IS ALLOWED.  DRIVE 0 IS ALWAYS
1667                                 ;SELECTED SINCE THE DRIVE IS NOT NECESSARY.
1668   016506                        STARS
  (2)                                ;;*****************************************************************
1669
1670
1671   016506  005737  002260                  TST     T.CNTLR         ;RLV11??
1672   016512  001410                          BEQ     99$             ;YES SKIP TEST
1673
1674
1675   016514  004537  013014                  JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
1676   016520  000000                          NOOP0                   ;NOOP(0) FUNCTION
1677   016522  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
1678   016526                        2$:       CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  016526  104006                          EMT     C$CLP1
1679
1680   016530  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
1681
1682   016534                        99$:
1683   016534                        ENDTST                          ;****END OF TEST****
  (3)  016534                        L10043:
  (3)  016534  104001                          EMT     C$ETST
1684
1685
```

```
                                .SBTTL  **TEST 22** - TEST NOOP DOES NOTHING

016536                          BGNTST                          ;****START OF TEST****

016536                          STARS
 (2)                            ;;*****************************************************************
                                ;TEST THAT ISSUING A NOOP FUNCTION DOES NOTHING.  THIS IS DONE BY WRITING
                                ;THE RLBA, AND RLDA, READING THE RLMP AND MAKING SURE NOTHING CHANGES.
016536                          STARS
 (2)                            ;;*****************************************************************

016536  005737  002260                TST     T.CNTLR         ;RLV11??
016542  001476                        BEQ     3$

016544  012777  000001  163362        MOV     #1,@RLDA        ;LOAD DISK ADDRESS
016552  012777  000002  163352        MOV     #2,@RLBA        ;LOAD BUS ADDRESS
016560  005077  163352                CLR     @RLMP
016564  017737  163346  002234        MOV     @RLMP,GDDAT     ;READ RLMP

016572  004537  013014                JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
016576  000000                        NOOP0
016600  004537  013624                JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
016604                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)    016604  104006                EMT     C$CLP1

016606  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
016612                                ESCAPE  TST             ;IF /FL:LOE SET LOOP, ELSE EXIT TST
 (3)    016612  104010                EMT     C$ESCAPE
 (3)    016614  000124                .WORD   L10044-.

016616  017737  163314  002236        MOV     @RLMP,BDDAT     ;READ RLMP
016624  023737  002234  002236        CMP     GDDAT,BDDAT     ;RLMP OK?
016632  001404                        BEQ     1$

016634                                ERRDF   202.,EM14,ERR2
 (3)    016634  104462                TRAP    T$ERCODE
 (5)    016636  000312                .WORD   202
 (5)    016640  005141                .WORD   EM14
 (5)    016642  010046                .WORD   ERR2

016644                          1$:   CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)    016644  104006                EMT     C$CLP1

016646  012737  000002  002234        MOV     #2,GDDAT        ;SET UP EXP'D BA
016654  017737  163252  002236        MOV     @RLBA,BDDAT     ;READ BA
016662  023737  002234  002236        CMP     GDDAT,BDDAT     ;BA OK?
016670  001404                        BEQ     2$              ;YES

016672                                ERRDF   203.,EM15,ERR2
 (3)    016672  104462                TRAP    T$ERCODE
 (5)    016674  000313                .WORD   203
 (5)    016676  005167                .WORD   EM15
 (5)    016700  010046                .WORD   ERR2

016702                          2$:   CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)    016702  104006                EMT     C$CLP1
```

```
1727
1728  016704  012737  000001  002234          MOV     #1,GDDAT        ;SET UP EXP'D DA
1729  016712  017737  163216  002236          MOV     @RLDA,BDDAT     ;READ DA
1730  016720  023737  002234  002236          CMP     GDDAT,BDDAT     ;DA OKAY
1731  016726  001404                          BEQ     3$
1732
1733  016730                                  ERRDF   204.,EM16,ERR2
 (3)  016730  104462                          TRAP    T$ERCODE
 (5)  016732  000314                          .WORD   204
 (5)  016734  005215                          .WORD   EM16
 (5)  016736  010046                          .WORD   ERR2
1734
1735  016740                          3$:
1736
1737  016740                          ENDTST                          ;****END OF TEST****
 (3)  016740                          L10044:
 (3)  016740  104001                          EMT     C$ETST
1738
1739
1740                                  .SBTTL  **TEST 23** - TEST OF INTERRUPT
1741
1742  016742                          BGNTST                          ;****START OF TEST****
1743
1744  016742                          STARS
 (2)                                  ;;*****************************************************************
1745                                  ;CHECK THE INTERRUPT WITH A NOOP.  WE WILL SET UP THE
1746                                  ;INTERRUPT VECTOR, LOWER THE PSW TO ZERO AND ISSUE
1747                                  ;A NOOP.  THE INTERRUPT SERVICE ROUTINE WILL SET A
1748                                  ;FLAG UPON INTERRUPT AND RETURN IN LINE.  WE WAIT 200 MILLISECONDS
1749                                  ;LOOKING FOR THAT FLAG TO BE SET BEFORE CALLING IT
1750                                  ;AN ERROR.  IF THE INTERRUPT SENDS US TO ANOTHER
1751                                  ;VECTOR ADDRESS THEN THE ERROR HANDLER WILL REPORT
1752                                  ;''TRAP TO XXXX FROM YYYY'' AND RETURN  TO DIAG SUP MONITOR.  IF THE
1753                                  ;INTERRUPT GOES TO ABOVE 1000 WHO KNOWS WHAT WILL HAPPEN.
1754  016742                          STARS
 (2)                                  ;;*****************************************************************
1755
1756
1757  016742  005737  002260                  TST     T.CNTLR
1758  016746  001426                          BEQ     99$
1759
1760  016750  005037  002202                  CLR     INTFLG          ;CLEAR INTERRUPT OCCRUANCE FLAG
1761  016754                                  SETPRI  #PRI00          ;SET PSW TO 0
 (3)  016754  012700  000000                  MOV     #PRI00,R0
 (3)  016760  104041                          EMT     C$SPRI
1762  016762  004537  013014                  JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
1763  016766  000100                          NOOP0!INTEN             ;NOOP AND INTERRUPT ENABLE
1764  016770  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
1765  016774  005737  002202                  TST     INTFLG          ;DID INTERRUPT OCCUR
1766  017000  001004                          BNE     2$              ;IF SO BRANCH
1767  017002                                  ERRDF   22.,EM13,ERR0
 (3)  017002  104462                          TRAP    T$ERCODE
 (5)  017004  000026                          .WORD   22
 (5)  017006  005107                          .WORD   EM13
 (5)  017010  010016                          .WORD   ERR0
1768  017012  005037  002202          2$:     CLR     INTFLG
```

```
1769  017016                                        CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  017016  104006                                EMT     C$CLP1
1770  017020  004537  012514                        JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
1771
1772
1773  017024                              99$:
1774  017024                              ENDTST                          ;****END OF TEST****
 (3)  017024                              L10045:
 (3)  017024  104001                                EMT     C$ETST
1775
1776
1777                                      .SBTTL  **TEST 24** - TEST PRIORITY BR LEVEL
1778
1779  017026                              BGNTST                          ;****START OF TEST****
1780
1781  017026                              STARS
 (2)                                      ;;*****************************************************************
1782                                      ;TEST THAT PRIORITY GIVEN IS ACTUAL PRIORITY OF CONTROLLER.  WE KNOW
1783                                      ;THE BOARD WILL INTERRUPT.  WE WILL START TRYING TO INTERRUPT AT 7
1784                                      ;AND WORK DOWN TIL IT DOES INTERRUPT.
1785  017026                              STARS
 (2)                                      ;;*****************************************************************
1786
1787  017026  005737  002260                        TST     T.CNTLR         ;RLV11??
1788  017032  001456                                BEQ     6$              ;YES, SKIP TEST
1789
1790  017034  012737  000340  002236                MOV     #340,BDDAT      ;SET UP INITIAL OF 7
1791  017042  013737  002142  002234                MOV     BPRIOR,GDDAT    ;GET GIVEN PRIORITY
1792
1793  017050                              BGNSEG                          ;****START OF SEGMENT****
 (3)  017050  104004                                EMT     C$BSEG
1794
1795  017052  005037  002202              5$:       CLR     INTFLG          ;CLEAR INTERRUPT OCCURRANCE
1796  017056                                        SETPRI  BDDAT           ;SET PRIORITY
 (3)  017056  013700  002236                        MOV     BDDAT,R0
 (3)  017062  104041                                EMT     C$SPRI
1797
1798  017064  004537  013014                        JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
```

```
                                NOOP0!INTEN

                                JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
                                ESCAPE  TST             ;IF /FL:LOE SET LOOP, ELSE EXIT TST
                                EMT     C$ESCAPE
                                .WORD   L10046-.

                                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
                                ESCAPE  TST             ;IF /FL:LOE SET LOOP, ELSE EXIT TST
                                EMT     C$ESCAPE
                                .WORD   L10046-.

                                CMP     BDDAT,GDDAT     ;SHOULD IT INTERRUPT
                                BGE     1$              ;NO, BRANCH

                                TST     INTFLG          ;DID INTERRUPT OCCUR
                                BNE     2$              ;YES, OK

                        3$:     ERRDF   204.,EM17,ERR7
                                TRAP    T$ERCODE
                                .WORD   204
                                .WORD   EM17
                                .WORD   ERR7

                        2$:     ESCAPE  SEG             ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
                                EMT     C$ESCAPE
                                .WORD   10000$-.
                                BR      4$
                        1$:     TST     INTFLG          ;DID INTERRUPT OCCUR
                                BEQ     2$              ;NO, OK
                                BR      3$              ;YES, ERROR

                        ENDSEG                          ;****END OF SEGMENT****
                        10000$:
                                EMT     C$ESEG
                        4$:     SUB     #40,BDDAT       ;NEXT LEVEL
                                BPL     5$

                        6$:
                        ENDTST                          ;****END OF TEST****
                        L10046:
                                EMT     C$ETST

                        .SBTTL  **TEST 25** - GET STATUS FUNCTION

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


                        STARS
                        ;;*************************************************************************
                        ;TEST GET STATUS FUNCTION.  THE GET STATUS FUNCTION WILL
                        ;WORK IF DRIVE IS LOADED AND READY OR NOT.  THE RLDA
                        ;IS LOADED WITH THE GET STATUS AND MARKER BITS (BITS 1,0)
                        ;AND THE FUNCTION IS ISSUED.  WE WAIT 200 MILLISECONDS
                        ;FOR CONTROLLER READY.  VERIFY THAT NO ERRORS OCCUR.
                        STARS
```

```
 (2)                                    ;;*****************************************************************************
1841
1842
1843  017172  012777  000013  162734            MOV     #GSBIT!MK!DRST,@RLDA    ;SET GET STATUS AND MARKER BIT
1844  017200  004537  013014                    JSR     R5,LDFUNC               ;ISSUE FUNCTION OF FOLLOWING WORD
1845  017204  000004                            GSTAT                           ;GET STATUS
1846  017206  004537  013624                    JSR     R5,WTCRDY               ;WAIT FOR CONTROLLER READY HIGH
1847  017212                            2$:     CKLOOP                          ;CHECK IF /FL:LOE IS SET
 (3)  017212  104006                            EMT     C$CLP1
1848
1849  017214  004537  012514                    JSR     R5,CHERR                ;CHECK CONTROLLER FOR ERRORS
1850
1851  017220                            ENDTST                                  ;****END OF TEST****
 (3)  017220                            L10047:
 (3)  017220  104001                            EMT     C$ETST
1852
1853
1854                                    .SBTTL  **TEST 26** - GET STATUS FUNCTION INTERRUPT
1855
1856  017222                            BGNTST                                  ;****START OF TEST****
1857
1858                                    ;CHECK GET STATUS UNDER INTERRUPT - SHOULD GET AN INTERRUPT
1859
1860  017222  005037  002202                    CLR     INTFLG                  ;CLEAR INTERRUPT OCCURANCE
1861  017226                                    SETPRI  #PRI00                  ;PSW TO LEVEL 0
 (3)  017226  012700  000000                    MOV     #PRI00,R0
 (3)  017232  104041                            EMT     C$SPRI
1862  017234  012777  000003  162672            MOV     #GSBIT!MK,@RLDA ;SET UP DA
1863  017242  004537  013014                    JSR     R5,LDFUNC               ;ISSUE FUNCTION OF FOLLOWING WORD
1864  017246  000104                            GSTAT!INTEN                     ;GET STATUS, INT ENABLE
1865  017250  004537  013624                    JSR     R5,WTCRDY               ;WAIT FOR CONTROLLER READY HIGH
1866  017254                                    SETPRI  #PRI07
 (3)  017254  012700  000340                    MOV     #PRI07,R0
 (3)  017260  104041                            EMT     C$SPRI
1867  017262  005737  002202                    TST     INTFLG                  ;DID INTERRUPT OCCUR
1868  017266  001004                            BNE     2$                      ;YES-BRANCH
1869  017270                                    ERRDF   28.,EM30,ERR0
 (3)  017270  104462                            TRAP    T$ERCODE
 (5)  017272  000034                            .WORD   28
 (5)  017274  005276                            .WORD   EM30
 (5)  017276  010016                            .WORD   ERR0
1870  017300                            2$:     CKLOOP                          ;CHECK IF /FL:LOE IS SET
 (3)  017300  104006                            EMT     C$CLP1
1871  017302  004537  012514                    JSR     R5,CHERR                ;CHECK CONTROLLER FOR ERRORS
1872
1873  017306  005037  002202                    CLR     INTFLG                  ;CLEAR INTERRUPT OCCURANCE
1874  017312                                    SETPRI  #PRI00                  ;PSW TO LEVEL 0
 (3)  017312  012700  000000                    MOV     #PRI00,R0
 (3)  017316  104041                            EMT     C$SPRI
1875  017320  012777  000003  162606            MOV     #GSBIT!MK,@RLDA ;SET UP DA FOR GET STATUS COMMAND
1876  017326  004537  013014                    JSR     R5,LDFUNC               ;ISSUE FUNCTION OF FOLLOWING WORD
1877  017332  000004                            GSTAT                           ;GET STATUS - SHOULD NOT CAUSE AN INTERRUPT
1878  017334  004537  013624                    JSR     R5,WTCRDY               ;WAIT FOR CONTROLLER READY HIGH
1879  017340                                    SETPRI  #PRI07
 (3)  017340  012700  000340                    MOV     #PRI07,R0
 (3)  017344  104041                            EMT     C$SPRI
```

```
1880  017346  005737  002202          TST     INTFLG          ;DID INTERRUPT OCCUR
1881  017352  001404                  BEQ     3$              ;NO - BRANCH (OK)
1882  017354                          ERRDF   281.,EM30A,ERR0
 (3)  017354  104462                  TRAP    T$ERCODE
 (5)  017356  000431                  .WORD   281
 (5)  017360  005335                  .WORD   EM30A
 (5)  017362  010016                  .WORD   ERR0
1883  017364                  3$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  017364  104006                  EMT     C$CLP1
1884  017366  004537  012514          JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
1885
1886  017372                  ENDTST                          ;****END OF TEST****
 (3)  017372                  L10050:
 (3)  017372  104001                  EMT     C$ETST
1887
1888
1889                          .SBTTL  **TEST 27** - GET STATUS FUNCTION GENERATES OPI W/O GS BIT
1890
1891  017374                  BGNTST                          ;****START OF TEST****
1892
1893  017374                  STARS
 (2)                          ;;*****************************************************************
1894                          ;VERIFY THAT GET STATUS FUNCTION WILL NOT COMPLETE
1895                          ;WITHOUT SENDING OUT THE GET STATUS BIT IN THE RLDA.
1896                          ;WE SET MARKER BUT NO GET STATUS BIT IN THE RLDA AND
1897                          ;ISSUE A GET STATUS WE SHOULD RECIEVE AN OPI ERROR.
1898                          ;VERIFY THAT CONTROLLER READY SETS AND OPI SETS
1899  017374                  STARS
 (2)                          ;;*****************************************************************
1900
1901
1902  017374  012777  000001  162532          MOV     #MK,@RLDA       ;SET ONLY MARKER BIT!!
1903  017402  004537  013014                  JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
1904  017406  000004                          GSTAT                   ;GET STATUS
1905  017410  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
1906  017414  032737  074000  002162          BIT #74000,E.CS
1907  017422  001405                          BEQ     1$
1908  017424  012737  003710  012776          MOV     #OPIERR,RESTMS
1909  017432  004537  012514                  JSR     R5,CHERR
1910  017436                          1$:     CKLOOP
 (3)  017436  104006                          EMT     C$CLP1
1911  017440  032737  002000  002162          BIT     #OPI,E.CS       ;IS OPI SET?
1912  017446  001004                          BNE     2$              ;YES-BRANCH NO-CHECK TIMEOUT
1913  017450                                  ERRDF   29.,EM33,ERR0
 (3)  017450  104462                          TRAP    T$ERCODE
 (5)  017452  000035                          .WORD   29
 (5)  017454  005431                          .WORD   EM33
 (5)  017456  010016                          .WORD   ERR0
1914  017460                          2$:
1915
1916  017460                  ENDTST                          ;****END OF TEST****
 (3)  017460                  L10051:
 (3)  017460  104001                  EMT     C$ETST
1917
1918
1919                          .SBTTL  **TEST 28** - OPI UNDER INTERRUPT
```

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

                                        SETPRI  #PRI00
                                        MOV     #PRI00,R0
                                        EMT     C$SPRI
                                        CLR     INTFLG
                                        MOV     #MK,@RLDA       ;SET ONLY MARKER BIT!!
                                        JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
                                        GSTAT!INTEN             ;GET STATUS
                                        JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
                                        SETPRI  #PRI07
                                        MOV     #PRI07,R0
                                        EMT     C$SPRI
                                        TST     INTFLG          ;INTERRUPT OCCUR
                                        BNE     2$

                                        ERRDF   30.,EM11,ERR0
                                        TRAP    T$ERCODE
                                        .WORD   30
                                        .WORD   EM11
                                        .WORD   ERR0

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

                                        BIT     #74000,E.CS
                                        BEQ     1$
                                        MOV     #OPIERR,RESTMS
                                        JSR     R5,CHERR
                                1$:     CKLOOP
                                        EMT     C$CLP1
                                        BIT     #OPI,E.CS       ;IS OPI SET?
                                        BNE     3$              ;YES-BRANCH NO-CHECK TIMEOUT
                                        ERRDF   31.,EM33,ERR0
                                        TRAP    T$ERCODE
                                        .WORD   31
                                        .WORD   EM33
                                        .WORD   ERR0
                                3$:


                                ENDTST                          ;****END OF TEST****
                                L10052:
                                        EMT     C$ETST

                                .SBTTL  **TEST 29** - READ HEADER FUNCTION
```

```
1958
1959  017610                        BGNTST                          ;****START OF TEST****
1960  017610                        STARS
 (2)                                ;;*****************************************************************
1961                                ;CHECK THAT READ HEADER WORKS, THAT WE CAN ISSUE
1962                                ;IT, GET READY BACK WITHOUT ANY ERRORS SETTING.
1963  017610                        STARS
 (2)                                ;;*****************************************************************
1964
1965
1966
1967  017610  004537  013014                JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
1968  017614  000010                RDHDR                           ;READ HEADER
1969  017616  004537  013624                JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
1970                                                                ;READY
1971  017622                        2$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  017622  104006                        EMT     C$CLP1
1972
1973  017624  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
1974
1975  017630                        ENDTST                          ;****END OF TEST****
 (3)  017630                        L10053:
 (3)  017630  104001                        EMT     C$ETST
1976
1977
1978                                .SBTTL  **TEST 30** - READ HEADER FUNCTION INTERRUPT
1979
1980  017632                        BGNTST                          ;****START OF TEST****
1981
1982  017632                        STARS
 (2)                                ;;*****************************************************************
1983                                ;CHECK THAT READ HEADER WILL GENERATE AN INTERRRUPT
1984                                ;UPON COMPLETION WITHOUT ANY ERRORS SETTING
1985  017632                        STARS
 (2)                                ;;*****************************************************************
1986
1987
1988  017632                                SETPRI  #PRI00          ;PSW TO 0
 (3)  017632  012700  000000                MOV     #PRI00,R0
 (3)  017636  104041                        EMT     C$SPRI
1989  017640  005037  002202                CLR     INTFLG          ;CLEAR INTERRUPT OCCURANCE
1990  017644  004537  013014                JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
1991  017650  000110                RDHDR!INTEN                     ;READ HEADER, INTR. ENA
1992  017652  004537  013624                JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
1993  017656                                SETPRI  #PRI07
 (3)  017656  012700  000340                MOV     #PRI07,R0
 (3)  017662  104041                        EMT     C$SPRI
1994  017664  005737  002202                TST     INTFLG          ;INTERRUPT HAPPEN
1995  017670  001004                        BNE     2$              ;YES-CONTINUE
1996  017672                                ERRDF   35.,EM37,ERR0
 (3)  017672  104462                        TRAP    T$ERCODE
 (5)  017674  000043                        .WORD   35
 (5)  017676  005553                        .WORD   EM37
 (5)  017700  010016                        .WORD   ERR0
1997  017702                        2$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  017702  104006                        EMT     C$CLP1
```

```
1998
1999  017704  004537  012514              JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2000
2001  017710                        ENDTST                        ;****END OF TEST****
 (3)  017710                        L10054:
 (3)  017710  104001                       EMT     C$ETST
2002
2003
2004                                .SBTTL  **TEST 31** - REPEATED RD HDRS YIELD SAME CYL AND HD
2005
2006  017712                        BGNTST                        ;****START OF TEST****
2007
2008
2009  017712                        STARS
 (2)                                ;;*****************************************************************
2010                                ;CHECKT THAT READ HEADERS WILL RELIABLY READ THE SAME
2011                                ;CYLINDER AND HEAD SELECT.  WE WILL READ HEADERS VERIFYING
2012                                ;THAT WE ALWAYS READ THE SAME CYLINDER AND HEAD SELECT.
2013  017712                        STARS
 (2)                                ;;*****************************************************************
2014
2015
2016  017712  012701  000144              MOV     #100.,R1        ;SET UP TO DO 100 RD HDR'S
2017  017716  004537  013014              JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2018  017722  000010                      RDHDR                   ;READ HEADER
2019  017724  004537  013624              JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2020  017730                        99$:    ESCAPE  TST             ;IF /FL:LOE SET LOOP, ELSE EXIT TST
 (3)  017730  104010                      EMT     C$ESCAPE
 (3)  017732  000122                      .WORD   L10055-.
2021
2022  017734  004537  012514              JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2023  017740                              ESCAPE  TST             ;IF /FL:LOE SET LOOP, ELSE EXIT TST
 (3)  017740  104010                      EMT     C$ESCAPE
 (3)  017742  000112                      .WORD   L10055-.
2024
2025  017744  013737  002170  002234      MOV     E.MP,GDDAT      ;READ FIRST HEADER (ASSUME GOOD)
2026  017752  043737  002206  002234      BIC     SECMSK,GDDAT    ;MASK AWAY SECTOR BITS
2027  017760                        BGNSEG                        ;****START OF SEGMENT****
 (3)  017760  104004                      EMT     C$BSEG
2028  017762                        2$:
2029  017762  004537  013014              JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2030  017766  000010                      RDHDR
2031  017770  004537  013624              JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2032  017774                        97$:    ESCAPE  SEG             ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
 (3)  017774  104010                      EMT     C$ESCAPE
 (3)  017776  000054                      .WORD   10000$-.
2033
2034  020000  004537  012514              JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2035  020004                              ESCAPE  SEG             ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
 (3)  020004  104010                      EMT     C$ESCAPE
 (3)  020006  000044                      .WORD   10000$-.
2036
2037  020010  013737  002170  002236      MOV     E.MP,BDDAT      ;READ HEADER
2038  020016  043737  002206  002236      BIC     SECMSK,BDDAT    ;MASK AWAY SECTOR BITS
2039  020024  023737  002234  002236      CMP     GDDAT,BDDAT     ;IS HEADER CORRECT
2040  020032  001404                      BEQ     4$
```

```
2041
2042  020034                          ERRDF   36.,EM41,ERR4
 (3)  020034  104462                  TRAP    T$ERCODE
 (5)  020036  000044                  .WORD   36
 (5)  020040  005613                  .WORD   EM41
 (5)  020042  010162                  .WORD   ERR4
2043                                                    ;CONSTANT CYL & HS
2044  020044                    4$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  020044  104006                  EMT     C$CLP1
2045
2046  020046  005301                  DEC     R1              ;PERFORM ALL READ HDR'S
2047  020050  001344                  BNE     2$              ;IF NOT GO BACK AND DO ANOTHER
2048  020052                    ENDSEG                        ;****END OF SEGMENT****
 (3)  020052                    10000$:
 (3)  020052  104005                  EMT     C$ESEG
2049  020054                    ENDTST                        ;****END OF TEST****
 (3)  020054                    L10055:
 (3)  020054  104001                  EMT     C$ETST
2050
2051
2052                            .SBTTL  **TEST 32** - CHECK OF HEADER CRC
2053
2054  020056                    BGNTST                        ;****START OF TEST****
2055
2056  020056                    STARS
 (2)                            ;;*************************************************************
2057                            ;CHECK THAT WE CAN READ THE HDCRC AFTER A
2058                            ;READ HEADER AND THAT IT IS THE CORRECT CRC
2059                            ;FOR THE HEADER.
2060  020056                    STARS
 (2)                            ;;*************************************************************
2061
2062
2063  020056  005037  020126            CLR     3$
2064  020062  004537  013014            JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2065  020066  000010                    RDHDR                   ;READ HEADER
2066  020070  004537  013624            JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2067  020074                    ESCAPE  TST                     ;IF /FL:LOE SET LOOP, ELSE EXIT TST
 (3)  020074  104010                  EMT     C$ESCAPE
 (3)  020076  000114                  .WORD   L10056-.
2068
2069  020100  004537  012514            JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2070  020104                    ESCAPE  TST                     ;IF /FL:LOE SET LOOP, ELSE EXIT TST
 (3)  020104  104010                  EMT     C$ESCAPE
 (3)  020106  000104                  .WORD   L10056-.
2071
2072  020110  013737  002170  020124    MOV     E.MP,2$         ;READ HEADER WORD
2073  020116  004537  013372            JSR     R5,SIMBCC       ;GO CALCULATE CRC
2074  020122  000020                    16.                     ;16 BITS
2075  020124  000000            2$:     .WORD   0               ;HEADER GOES HERE
2076  020126  000000            3$:     .WORD   0               ;START WITH 0 CRC
2077  020130  013737  002216  020154    MOV     CALBCC,5$
2078  020136  013737  002172  020152    MOV     E.MP1,4$        ;GET SECOND HALF
2079  020144  004537  013372            JSR     R5,SIMBCC
2080  020150  000020                    16.
2081  020152  000000            4$:     .WORD   0
```

```
2082  020154  000000                          5$:     .WORD   0
2083  020156  013737  002216  002234                  MOV     CALBCC,GDDAT    ;STORE CALCULATED CRC AS GOOD
2084  020164  013737  002174  002236                  MOV     E.MP2,BDDAT     ;THIRD READ OF DB GETS CRC
2085  020172  023737  002234  002236                  CMP     GDDAT,BDDAT     ;IS CRC CORRECT?
2086  020200  001404                                  BEQ     6$              ;IF SO CONTINUE
2087
2088  020202                                          ERRDF   37.,EM42,ERR4
  (3) 020202  104462                                  TRAP    T$ERCODE
  (5) 020204  000045                                  .WORD   37
  (5) 020206  005704                                  .WORD   EM42
  (5) 020210  010162                                  .WORD   ERR4
2089  020212                                    6$:
2090
2091  020212                                    ENDTST                          ;****END OF TEST****
  (3) 020212                                    L10056:
  (3) 020212  104001                                  EMT     C$ETST
2092
2093
2094                                            .SBTTL  **TEST 33** - CHECK CONSECUTIVE HEADERS
2095
2096  020214                                    BGNTST                          ;****START OF TEST****
2097
2098
2099  020214                                    STARS
  (2)                                           ;;**************************************************************
2100                                            ;CHECK THAT THE HEADERS ARE CONSECUTIVE.  WE WILL DO
2101                                            ;40 (FORTY) READ HEADERS AND STORE EACH.  AFTER WE HAVE
2102                                            ;READ THE FORTIETH HEADER WE WILL VERIFY THAT
2103                                            ;THEY CAME IN SEQUENTIAL, THAT 0 FOLLOWS 39,
2104                                            ;THAT THERE WERE NO ERRORS.
2105  020214                                    STARS
  (2)                                           ;;**************************************************************
2106
2107
2108  020214  005037  002240                          CLR     FIRST           ;CLEAR FIRST READ DONE FLAG
2109  020220  012703  003070                          MOV     #HDRBUF,R3      ;STORE HEADERS
2110  020224  012701  000050                          MOV     #40.,R1         ;FOURTY HEADERS
2111  020230  012737  000210  002150                  MOV     #RDHDR!CRDY,B.CS
2112  020236  053737  002146  002150                  BIS     DRIVE,B.CS
2113  020244  013777  002150  161656                  MOV     B.CS,@RLCS
2114  020252  042777  000200  161650  2$:             BIC     #200,@RLCS
2115  020260  032777  000200  161642  1$:             BIT     #200,@RLCS      ;DONE?
2116  020266  001774                                  BEQ     1$
2117  020270  017723  161634                          MOV     @RLCS,(R3)+
2118  020274  017723  161636                          MOV     @RLMP,(R3)+
2119  020300  017723  161632                          MOV     @RLMP,(R3)+
2120  020304  017723  161626                          MOV     @RLMP,(R3)+
2121  020310  005301                                  DEC     R1              ;HAVE WE READ FOURTY HEADERS
2122  020312  001357                                  BNE     2$              ;GO BACK UNTIL FOURTY DONE
2123  020314  012703  003070                          MOV     #HDRBUF,R3      ;GET LIST OF HEADERS
2124  020320  012701  000050                          MOV     #40.,R1         ;CHECK FOURTY OF THEM
2125  020324  011337  002162                          MOV     (R3),E.CS
2126  020330  005737  002162                          TST     E.CS
2127  020334  100016                                  BPL     99$
2128  020336  012737  004147  012776                  MOV     #RHDMES,RESTMS
2129  020344  005723                                  TST     (R3)+
```

```
 2130  020346  012337  002170                    MOV     (R3)+,E.MP
 2131  020352  012337  002172                    MOV     (R3)+,E.MP1
 2132  020356  012337  002174                    MOV     (R3)+,E.MP2
 2133  020362  004537  012514                    JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
 2134  020366  000137  020530                    JMP     7$
 2135  020372  005723                  99$:      TST     (R3)+
 2136  020374  011337  002236                    MOV     (R3),BDDAT      ;GET HEADER
 2137  020400  005737  002240                    TST     FIRST           ;IS THIS FIRST READ?
 2138  020404  001007                            BNE     4$              ;NO, BRANCH
 2139  020406  012737  000001  002240            MOV     #1,FIRST        ;SET FIRST READ DONE FLAG
 2140  020414  013737  002236  002234  3$:       MOV     BDDAT,GDDAT     ;SET UP NEXT READ EXPECTED
 2141  020422  000435                            BR      6$              ;GO SEE IF TEST IS DONE
 2142  020424  005237  002234                4$:  INC     GDDAT           ;INCREMENT EXP'D HEADER
 2143  020430  023737  002236  002234            CMP     BDDAT,GDDAT     ;IS NEW HEADER SEQUENTIAL?
 2144  020436  001766                            BEQ     3$              ;YES THEN BRANCH
 2145  020440  033737  002206  002236            BIT     SECMSK,BDDAT    ;IS NEW HEADER ZERO?
 2146  020446  001015                            BNE     5$              ;NO, THEN ERROR GO REPORT IT
 2147  020450  013737  002234  002220            MOV     GDDAT,TEMP2     ;YES, CHECK IF LAST HEADER WAS
 2148  020456  043737  002242  002220            BIC     CYLMSK,TEMP2    ;MAX ADDRESS, IF SO BRANCH
 2149  020464  023737  002244  002220            CMP     MXSEC1,TEMP2    ;STORE NEW DATA AS OLD
 2150  020472  001750                            BEQ     3$              ;AND PERFORM NEW RD HDR
 2151  020474  043737  002206  002234            BIC     SECMSK,GDDAT    ;EXPECTING ZERO SECTOR
 2152
 2153  020502                          5$:
 2154
 2155  020502  005037  002240                    CLR     FIRST           ;ERROR WILL MAKE US MISS
 2156                                                                    ;NEXT SECTOR SEQUENTIALLY
 2157                                                                    ;START OVER; CLEAR FIRST FLAG
 2158  020506                                    ERRDF   38.,EM43,ERR2
  (3)  020506  104462                            TRAP    T$ERCODE
  (5)  020510  000046                            .WORD   38
  (5)  020512  005742                            .WORD   EM43
  (5)  020514  010046                            .WORD   ERR2
 2159  020516                          6$:       CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  020516  104006                            EMT     C$CLP1
 2160
 2161  020520  062703  000006                    ADD     #6,R3
 2162  020524  005301                            DEC     R1              ;HAVE WE DONE THIS ENOUGH
 2163  020526  001321                            BNE     99$             ;NO, GO BACK DO IT AGAIN
 2164  020530                          7$:
 2165  020530                          ENDTST                          ;****END OF TEST****
  (3)  020530                          L10057:
  (3)  020530  104001                            EMT     C$ETST
 2166
 2167
 2168                                  .SBTTL  **TEST 34** - SEEK FUNCTION
 2169
 2170  020532                          BGNTST                          ;****START OF TEST****
 2171  020532                          STARS
  (2)                                 ;;**************************************************************
 2172                                 ;CHECK THE SEEK FUNCTION RETURNS CONTROLLER READY
 2173                                 ;WITH NO ERRORS.  WE ISSUE A ONE TRACK IN WORD SEEK.
 2174                                 ;WE DO NOT CHECK THE RESULT FOR POSITION
 2175  020532                          STARS
  (2)                                 ;;**************************************************************
 2176
```

```
2177
2178  020532  012777  000205  161374          MOV     #BIT7!MK!SIGN,@RLDA  ;SET UP DA-DIFF=1,MARKER,TOWARDS
2179  020540  004537  013014                  JSR     R5,LDFUNC            ;ISSUE FUNCTION OF FOLLOWING WORD
2180  020544  000006                          SEEK                         ;SEEK
2181  020546  004537  013624                  JSR     R5,WTCRDY            ;WAIT FOR CONTROLLER READY HIGH
2182  020552                                  WAITMS  #2.
  (3) 020552  012700  000002                  MOV     #2.,R0
  (3) 020556  104026                          EMT     C$WTM
2183  020560                          2$:     CKLOOP                       ;CHECK IF /FL:LOE IS SET
  (3) 020560  104006                          EMT     C$CLP1
2184
2185
2186  020562  004537  012514                  JSR     R5,CHERR             ;CHECK CONTROLLER FOR ERRORS
2187
2188  020566                          ENDTST                               ;****END OF TEST****
  (3) 020566                          L10060:
  (3) 020566  104001                          EMT     C$ETST
2189
2190
2191                                  .SBTTL  **TEST 35** - CHECK DRIVE READY ON SEEK
2192
2193  020570                          BGNTST                               ;****START OF TEST****
2194
2195
2196  020570                          STARS
  (2)                               ;;*****************************************************************
2197                                ;CHECK THE SEEK FUNCTION RETURNS DRIVE READY WITH
2198                                ;NO ERRORS.  WE ISSUE A ONE TRACK INWARD SEEK.  WE DO
2199                                ;NOT CHECK THE RESULT FOR POSITION
2200  020570                          STARS
  (2)                               ;;*****************************************************************
2201
2202
2203
2204  020570  012777  000201  161336          MOV     #BIT7!MK,@RLDA       ;SET DA, MARKER, DIFF=1.
2205  020576  004537  013014                  JSR     R5,LDFUNC            ;ISSUE FUNCTION OF FOLLOWING WORD
2206  020602  000006                          SEEK                         ;SEEK
2207  020604  004537  013624                  JSR     R5,WTCRDY            ;WAIT FOR CONTROLLER READY HIGH
2208  020610                                  CKLOOP                       ;CHECK IF /FL:LOE IS SET
  (3) 020610  104006                          EMT     C$CLP1
2209
2210  020612  004537  012514                  JSR     R5,CHERR             ;CHECK CONTROLLER FOR ERRORS
2211  020616                                  CKLOOP                       ;CHECK IF /FL:LOE IS SET
  (3) 020616  104006                          EMT     C$CLP1
2212
2213  020620  004537  013560                  JSR     R5,WTDRDY            ;WAIT FOR DRIVE READY
2214  020624                                  CKLOOP                       ;CHECK IF /FL:LOE IS SET
  (3) 020624  104006                          EMT     C$CLP1
2215
2216  020626  004537  012514                  JSR     R5,CHERR             ;CHECK CONTROLLER FOR ERRORS
2217
2218  020632                          ENDTST                               ;****END OF TEST****
  (3) 020632                          L10061:
  (3) 020632  104001                          EMT     C$ETST
2219
2220
```

```
                                .SBTTL  **TEST 36** - SEEK FUNCTION INTERRUPT

020634                          BGNTST                          ;****START OF TEST****

020634                          STARS
                                ;;*****************************************************************
                                ;CHECK THAT CONTROLLER READY RESETTING WHEN THE SEEK IS
                                ;INITIATED CAUSES AN INTERRUPT BUT DRIVE READY WILL
                                ;NOT.  WE ALSO MONITOR FOR ANY ERROR BITS SETTING.
020634                          STARS
                                ;;*****************************************************************

020634  005037  002202                  CLR     INTFLG
020640                          SETPRI  #PRI00          ;SET PSW TO 0
020640  012700  000000                  MOV     #PRI00,R0
020644  104041                          EMT     C$SPRI
020646  012777  000205  161260          MOV     #BIT7!MK!SIGN,@RLDA  ;SET UP RLDA
020654  004537  013014                  JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
020660  000106                          SEEK!INTEN              ;SEEK AND INTR. ENA.
020662  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
020666  000240                          NOP                     ;
020670  005737  002202          1$:     TST     INTFLG          ;DID INTERRUPT OCCUR
020674  001004                          BNE     2$              ;YES, GO CHECK DRDY
020676                          ERRDF   40.,EM47,ERR0
020676  104462                  TRAP    T$ERCODE
020700  000050                  .WORD   40
020702  006162                  .WORD   EM47
020704  010016                  .WORD   ERR0
020706                  2$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
020706  104006                  EMT     C$CLP1

020710  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
020714                          CKLOOP                  ;CHECK IF /FL:LOE IS SET
020714  104006                  EMT     C$CLP1

020716  005037  002202                  CLR     INTFLG          ;CLEAR INTERRUPT OCCURANCE

020722  004537  013560                  JSR     R5,WTDRDY       ;WAIT FOR DRIVE READY
020726                  5$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
020726  104006                  EMT     C$CLP1

020730                          SETPRI  #PRI07
020730  012700  000340                  MOV     #PRI07,R0
020734  104041                          EMT     C$SPRI
020736  005737  002202                  TST     INTFLG          ;DID DRIVE READY CAUSE INTERRUPT
020742  001404                          BEQ     6$              ;NO, CONTINUE

020744                          ERRDF   42.,EM52,ERR0
020744  104462                  TRAP    T$ERCODE
020746  000052                  .WORD   42
```

```
  (5)  020750  006213                        .WORD   EM52
  (5)  020752  010016                        .WORD   ERR0
2262   020754                        6$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  020754  104006                        EMT     C$CLP1
2263
2264   020756                        ENDTST                          ;****END OF TEST****
  (3)  020756                        L10062:
  (3)  020756  104001                        EMT     C$ETST
2265
2266
2267                                 .SBTTL  **TEST 37** - TEST DIFFERENCE WORD TRANSMISSION
2268
2269   020760                        BGNTST                          ;****START OF TEST****
2270
2271
2272
2273
2274   020760                        STARS
  (2)                                ;;**************************************************************
2275                                 ;VERIFY THAT THE DIFFERENCE WORD LOADS AND IS
2276                                 ;TRANSMITTED CORRECTLY.  WE WILL ISSUE SEEKS WITH THE
2277                                 ;DIFFERENCE WORD CONTAINING ALL OF THE BIT PATTERNS FLOATING 1,
2278                                 ;GROWING 1, GROWING 0 AND SHITING 0.  THE SEEK WILL
2279                                 ;START FROM TRACK 0 EACH TIME AND WILL RETURN THERE
2280                                 ;EACH, THUS BOTH DIRECTIONS FOR PATTERNS WILL BE CHECKED.
2281                                 ;READ HEADERS ARE USED TO VERIFY THE SEEK CORRECTNESS.
2282                                 ;ERRORS ARE MONITORED AND REPORTED.
2283   020760                        STARS
  (2)                                ;;**************************************************************
2284
2285
2286   020760  012703  002474                    MOV     #SKLST,R3       ;GET LIST OF DIFFERENCE WORDS
2287   020764                        BGNSEG                              ;****START OF SEGMENT****
  (3)  020764  104004                        EMT     C$BSEG
2288   020766                        1$:
2289   020766  004537  013014                    JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2290   020772  000010                           RDHDR                   ;READ HEADER
2291   020774  004537  013624                    JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2292   021000                        98$:    CKLOOP                      ;CHECK IF /FL:LOE IS SET
  (3)  021000  104006                        EMT     C$CLP1
2293
2294   021002  004537  012514                    JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2295   021006                                   CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  021006  104006                        EMT     C$CLP1
2296
2297   021010  013737  002170  002236           MOV     E.MP,BDDAT      ;READ HEADER
2298   021016  043737  002206  002236           BIC     SECMSK,BDDAT    ;CLEAR OUT SECTOR
2299   021024  001462                           BEQ     99$             ;IF ON TRACK ZERO, H.S. ZERO, OK
2300
2301                                 ;NOT ON TRACK ZERO CALCULATE DIFFERENCE WORD AND PUT IT BACK
2302                                 ;ON ZERO.
2303
2304   021026  042737  000100  002236           BIC     #RHHS,BDDAT     ;CLEAR OUT HEAD SELECT
2305   021034  013777  002236  161072           MOV     BDDAT,@RLDA     ;PUT CYLINDER AS DIFFERENCE WORD
2306   021042  052777  000001  161064           BIS     #1,@RLDA        ;SET MARKER BIT
2307   021050  004537  013014                    JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
```

```
2308  021054  000006                        SEEK                    ;SEEK
2309  021056  004537  013624                JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2310  021062                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 021062  104006                        EMT     C$CLP1
2311
2312  021064  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2313  021070                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 021070  104006                        EMT     C$CLP1
2314
2315  021072  004537  013560                JSR     R5,WTDRDY       ;WAIT FOR DRIVE READY
2316  021076                        89$:    CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 021076  104006                        EMT     C$CLP1
2317
2318  021100  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2319  021104                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 021104  104006                        EMT     C$CLP1
2320
2321  021106  004537  013014                JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2322  021112  000010                        RDHDR                   ;READ HEADER
2323  021114  004537  013624                JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2324  021120                        96$:    CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 021120  104006                        EMT     C$CLP1
2325
2326  021122  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2327  021126                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 021126  104006                        EMT     C$CLP1
2328
2329  021130  005037  002234                CLR     GDDAT           ;CLEAR EXPECTED
2330  021134  013737  002236  002250        MOV     BDDAT,DWORD     ;SAVE DIFFERENCE WORD
2331  021142  013737  002170  002236        MOV     E.MP,BDDAT      ;READ HEADER
2332  021150  043737  002206  002236        BIC     SECMSK,BDDAT    ;MASK OUT SECTOR BITS
2333  021156  001404                        BEQ     5$              ;BRANCH IF ON ZERO TRACK
2334
2335  021160                                ERRDF   43.,EM54,ERR3
  (3) 021160  104462                        TRAP    T$ERCODE
  (5) 021162  000053                        .WORD   43
  (5) 021164  006263                        .WORD   EM54
  (5) 021166  010110                        .WORD   ERR3
2336  021170                        5$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 021170  104006                        EMT     C$CLP1
2337
2338  021172  011377  160736        99$:    MOV     (R3),@RLDA      ;GET DIFFERENCE WORD
2339  021176  052777  000005  160730        BIS     #SIGN!MK,@RLDA  ;SET SIGN (TOWARDS SPINDLE) AND MARKER
2340  021204  004537  013014                JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2341  021210  000006                        SEEK                    ;SEEK
2342  021212  004537  013624                JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2343  021216                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 021216  104006                        EMT     C$CLP1
2344
2345  021220  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2346  021224                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 021224  104006                        EMT     C$CLP1
2347
2348  021226  004537  013560                JSR     R5,WTDRDY       ;WAIT FOR DRIVE READY
2349  021232                        87$:    CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 021232  104006                        EMT     C$CLP1
```

```
2350
2351  021234  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2352  021240                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  021240  104006                        EMT     C$CLP1
2353
2354  021242  004537  013014                JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2355  021246  000010                        RDHDR                   ;READ HEADER
2356
2357  021250  004537  013624                JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2358  021254                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  021254  104006                        EMT     C$CLP1
2359
2360  021256  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2361  021262                                ESCAPE  SEG             ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
 (3)  021262  104010                        EMT     C$ESCAPE
 (3)  021264  000064                        .WORD   10000$-.
2362
2363  021266  011337  002234                MOV     (R3),GDDAT      ;GET EXPECTED CYLINDER
2364  021272  011337  002250        8$:     MOV     (R3),DWORD      ;SET UP DIFFERENCE FOR SEEK
2365  021276  013737  002170  002236        MOV     E.MP,BDDAT      ;READ HEADER FROM RLMP
2366  021304  043737  002206  002236        BIC     SECMSK,BDDAT    ;CLEAR OUT SECTOR BITS
2367  021312  023737  002234  002236        CMP     GDDAT,BDDAT     ;DID SEEK GO TO THE RIGHT
2368  021320  001404                        BEQ     9$              ;TRACK, IF SO, GO GET NEXT
2369
2370  021322                                ERRDF   44.,EM54,ERR3
 (3)  021322  104462                        TRAP    T$ERCODE
 (5)  021324  000054                        .WORD   44
 (5)  021326  006263                        .WORD   EM54
 (5)  021330  010110                        .WORD   ERR3
2371  021332                        9$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  021332  104006                        EMT     C$CLP1
2372
2373  021334  005723                        TST     (R3)+           ;BUMP PATTERN
2374  021336  020327  002574                CMP     R3,#SKEND       ;DID WE DO ALL PATTERNS?
2375  021342  001402                        BEQ     10$             ;YES, GO TO NEXT TEST
2376  021344  000137  020766                JMP     1$              ;NO, GO BACK WITH NEXT PATTERN
2377
2378  021350                        10$:
2379
2380  021350                                ENDSEG                  ;****END OF SEGMENT****
 (3)  021350                        10000$:
 (3)  021350  104005                        EMT     C$ESEG
2381  021352                                ENDTST                  ;****END OF TEST****
 (3)  021352                        L10063:
 (3)  021352  104001                        EMT     C$ETST
2382
2383
2384                                        .SBTTL  **TEST 38** - VERIFY HEAD SELECT 0 VIA RD HDR
2385
2386  021354                                BGNTST                  ;****START OF TEST****
2387
2388                                        ;
2389
2390  021354                                STARS
 (2)                                        ;;**********.**********.**********.**********.**********
2391                                        ;CHECK THAT WE CAN SELECT HEAD SELECT ZERO.  ISSUE
```

```
                                        ;SEEK TO HEAD SELECT 0 AND VERIFY WITH READ HEADER.
021354                                  STARS
                                        ;;*****************************************************************
                                        
021354  012777  000001  160552  99$:    MOV     #MK,@RLDA       ;SET MARKER IN RLDA
021362  005037  002234                  CLR     GDDAT           ;SET EXPECTED
                                                                ;LOAD HS=0 INTO RLDA
021366                          2$:
021366  004537  013014                  JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
021372  000006                          SEEK                    ;SEEK
021374  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
021400                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
(3) 021400  104006                      EMT     C$CLP1
                                        
021402  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
021406                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
(3) 021406  104006                      EMT     C$CLP1
                                        
021410  004537  013560                  JSR     R5,WTDRDY       ;WAIT FOR DRIVE READY
021414                          89$:    CKLOOP                  ;CHECK IF /FL:LOE IS SET
(3) 021414  104006                      EMT     C$CLP1
                                        
021416  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
021422                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
(3) 021422  104006                      EMT     C$CLP1
                                        
021424  004537  013014                  JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
021430  000010                          RDHDR                   ;READ HEADER
021432  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
021436                          96$:    CKLOOP                  ;CHECK IF /FL:LOE IS SET
(3) 021436  104006                      EMT     C$CLP1
                                        
021440  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
021444                                  ESCAPE  TST             ;IF /FL:LOE SET LOOP, ELSE EXIT TST
(3) 021444  104010                      EMT     C$ESCAPE
(3) 021446  000036                      .WORD   L10064-.
                                        
021450  013737  002170  002236          MOV     E.MP,BDDAT      ;READ HEADER FOR HEAD SELECT
021456  042737  177677  002236          BIC     #177677,BDDAT   ;MASK ONLY HEAD SELECT
021464  023737  002234  002236          CMP     GDDAT,BDDAT     ;COMPARE HEAD SELECTS
021472  001404                          BEQ     5$              ;IF EQUAL CONTINUE
                                        
021474                                  ERRDF   45.,EM55,ERR4
(3) 021474  104462                      TRAP    T$ERCODE
(5) 021476  000055                      .WORD   45
(5) 021500  006322                      .WORD   EM55
(5) 021502  010162                      .WORD   ERR4
021504                          5$:
                                        
021504                          ENDTST                          ;****END OF TEST****
(3) 021504                      L10064:
(3) 021504  104001                      EMT     C$ETST
                                        
                                        
                                        .SBTTL  **TEST 39 ** - VERIFY HEAD SELECT 1 VIA RD HDR
```

```
2434  021506                          BGNTST                          ;****START OF TEST****
2435
2436
2437  021506                          STARS
 (2)                                  ;;*****************************************************************
2438                                  ;CHECK THAT WE CAN SELECT HEAD SELECT ONE.  ISSUE
2439                                  ;SEEK TO HEAD SELECT 1 AND VERIFY WITH READ HEADER.
2440  021506                          STARS
 (2)                                  ;;*****************************************************************
2441
2442
2443  021506  012777  000001  160420  99$:    MOV     #MK,@RLDA       ;SET MARKER IN RLDA
2444  021514  052777  000020  160412          BIS     #DAHS,@RLDA     ;LOAD HS=1 INTO RLDA
2445  021522  004537  013014          2$:     JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2446  021526  000006                          SEEK                    ;SEEK
2447  021530  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2448  021534                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  021534  104006                          EMT     C$CLP1
2449
2450  021536  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2451  021542                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  021542  104006                          EMT     C$CLP1
2452
2453  021544  004537  013560                  JSR     R5,WTDRDY       ;WAIT FOR DRIVE CLEAR
2454  021550                          89$:    CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  021550  104006                          EMT     C$CLP1
2455
2456  021552  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2457  021556                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  021556  104006                          EMT     C$CLP1
2458
2459  021560  004537  013014                  JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2460  021564  000010                          RDHDR                   ;READ HEADER
2461  021566  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2462  021572                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  021572  104006                          EMT     C$CLP1
2463
2464  021574  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2465  021600                                  ESCAPE  TST             ;IF /FL:LOE SET LOOP, ELSE EXIT TST
 (3)  021600  104010                          EMT     C$ESCAPE
 (3)  021602  000044                          .WORD   L10065-.
2466
2467  021604  013737  002170  002236          MOV     E.MP,BDDAT      ;READ HEADER
2468  021612  042737  177677  002236          BIC     #177677,BDDAT   ;MASK FOR H.S.
2469  021620  012737  000100  002234          MOV     #RHHS,GDDAT     ;SET EXPECTED
2470  021626  023737  002234  002236          CMP     GDDAT,BDDAT     ;CORRECT HEAD
2471  021634  001404                          BEQ     5$              ;YES, CONTINUE
2472
2473  021636                                  ERRDF   46.,EM55,ERR4
 (3)  021636  104462                          TRAP    T$ERCODE
 (5)  021640  000056                          .WORD   46
 (5)  021642  006322                          .WORD   EM55
 (5)  021644  010162                          .WORD   ERR4
2474  021646                          5$:
2475
2476  021646                          ENDTST                          ;****END OF TEST****
```

```
  (3)  021646                          L10065:
  (3)  021646  104001                          EMT     C$ETST
2477
2478
2479                                   .SBTTL  **TEST 40** - VERIFY HEAD SELECT 0 VIA GET STATUS
2480
2481  021650                           BGNTST                          ;****START OF TEST****
2482
2483  021650                           STARS
  (2)                                  ;;*****************************************************************
2484                                   ;CHECK THAT WE CAN READ BACK HEAD SELECT 0 WITH
2485                                   ;A GET STATUS FUNCTION.  SELECT H.S. 0 WITH A SEEK
2486                                   ;VERIFY WITH GET STATUS
2487  021650                           STARS
  (2)                                  ;;*****************************************************************
2488
2489  021650  012777  000001  160256           MOV     #MK,@RLDA       ;SET MARKER IN RLDA
2490                                                                   ;LOAD HS=0 INTO RLDA
2491  021656  005037  002234          2$:     CLR     GDDAT           ;SET UP EXP'D
2492  021662  004537  013014          3$:     JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2493  021666  000006                          SEEK                    ;SEEK
2494  021670  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2495  021674                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  021674  104006                          EMT     C$CLP1
2496
2497  021676  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2498  021702                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  021702  104006                          EMT     C$CLP1
2499
2500  021704  004537  013560                  JSR     R5,WTDRDY       ;WAIT FOR DRIVE READY
2501  021710                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  021710  104006                          EMT     C$CLP1
2502
2503  021712  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2504  021716                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  021716  104006                          EMT     C$CLP1
2505
2506  021720  012777  000003  160206          MOV     #GSBIT!MK,@RLDA ;SET UP FOR GET STATUS IN DA
2507  021726  004537  013014                  JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2508  021732  000004                          GSTAT                   ;GET STATUS
2509  021734  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2510  021740                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  021740  104006                          EMT     C$CLP1
2511
2512  021742  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2513  021746                                  ESCAPE  TST             ;IF /FL:LOE SET LOOP, ELSE EXIT TST
  (3)  021746  104010                          EMT     C$ESCAPE
  (3)  021750  000036                          .WORD   L10066-.
2514
2515  021752  013737  002170  002236  MOV     E.MP,BDDAT      ;READ STATUS FOR HEAD SELECT BIT
2516  021760  042737  177677  002236  BIC     #177677,BDDAT   ;LEAVE ONLY H.S. BIT
2517  021766  023737  002234  002236  CMP     GDDAT,BDDAT     ;IS HEAD SELECT CORRECT?
2518  021774  001404                  BEQ     6$              ;YES, CONTINUE
2519
2520  021776                          ERRDF   47.,EM56,ERR4
  (3)  021776  104462                  TRAP    T$ERCODE
```

```
  (5)  022000  000057                        .WORD   47
  (5)  022002  006355                        .WORD   EM56
  (5)  022004  010162                        .WORD   ERR4
2521   022006                          6$:
2522
2523   022006                          ENDTST                          ;****END OF TEST****
  (3)  022006                          L10066:
  (3)  022006  104001                        EMT     C$ETST
2524
2525
2526                                   .SBTTL  **TEST 41** - VERIFY HEAD SELECT 1 VIA GET STATUS
2527
2528   022010                          BGNTST                          ;****START OF TEST****
2529
2530   022010                          STARS
  (2)                                  ;;**************************************************************
2531                                   ;CHECK THAT WE CAN READ BACK HEAD SELECT 1 WITH A GET
2532                                   ;STATUS FUNCTION.  SELECT H.S. 1 WITH A SEEK AND VERIFIY WITH
2533                                   ;GET STATUS
2534   022010                          STARS
  (2)                                  ;;**************************************************************
2535
2536
2537   022010  012777  000001  160116          MOV     #MK,@RLDA       ;SET MARKER IN RLDA
2538   022016  052777  000020  160110          BIS     #DAHS,@RLDA     ;LOAD HS=1 INTO RLDA
2539   022024  012737  000100  002234  2$:     MOV     #STHS,GDDAT     ;SET UP EXP'D
2540   022032  004537  013014          3$:     JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2541   022036  000006                          SEEK                    ;SEEK
2542   022040  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2543   022044                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  022044  104006                          EMT     C$CLP1
2544
2545   022046  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2546   022052                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  022052  104006                          EMT     C$CLP1
2547
2548   022054  004537  013560                  JSR     R5,WTDRDY       ;WAIT FOR DRIVE READY
2549   022060                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  022060  104006                          EMT     C$CLP1
2550
2551   022062  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2552   022066                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  022066  104006                          EMT     C$CLP1
2553
2554   022070  012777  000003  160036          MOV     #GSBIT!MK,@RLDA ;SET UP FOR GET STATUS IN DA
2555   022076  004537  013014                  JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2556   022102  000004                          GSTAT                   ;GET STATUS
2557   022104  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2558   022110                                  ESCAPE  TST             ;IF /FL:LOE SET LOOP, ELSE EXIT TST
  (3)  022110  104010                          EMT     C$ESCAPE
  (3)  022112  000046                          .WORD   L10067-.
2559
2560   022114  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2561   022120                                  ESCAPE  TST             ;IF /FL:LOE SET LOOP, ELSE EXIT TST
  (3)  022120  104010                          EMT     C$ESCAPE
  (3)  022122  000036                          .WORD   L10067-.
```

```
2562
2563  022124  013737  002170  002236          MOV     E.MP,BDDAT      ;READ STATUS FOR HEAD SELECT BIT
2564  022132  042737  177677  002236          BIC     #177677,BDDAT   ;LEAVE ONLY H.S. BIT
2565  022140  023737  002234  002236          CMP     GDDAT,BDDAT     ;IS HEAD SELECT CORRECT?
2566  022146  001404                          BEQ     6$              ;YES, CONTINUE
2567
2568  022150                                  ERRDF   48.,EM56,ERR4
  (3) 022150  104462                          TRAP    T$ERCODE
  (5) 022152  000060                          .WORD   48
  (5) 022154  006355                          .WORD   EM56
  (5) 022156  010162                          .WORD   ERR4
2569  022160                          6$:
2570
2571  022160                          ENDTST                          ;****END OF TEST****
  (3) 022160                          L10067:
  (3) 022160  104001                          EMT     C$ETST
2572
2573
2574                                  .SBTTL  **TEST 42** - TEST TIME AT WHICH DIF WD GETS TRANSMITTED
2575
2576  022162                          BGNTST                          ;****START OF TEST****
2577
2578
2579  022162                          STARS
  (2)                               ;;**********************************************************************
2580                                ;VERIFY THAT THE DIFFERENCE WORD ON A SEEK IS
2581                                ;TRANSMITTED PRIOR TO CONTROLLER READY SETTING.  THIS
2582                                ;IS DONE BY SETTING A KNOWN DIFFERENCE WORD IN
2583                                ;THE RLDA ISSUING A A SEEK, WAITING FOR CONTROLLER READY
2584                                ;(BUT NOT DRIVE READY), WRITING A DIFFERENT RLDA AND WAITING
2585                                ;FOR DRIVE READY.  THE RESULTANT POSITION SHOULD BE THAT
2586                                ;OF THE FIRST RLDA ONLY.
2587  022162                        STARS
  (2)                               ;;**********************************************************************
2588
2589
2590  022162  004537  013014                  JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2591  022166  000010                          RDHDR                   ;READ HEADER
2592  022170  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2593  022174                          99$:    CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022174  104006                          EMT     C$CLP1
2594
2595  022176  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2596  022202                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022202  104006                          EMT     C$CLP1
2597
2598  022204  013737  002170  002234          MOV     E.MP,GDDAT      ;READ HEADER
2599  022212  043737  002206  002234          BIC     SECMSK,GDDAT    ;CLEAR SECTOR BITS
2600  022220  012777  000001  157706          MOV     #MK,@RLDA       ;SET MARKER IN RLDA
2601  022226  032737  000100  002234          BIT     #RHHS,GDDAT     ;TEST H.S.
2602  022234  001403                          BEQ     2$              ;IF ZERO, CONTINUE
2603  022236  052777  000020  157670          BIS     #DAHS,@RLDA     ;ONE, SET SO WE WILL REMAIN THERE
2604  022244  013737  002234  002226  2$:     MOV     GDDAT,TMP0      ;STORE HEADER
2605  022252  042737  000100  002226          BIC     #RHHS,TMP0      ;CLEAR H.S. FROM STORED WORD
2606  022260  023737  002226  002552          CMP     TMP0,HALMAX     ;WHERE ARE WE?
2607  022266  101007                          BHI     3$              ;BRANCH IF ON INNER HALF
```

```
2608  022270  052777  000004  157636          BIS     #SIGN,@RLDA     ;ON OUTER HALF, SET SEEK TO GO IN
2609  022276  063737  002550  002234          ADD     QUAMAX,GDDAT    ;SET UP EXPECTED HEADER
2610  022304  000403                          BR      4$              ;CONTINUE
2611  022306  163737  002550  002234  3$:     SUB     QUAMAX,GDDAT    ;SET UP EXPECTED HEADER
2612  022314  053777  002550  157612  4$:     BIS     QUAMAX,@RLDA    ;SET DIFFERENCE WORD IN RLDA
2613  022322  012737  000001  002230          MOV     #MK,TMP1        ;SET UP ANOTHER 'RLDA" FOR LOADING
2614  022330  032777  000020  157576          BIT     #DAHS,@RLDA     ;AFTER SEEK, TO CHANGE ONLY
2615  022336  001003                          BNE     5$              ;HEAD
2616  022340  052737  000020  002230          BIS     #DAHS,TMP1
2617  022346  004537  013014          5$:     JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2618  022352  000006                          SEEK                    ;SEEK
2619  022354  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2620  022360                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022360  104006                          EMT     C$CLP1
2621
2622
2623  022362  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2624  022366                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022366  104006                          EMT     C$CLP1
2625
2626  022370  013777  002230  157536          MOV     TMP1,@RLDA      ;SEND IN NEW DIFFERENCE WORD
2627  022376  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2628  022402                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022402  104006                          EMT     C$CLP1
2629
2630  022404  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2631  022410                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022410  104006                          EMT     C$CLP1
2632
2633  022412  004537  013560                  JSR     R5,WTDRDY       ;WAIT FOR DRIVE READY
2634  022416                          8$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022416  104006                          EMT     C$CLP1
2635
2636
2637  022420  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2638  022424                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022424  104006                          EMT     C$CLP1
2639
2640  022426  004537  013014                  JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2641  022432  000010                          RDHDR                   ;READ HEADER
2642  022434  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2643  022440                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022440  104006                          EMT     C$CLP1
2644
2645  022442  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2646  022446                                  ESCAPE  TST             ;IF /FL:LOE SET LOOP, ELSE EXIT TST
  (3) 022446  104010                          EMT     C$ESCAPE
  (3) 022450  000036                          .WORD   L10070-.
2647
2648  022452  013737  002170  002236          MOV     E.MP,BDDAT      ;READ HEADER
2649  022460  043737  002206  002236          BIC     SECMSK,BDDAT    ;CLEAR SECTOR ADDRESS
2650  022466  023737  002234  002236          CMP     GDDAT,BDDAT     ;IS HEADER CORRECT?
2651  022474  001404                          BEQ     10$             ;IF SO BRANCH
2652
2653  022476                                  ERRDF   50.,EM57,ERR4
  (3) 022476  104462                          TRAP    T$ERCODE
```

```
  (5)  022500  000062                    .WORD   50
  (5)  022502  006414                    .WORD   EM57
  (5)  022504  010162                    .WORD   ERR4
2654   022506                        10$:
2655
2656   022506                        ENDTST                          ;****END OF TEST****
  (3)  022506                        L10070:
  (3)  022506  104001                       EMT     C$ETST
2657
2658
2659                                 .SBTTL  **TEST 43** - EXTENSIVE CHECK OF HEADER CRC
2660
2661   022510                        BGNTST                          ;****START OF TEST****
2662   022510                        STARS
  (2)                                ;;**************************************************************
2663                                 ;MORE EXTENSIVE CHECK OF HEADER CRC.  WE WILL SEEK
2664                                 ;AND READ HEADERS VERIFYING HDR CRC ACROSS THE
2665                                 ;PLATTER USING THE GROWING 0, GROWING 1, SHIFTING 0 AND
2666                                 ;GROWING 0 PATTERNS FOR TRACK ADDRESSES.
2667   022510                        STARS
  (2)                                ;;**************************************************************
2668
2669
2670   022510  012703  002474                MOV     #SKLST,R3       ;GET LIST OF DIFFERENCE WORDS
2671   022514                        BGNSEG                          ;****START OF SEGMENT****
  (3)  022514  104004                       EMT     C$BSEG
2672   022516                        1$:
2673   022516  004537  013014                JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2674   022522  000010                RDHDR                           ;READ HEADER
2675   022524  004537  013624                JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2676   022530                        98$:    CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3)  022530  104006                       EMT     C$CLP1
2677
2678   022532  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2679   022536                        CKLOOP                          ;CHECK IF /FL:LOE IS SET
  (3)  022536  104006                       EMT     C$CLP1
2680
2681   022540  013737  002170  002236        MOV     E.MP,BDDAT      ;READ HEADER
2682   022546  043737  002206  002236        BIC     SECMSK,BDDAT    ;CLEAR OUT SECTOR
2683   022554  001461                       BEQ     5$              ;IF ON TRACK ZERO, H.S. ZERO, OK
2684
2685                                 ;NOT ON TRACK ZERO CALCULATE DIFFERENCE WORD AND PUT IT BACK
2686                                 ;ON ZERO.
2687
2688   022556  042737  000100  002236        BIC     #RHHS,BDDAT     ;CLEAR OUT HEAD SELECT
2689   022564  013777  002236  157342        MOV     BDDAT,@RLDA     ;PUT CYLINDER AS DIFFERENCE WORD
2690   022572  052777  000001  157334        BIS     #MK,@RLDA       ;SET MARKER BIT
2691   022600  004537  013014                JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2692   022604  000006                SEEK                            ;SEEK
2693   022606  004537  013624                JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2694   022612                        CKLOOP                          ;CHECK IF /FL:LOE IS SET
  (3)  022612  104006                       EMT     C$CLP1
2695
2696   022614  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2697   022620                        CKLOOP                          ;CHECK IF /FL:LOE IS SET
  (3)  022620  104006                       EMT     C$CLP1
```

```
2698
2699  022622  004537  013560          89$:  JSR     R5,WTDRDY       ;WAIT FOR DRIVE READY
2700  022626                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022626  104006                        EMT     C$CLP1
2701
2702  022630  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2703  022634                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022634  104006                        EMT     C$CLP1
2704
2705
2706  022636  004537  013014                JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2707  022642  000010                        RDHDR                   ;READ HEADER
2708  022644  004537  013624                JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2709  022650                          96$:  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022650  104006                        EMT     C$CLP1
2710
2711  022652  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2712  022656                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022656  104006                        EMT     C$CLP1
2713
2714  022660  005037  002234                CLR     GDDAT           ;CLEAR EXPECTED
2715  022664  013737  002236  002250        MOV     BDDAT,DWORD     ;SAVE DIFFERENCE WORD
2716  022672  013737  002170  002236        MOV     E.MP,BDDAT      ;READ HEADER
2717  022700  043737  002206  002236        BIC     SECMSK,BDDAT    ;MASK OUT SECTOR BITS
2718  022706  001404                        BEQ     5$              ;BRANCH IF ON ZERO TRACK
2719
2720  022710                                ERRDF   51.,EM54,ERR3
  (3) 022710  104462                        TRAP    T$ERCODE
  (5) 022712  000063                        .WORD   51
  (5) 022714  006263                        .WORD   EM54
  (5) 022716  010110                        .WORD   ERR3
2721  022720                          5$:   CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022720  104006                        EMT     C$CLP1
2722
2723  022722  011377  157206                MOV     (R3),@RLDA      ;GET DIFFERENCE WORD
2724  022726  052777  000005  157200        BIS     #SIGN!MK,@RLDA  ;SET SIGN (TOWARDS SPINDLE) AND MARKER
2725  022734  004537  013014                JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2726  022740  000006                        SEEK                    ;SEEK
2727  022742  004537  013624                JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2728  022746                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022746  104006                        EMT     C$CLP1
2729
2730  022750  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2731  022754                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022754  104006                        EMT     C$CLP1
2732
2733  022756  004537  013560                JSR     R5,WTDRDY       ;WAIT FOR DRIVE READY
2734  022762                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022762  104006                        EMT     C$CLP1
2735
2736
2737  022764  004537  012514                JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2738  022770                                CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 022770  104006                        EMT     C$CLP1
2739
2740  022772  004537  013014                JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
```

```
2741  022776  000010                          RDHDR                   ;READ HEADER
2742  023000  004537  013624                  JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2743  023004                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 023004  104006                          EMT     C$CLP1
2744
2745
2746  023006  004537  012514                  JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2747  023012                                  CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 023012  104006                          EMT     C$CLP1
2748
2749  023014  011337  002234                  MOV     (R3),GDDAT      ;GET EXPECTED CYLINDER
2750  023020  011337  002250          8$:     MOV     (R3),DWORD      ;SET UP DIFFERENCE FOR SEEK
2751  023024  013737  002170  002236          MOV     E.MP,BDDAT      ;READ HEADER FROM RLMP
2752  023032  043737  002206  002236          BIC     SECMSK,BDDAT    ;CLEAR OUT SECTOR BITS
2753  023040  023737  002234  002236          CMP     GDDAT,BDDAT     ;DID SEEK GO TO THE RIGHT
2754  023046  001404                          BEQ     9$              ;TRACK, IF SO, GO GET NEXT
2755
2756  023050                                  ERRDF   52.,EM54,ERR3
  (3) 023050  104462                          TRAP    T$ERCODE
  (5) 023052  000064                          .WORD   52
  (5) 023054  006263                          .WORD   EM54
  (5) 023056  010110                          .WORD   ERR3
2757  023060                          9$:     CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 023060  104006                          EMT     C$CLP1
2758
2759  023062  013737  002170  023076          MOV     E.MP,10$        ;GET HEADER WORD
2760  023070  004537  013372                  JSR     R5,SIMBCC       ;GO CALCULATE HEADER CRC
2761  023074  000020                          16.                     ;16 BITS
2762  023076  000000                  10$:    .WORD   0               ;HEADER GOES HERE
2763  023100  000000                          .WORD   0               ;START WITH ZERO CRC
2764  023102  013737  002216  023126          MOV     CALBCC,20$
2765  023110  013737  002172  023124          MOV     E.MP1,21$
2766  023116  004537  013372                  JSR     R5,SIMBCC
2767  023122  000020                          16.
2768  023124  000000                  21$:    .WORD   0
2769  023126  000000                  20$:    .WORD   0
2770  023130  013737  002216  002234          MOV     CALBCC,GDDAT    ;MOVE CALCULATED CRC TO GDDAT
2771  023136  013737  002174  002236          MOV     E.MP2,BDDAT     ;GET HEADER CRC FROM RLMP
2772  023144  023737  002234  002236          CMP     GDDAT,BDDAT     ;IS CRC CORRECT?
2773  023152  001404                          BEQ     11$             ;IF SO CONTINUE
2774
2775  023154                                  ERRDF   53.,EM42,ERR4
  (3) 023154  104462                          TRAP    T$ERCODE
  (5) 023156  000065                          .WORD   53
  (5) 023160  005704                          .WORD   EM42
  (5) 023162  010162                          .WORD   ERR4
2776  023164                          11$:    CKLOOP                  ;CHECK IF /FL:LOE IS SET
  (3) 023164  104006                          EMT     C$CLP1
2777
2778
2779  023166  005723                          TST     (R3)+           ;BUMP PATTERN
2780  023170  020327  002574                  CMP     R3,#SKEND       ;DID WE DO ALL PATTERNS?
2781  023174  001402                          BEQ     12$             ;YES, CONTINUE
2782  023176  000137  022516                  JMP     1$              ;NO, GO BACK FOR NEXT PATTERN
2783  023202                          12$:
2784
```

```
2785  023202                          ENDSEG                          ;****END OF SEGMENT****
 (3)  023202                          10000$:
 (3)  023202  104005                          EMT     C$ESEG
2786  023204                          ENDTST                          ;****END OF TEST****
 (3)  023204                          L10071:
 (3)  023204  104001                          EMT     C$ETST
2787
2788
2789                                 .SBTTL  **TEST 44** - VERIFY GET STATUS WHILE DRDY IS LOW
2790
2791  023206                         BGNTST                          ;****START OF TEST****
2792
2793  023206                         STARS
 (2)                                 ;;**************************************************************
2794                                 ;VERIFY THAT WE CAN ISSUE GET STATUS AND RECIEVE
2795                                 ;THE STATUS WORD WHILE THE DRIVE IS IN NOTION SEEKING
2796  023206                         STARS
 (2)                                 ;;**************************************************************
2797
2798
2799  023206                         1$:
2800  023206  004537  013014                 JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2801  023212  000010                 RDHDR                           ;READ HEADER
2802  023214  004537  013624                 JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2803  023220                         CKLOOP                          ;CHECK IF /FL:LOE IS SET
 (3)  023220  104006                         EMT     C$CLP1
2804
2805  023222  004537  012514                 JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2806  023226                         CKLOOP                          ;CHECK IF /FL:LOE IS SET
 (3)  023226  104006                         EMT     C$CLP1
2807
2808  023230  013737  002170  002236         MOV     E.MP,BDDAT      ;READ HEADER
2809  023236  043737  002206  002236         BIC     SECMSK,BDDAT    ;CLEAR OUT SECTOR
2810  023244  001461                         BEQ     5$              ;IF ON TRACK ZERO, H.S. ZERO, OK
2811
2812                                 ;NOT ON TRACK ZERO CALCULATE DIFFERENCE WORD AND PUT IT BACK
2813                                 ;ON ZERO.
2814
2815  023246  042737  000100  002236         BIC     #RHHS,BDDAT     ;CLEAR OUT HEAD SELECT
2816  023254  013777  002236  156652         MOV     BDDAT,@RLDA     ;PUT CYLINDER AS DIFFERENCE WORD
2817  023262  052777  000001  156644         BIS     #MK,@RLDA       ;SET MARKER BIT
2818  023270  004537  013014                 JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2819  023274  000006                 SEEK                            ;SEEK
2820  023276  004537  013624                 JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2821  023302                         CKLOOP                          ;CHECK IF /FL:LOE IS SET
 (3)  023302  104006                         EMT     C$CLP1
2822
2823  023304  004537  012514                 JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2824  023310                         CKLOOP                          ;CHECK IF /FL:LOE IS SET
 (3)  023310  104006                         EMT     C$CLP1
2825
2826  023312  004537  013560                 JSR     R5,WTDRDY       ;WAIT FOR DRIVE READY
2827  023316                         CKLOOP                          ;CHECK IF /FL:LOE IS SET
 (3)  023316  104006                         EMT     C$CLP1
2828
2829  023320  004537  012514                 JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
```

```
2830  023324                                    CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  023324  104006                            EMT     C$CLP1
2831
2832
2833  023326  004537  013014                    JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2834  023332  000010                            RDHDR                   ;READ HEADER
2835  023334  004537  013624                    JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2836  023340                                    CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  023340  104006                            EMT     C$CLP1
2837
2838  023342  004537  012514                    JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2839  023346                                    CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  023346  104006                            EMT     C$CLP1
2840
2841  023350  005037  002234                    CLR     GDDAT           ;CLEAR EXPECTED
2842  023354  013737  002236  002250            MOV     BDDAT,DWORD     ;SAVE DIFFERENCE WORD
2843  023362  013737  002170  002236            MOV     E.MP,BDDAT      ;READ HEADER
2844  023370  043737  002206  002236            BIC     SECMSK,BDDAT    ;MASK OUT SECTOR BITS
2845  023376  001404                            BEQ     5$              ;BRANCH IF ON ZERO TRACK
2846
2847  023400                                    ERRDF   54.,EM54,ERR3
 (3)  023400  104462                            TRAP    T$ERCODE
 (5)  023402  000066                            .WORD   54
 (5)  023404  006263                            .WORD   EM54
 (5)  023406  010110                            .WORD   ERR3
2848  023410                              5$:   CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  023410  104006                            EMT     C$CLP1
2849
2850  023412  012777  077601  156514            MOV     #77601,@RLDA    ;GET DIFFERENCE WORD
2851  023420  052777  000005  156506            BIS     #SIGN!MK,@RLDA  ;SET SIGN (TOWARDS SPINDLE) AND MARKER
2852  023426  004537  013014                    JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2853  023432  000006                            SEEK                    ;SEEK
2854  023434  004537  013624                    JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2855  023440                                    CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  023440  104006                            EMT     C$CLP1
2856
2857
2858  023442  004537  012514                    JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2859  023446                                    CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  023446  104006                            EMT     C$CLP1
2860  023450  012777  000003  156456            MOV     #MK!GSBIT,@RLDA
2861  023456  004537  013014                    JSR     R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
2862  023462  000004                            GSTAT
2863  023464  004537  013624                    JSR     R5,WTCRDY       ;WAIT FOR CONTROLLER READY HIGH
2864  023470                                    CKLOOP                  ;CHECK IF /FL:LOE IS SET
 (3)  023470  104006                            EMT     C$CLP1
2865  023472  004537  012514                    JSR     R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
2866
2867  023476                        ENDTST                              ;****END OF TEST****
 (3)  023476                        L10072:
 (3)  023476  104001                            EMT     C$ETST
2868
2869  023500                        BGNMOD  HRDPRM
2870
2871  023500                                    BGNHRD
 (3)  023500  000025                            .WORD L10073-L$HARD/2
```

```
2872
2873  023502
 (4)  023502  004130          GPRML   CNTMSG,CNT,1,YES
 (4)  023504  023570          .WORD   T$CODE
 (4)  023506  000001          .WORD   CNTMSG
                              .WORD   1
2874  023510                  GPRMA   CSRMSG,CSR,0,160000,177776,YES
 (4)  023510  000031          .WORD   T$CODE
 (4)  023512  023554          .WORD   CSRMSG
 (4)  023514  160000          .WORD   T$LOLIM
 (4)  023516  177776          .WORD   T$HILIM
2875  023520                  GPRMA   VECMSG,VECT,0,0,776,YES
 (4)  023520  001031          .WORD   T$CODE
 (4)  023522  023606          .WORD   VECMSG
 (4)  023524  000000          .WORD   T$LOLIM
 (4)  023526  000776          .WORD   T$HILIM
2876  023530                  GPRMD   BRMSG,PRIOR,0,340,0,7,YES
 (4)  023530  002032          .WORD   T$CODE
 (4)  023532  023575          .WORD   BRMSG
 (4)  023534  000340          .WORD   340
 (4)  023536  000000          .WORD   T$LOLIM
 (4)  023540  000007          .WORD   T$HILIM
2877  023542                  GPRMD   DRMSG,DRBT,0,03400,0,7,YES
 (4)  023542  003032          .WORD   T$CODE
 (4)  023544  023615          .WORD   DRMSG
 (4)  023546  003400          .WORD   03400
 (4)  023550  000000          .WORD   T$LOLIM
 (4)  023552  000007          .WORD   T$HILIM
2878
2879  023554                  ENDHRD
 (2)                          .EVEN
 (3)  023554          L10073:
2880
2881  023554  052502  020123  042101  CSRMSG: .ASCIZ  /BUS ADDRESS/
       023562  051104  051505  000123
2882  023570  046122  030461     000  CNTMSG: .ASCIZ  /RL11/
2883  023575     102  020122  042514  BRMSG:  .ASCIZ  /BR LEVEL/
       023602  042526  000114
2884  023606  042526  052103  051117  VECMSG: .ASCIZ  /VECTOR/
       023614     000
2885  023615     104  044522  042526  DRMSG:  .ASCIZ  /DRIVE/
       023622     000
2886          023624          .EVEN
2887
2888  023624                  ENDMOD
2889
2890
2891
2892  023624          BGNMOD  SFTPRM
2893
2894  023624                  BGNSFT
 (3)  023624  000014          .WORD L10074-L$SOFT/2
2895  023626                  GPRML   DMSG,DLT,1,YES
 (4)  023626  000130          .WORD   T$CODE
 (4)  023630  023656          .WORD   DMSG
 (4)  023632  000001          .WORD   1
2896  023634                  XFERF   1$
```

```
  (5)  023634  006044                          .WORD   T$CODE
2897   023636                                  GPRMD   EMSG,ELT,0,177777,0,177777,YES
  (4)  023636  001032                          .WORD   T$CODE
  (4)  023640  023713                          .WORD   EMSG
  (4)  023642  177777                          .WORD   177777
  (4)  023644  000000                          .WORD   T$LOLIM
  (4)  023646  177777                          .WORD   T$HILIM
2898   023650                          1$:     GPRML   SMSG,SIZE,1,YES
  (4)  023650  002130                          .WORD   T$CODE
  (4)  023652  023702                          .WORD   SMSG
  (4)  023654  000001                          .WORD   1
2899   023656                                  ENDSFT
  (2)                                          .EVEN
  (3)  023656                          L10074:
2900
2904
2905   023656  051104  050117  047440  DMSG:   .ASCIZ  /DROP ON ERROR LIMIT/
2906   023702  052501  047524  044523  SMSG:   .ASCIZ  /AUTOSIZE/
2907   023713     105  051122  051117  EMSG:   .ASCIZ  /ERROR LIMIT/
2908
2912
2913           023730                 .EVEN
2914
2915   023730                                  ENDMOD
2916
2917
2918           024114                 .=24114
2919
2920                                  ;AREA RESERVED AS PATCH AREA FOR DIAGNOSTICS.
2921                                  ;.=24114 WAS SELECTED AS ''LASTAD'' TO PROVIDE APT TO LSI-11 COMPATIBILITY.
2922                                  ;BIT 7 OF ''LASTAD'' MUST BE CLEARED TO ACHIEVE A VALID MAILBOX ADDRESS
2923                                  ;WHEN RUNNING ON THE LSI-11 UNDER APT.
2924
2925   024114                         LASTAD
  (2)                                          .EVEN
  (3)  024114                          L$LAST::
2926
```

```
 2928
13799  054710  000000                    .SBTTL  DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP
13800  054712  000000                    .WORD   0               ;SPACE FOR USER POOL POINTER
13801  054714  000000                    .WORD   0               ;SIZE
13802  054716  000000                    .WORD   0               ;CHECKSUM (NOT CURRENTLY USED)
13803          054722                    .WORD   0               ;SIZE OF H.W. PTAB. ALLOCATION
13804          000200                    END.SUPV=.+2
                                         .END 200
```

```
ABOFLA  024440 G
ABOPAS  024356 G
ABO.FM  026720
ADDCOD  012424 G
AFREG   004501
AFSI    024146 G
AFTER   013324
ALLOC   045060
APT.ER  026050
ARLBA   004436
ARLCS   004431
ARLDA   004444
ARLMP   004452
ASSEMB= 000010
A$AAV   030716
A$AAW   030732
A$AAX   030744
A$AAY   030752
A$AAZ   030766
A$ABA   030776
BATEST  014410
BA16  = 000020
BA17  = 000040
BCCFBK  002214
BCSR    002140
BDDAT   002236
BEFORE  013272
BEGPAT  002264
BEREG   004460
BGN.SU= 024114
BINMSG  043370
BIT0  = 000001 G
BIT00 = 000001 G
BIT01 = 000002 G
BIT02 = 000004 G
BIT03 = 000010 G
BIT04 = 000020 G
BIT05 = 000040 G
BIT06 = 000100 G
BIT07 = 000200 G
BIT08 = 000400 G
BIT09 = 001000 G
BIT1  = 000002 G
BIT10 = 002000 G
BIT11 = 004000 G
BIT12 = 010000 G
BIT13 = 020000 G
BIT14 = 040000 G
BIT15 = 100000 G
BIT2  = 000004 G
BIT3  = 000010 G
BIT4  = 000020 G
BIT5  = 000040 G
BIT6  = 000100 G
BIT7  = 000200 G
BIT8  = 000400 G
BIT9  = 001000 G
BLD.HW  031602
BLOCK   047214
BPRIOR  002142
BRMSG   023575
BVEC    002144
B$AAB   033204
B$AAF   033116
B.BA    002152
B.CS    002150
B.DA    002154
B.MP    002156
CALBCC  002216
CALLPC= 000022
CALLPS= 000024
CALLSP= 000026
CALLTC= 000030
CAL.CL  051602
CAL.TI  051640 G
CHERR   012514
CHKLUP  033220
CHKSTR  045422
CHKTTY  043510
CHK.MA  031360
CHK.PC  036510
CHK.SW  025550
CHRCNT  044742
CH.FLA  031066
CH.PAS  031104
CKERLT  012430
CLEAR.  032502
CLKACC  024354 G
CLKBFR  051604
CLKCNT  024352 G
CLKJUM  052210 G
CLKRES  053212 G
CLKSER  053346 G
CLKSON  024412 G
CLK.SE  031162
CLNCOD  012352 G
CLR.MA  031436
CNT   = 000010
CNTMSG  023570
CNVT    047660
COMMAN  024164 G
COMMTA  047474
COMP    003702
CONT    011724
CONTCL  053272 G
CONTIN  011576
CRDY  = 000200
CRLF    043572
CRTIM   004522
CSEND   002674
CSPAT   002576
CSR   = 000000
CSRMSG  023554
CSTEST  014270
CURR.S  024122 G
CURR.T  024124 G
CYLMSK  002242
C$AAD   036462
C$AAE   036474
C$AAK   037472
C$AAL   037636
C$ABRT= 000021
C$ADR = 000020
C$AU  = 000054
C$BRK = 000022
C$BSEG= 000004
C$BSUB= 000002
C$BUFF= 000030
C$CEFG= 000046
C$CLEA= 000012
C$CLP1= 000006
C$CVEC= 000036
C$DCLN= 000044
C$DODU= 000053
C$DRPT= 000024
C$DU  = 000055
C$EDIT= 000002
C$ERDF= 000002
C$ERHR= 000003
C$ERSF= 000001
C$ERSO= 000004
C$ESCA= 000010
C$ESEG= 000005
C$ESUB= 000003
C$ETST= 000001
C$EXIT= 000032
C$GMAN= 000043
C$GPHR= 000042
C$GPRI= 000040
C$GTIM= 000052
C$INIT= 000011
C$INLP= 000020
C$KWOF= 000035
C$KWON= 000034
C$LOOP- 000100
C$MANI= 000051
C$MSG = 000023
C$PNTB= 000014
C$PNTF= 000017
C$PNTS= 000016
C$PNTX= 000015
C$POIN= 000040
C$QIO = 000377
C$RDBU= 000007
C$REFG= 000050
C$REQT= 000045
C$RESE= 000033
C$REVI= 000002
C$RPT = 000025
C$SEFG= 000047
C$SPRI= 000041
C$SVEC= 000037
C$TPRI= 000013
C$UNBU= 000031
C$WTM = 000026
C$WTU = 000027
DAHS  = 000020
DATEST  014512
DCKMES  003663
DECMSG  043404
DEMES   003631
DERFLG  002160
DERR  = 040000
DEV.CO  024126 G
DIAGMC= 000000
DIAG.T  024446 G
DLT   = 000000
DLTMES  003670
DMSG    023656
DPDVD   054056 G
DPMUL   053744 G
DRBT  = 000006
DRDY  = 000001
DRIVE   002146
DRMSG   023615
DROP    011354
DRPCOD  012420 G
DRST  = 000010
DRTIM   004547
DSPCOD  011362 G
DS0   = 000000
DS1   = 000400
DS2   = 001000
DS3   = 001400
DUNIT.  024362 G
DVC.FT  037442
DWORD   002250
D$AAG   040346
D$AAH   040364
D$AAI   043132
D$AAJ   043136
D$AAK   043154
D$AAL   043172
D$AAM   043202
EF.CON= 000036 G
EF.NEW= 000035 G
EF.PWR= 000034 G
EF.RES= 000037 G
EF.STA= 000040 G
EF01  = 000001 G
EF02  = 000002 G
EF03  = 000003 G
EF04  = 000004 G
EF05  = 000005 G
EF06  = 000006 G
EF07  = 000007 G
EF08  = 000010 G
EF09  = 000011 G
EF10  = 000012 G
EF11  = 000013 G
EF12  = 000014 G
EF13  = 000015 G
EF14  = 000016 G
EF15  = 000017 G
EF16  = 000020 G
ELT   = 000002
EMSG    023713
EMT.TR  024444 G
EM1     004575
EM101   007560
EM102   007625
EM11    005046
EM13    005107
EM14    005141
EM15    005167
EM16    005215
EM17    005243
EM2     004622
EM3     004647
EM30    005276
EM30A   005335
EM32    005375
EM33    005431
EM34    005476
EM37    005553
EM4     004674
EM41    005613
EM42    005704
EM43    005742
EM44    006041
EM45    006074
EM46    006127
EM47    006162
EM5     004721
EM52    006213
```

```
EM54    006263          E.MP1   002172          GLBDAT  002122 G        INTFOR  037644          L$DTP   002040 G
EM55    006322          E.MP2   002174          GLBEQA  002122 G        INTSRV  013552          L$DU    012420 G
EM56    006355          FILL    044240          GLBERR  010016 G        INVAL.  030452          L$DUT   002076 G
EM57    006414          FILL.C  000204 G        GLBSUB  012430 G        INVINT  037502          L$DVTY  002114 G
EM6     004772          FIRST   002240          GLBTXT  003570 G        INV.SW  025504          L$EF    002056 G
EM61    006515          FIX     013260          GODRVR= 000202          IN.SUF  032454          L$EFLG  002034 G
EM62    006576          FLAGS   024160 G        GSBIT = 000002          I$AU  = 000041          L$EXP1  002042 G
EM63    006661          FLAGS1  024162 G        GSTAT = 000004          I$CLN = 000041          L$EXP2  002044 G
EM64    006742          FLAGTA  047412          GSTINT  004372          I$DU  = 000041          L$EXP3  002046 G
EM65    007025          FLAG.I  031146          GSTMES  004333          I$HRD = 000041          L$HARD  023502 G
EM66    007106          FLA.SE  047360          G$EXCP= 000400          I$INIT= 000041          L$HPCP  002016 G
EM67    007171          FLG.MA  031106          G$HILI= 000002          I$MOD = 000041          L$HPTP  002022 G
EM7     005020          FNDFNC  013232          G$LOLI= 000001          I$MSG = 000041          L$HW    011340 G
EM70    007226          FORM.T  040010          G$NO  = 000000          I$PWR = 000041          L$ICP   002104 G
EM71    007263          FREE    045316          G$OFFS= 000400          I$RPT = 000041          L$INIT  011514 G
EM72    007320          FRMT1   010672          G$OFSI= 000376          I$SEG = 000041          L$LADP  002026 G
EM73    007353          FRMT11  011164          G$PRMA= 000001          I$SFT = 000041          L$LAST  024114 G
EM74    007406          FRMT12  011225          G$PRMD= 000002          I$SRV = 000041          L$MREV  002050 G
EM75    007440          FRMT13  011304          G$PRML= 000000          I$SUB = 000041          L$NAME  002000 G
EM76    007472          FRMT2   010732          G$RADA= 000140          I$TST = 000041          L$REPP  002066 G
EM77    007525          FRMT2A  010751          G$RADB= 000000          J$JMP = 000167          L$REV   002010 G
END     012306          FRMT2B  010764          G$RADD= 000040          KBPTR   024224 G        L$SOFT  023626 G
ENDPAT  002472          FRMT3   011013          G$RADF= 000200          KBUF    024226 G        L$SPC   002062 G
END.OF  032470          FRMT4   011020          G$RADL= 000120          LDCSR   002204          L$SPCP  002020 G
END.SU= 054722          FRMT5   011056          G$RADO= 000020          LDFUNC  013014          L$SPTP  002024 G
ENVIRO  024166 G        FRMT6   011127          G$RADT= 000100          LF      003700          L$STA   002030 G
EOP.CH  053370 G        FRMT99  011053          G$XFER= 000004          LINE.F  024442 G        L$SW    011354 G
EOP.FM  026734          F$AU  = 000015          G$YES = 000010          LINE1   010342          L$TIML  002014 G
EOP.IN  031100          F$BGN = 000040          HALMAX  002552          LINE2   010376          L$TIMU  002054 G
ERCOUN  002700          F$CLEA= 000007          HCORED  030656          LINE3   010620          L$TIM1  002052 G
ERPOIN  002676          F$DU  = 000016          HCOREQ  030566          LOAD.F  031102          L$TSTI  002100 G
ERR   = 100000          F$END = 000041          HCORET  024402 G        LOGMSG  043412          L$UNIT  002012 G
ERRFOR  037714          F$HARD= 000004          HCRCME  003650          LPBFR   024222 G        L.CLK.  030512
ERRHAN  036514          F$HW  = 000013          HC.ADR  024152 G        LPCNTR  024220 G        L10000  010032
ERRVEC  002212          F$INIT= 000006          HC.DEF  024144 G        LPT.AD  030544          L10001  010044
ERR.HR  037452          F$JMP = 000050          HC.DIA  024142 G        LPT.RE  030540          L10002  010106
ERR.NU  024116 G        F$MOD = 000000          HDRBUF  003070          LSI.RE  030534          L10003  010160
ERR.SF  037456          F$MSG = 000011          HDRLST  013234          LUP     051506          L10004  010226
ERR0    010016 G        F$PWR = 000017          HERTZ.  030526          LUP.AD  036512          L10005  010240
ERR1    010034 G        F$RPT = 000012          HNFMES  003656          L$APT   002036 G        L10006  010302
ERR1FO  040000          F$SEG = 000003          HOLDSP= 000020          L$AU    012424 G        L10007  010340
ERR2    010046 G        F$SOFT= 000005          HPTCOD  011336 G        L$AUT   002074 G        L10010  011352
ERR3    010110 G        F$SRV = 000010          HRDPRM  023500 G        L$CCP   002106 G        L10011  011362
ERR4    010162 G        F$SUB = 000002          HW.ADR  024150 G        L$CLEA  012352 G        L10012  012350
ERR5    010230 G        F$SW  = 000014          H$AAB   050206          L$CO    002032 G        L10013  012416
ERR6    010242 G        F$TEST= 000001          ININIT  024372 G        L$DEPO  002011 G        L10014  012422
ERR7    010304 G        GARBAG  044744          INITCO  011514 G        L$DESC  002102 G        L10015  012426
ESC.PC  036506          GDDAT   002234          INITIA  043420          L$DEVP  002064 G        L10016  013556
EV.COU  024120 G        GETCHR  043450          INIT.M  031504          L$DISP  011364 G        L10017  013774
E.BA    002164          GETCMN  047034          INIT.R  024206 G        L$DR    002112 G        L10020  014070
E.CS    002162          GETPAR  040526          INPUTA  044346          L$DRCT  002070 G        L10021  014164
E.DA    002166          GETSWI  046030          INTEN = 000100          L$DRS   002072 G        L10022  014260
E.MP    002170          GET.TW  045600          INTFLG  002202          L$DRST  002112 G        L10023  014400
```

```
L10024  014502          MIN.IN  024172 G        PRI02 = 000100 G        SPTCOD  011352 G        TY.UNI  032474
L10025  014570          MIN.US  024174 G        PRI03 = 000140 G        SPV.SE  000400          T$ARGC= 000001
L10026  014714          MK    = 000001          PRI04 = 000200 G        START   011614          T$CODE= 002130
L10027  015040          MODR    053656 G        PRI05 = 000240 G        STARTC  053266 G        T$ERCO= 000062
L10030  015144          MSCRLF  003675          PRI06 = 000300 G        START1  011556          T$ERRN= 000066
L10031  015244          MSG.AD  024140 G        PRI07 = 000340 G        STHS  = 000100          T$EXCP= 000000
L10032  015334          MSG.TY  024114 G        PRNTST  044610          STRCHR  044300          T$FLAG= 000040
L10033  015434          MUL     053612 G        PRO.CM  031060          STRT.T  031064          T$HILI= 177777
L10034  015544          MXSEC1  002244          PTAB.S  024400 G        ST.SET  025716          T$LOLI= 000000
L10035  015616          NEWPRI  053336 G        PUTCHR  043424          SUNIT.  031070          T$LSYM= 010000
L10036  015654          NEXTAR  047576          PWRFLG  002122          SUPERV  026752          T$NEST= 177777
L10037  016000          NODRY   003606          PWR.FA  054550 G        SUPFLA  024360 G        T$NSK0= 000000
L10040  016140          NOOP0 = 000000          PWR.FL  024204 G        SUPV.T  024532 G        T$NSK1= 000005
L10041  016300          NOOP7 = 000016          PWR.MS  054676          SUP.PR  025470          T$SAVL= 177777
L10042  016504          NOPINT  004014          PWR.SA  054672          SVCGBL= 000000          T$SEGL= 177777
L10043  016534          NOPMES  003763          PWR.UP  054674          SVCHAN  033406          T$SEK0= 010000
L10044  016740          NOPWR   011536          P.CLK.  030520          SVCINS= 000000          T$SUBN= 000000
L10045  017024          NORES   003570          QUAMAX  002550          SVCSUB= 177777          T$TAGL= 177777
L10046  017170          NO.CLK  030502          RDHDR = 000010          SVCTAG= 000000          T$TAGN= 010075
L10047  017220          NO.FLA  047372          READ  = 000014          SVCTST= 177777          T$TEMP= 000000
L10050  017372          NO.LPT  044710          READ.P  051610 G        SVHD    002254          T$TEST= 000054
L10051  017460          NO.PTA  030706          REGBAC  054300 G        SWCHAN  030700          T$TSTM= 177777
L10052  017606          NR    = 000000          REGSAV  054264 G        SWITCH  047552          T$TSTS= 000001
L10053  017630          NUMBIN  040034          REQN.P  024170 G        SW.ADR  024154 G        T$$AU = 010015
L10054  017710          NUM.LA  040202          REQN.T  031062          SW.PTA  030664          T$$CLE= 010013
L10055  020054          NUM.NO  024156 G        REST    011654          SYS.FT  037432          T$$DU = 010014
L10056  020212          NUM.UN  024564          RESTMS  012776          S$LSYM= 010000          T$$HAR= 010073
L10057  020530          NUNITS  033172          RE.SET  025652          TEMP2   002220          T$$HW = 010010
L10060  020566          NXM   = 020000          RHDINT  004207          TEMP3   002222          T$$INI= 010012
L10061  020632          NXMMES  003636          RHDMES  004147          TEMP4   002224          T$$MSG= 010007
L10062  020756          NXT     011606          RHHS  = 000100          TERMI   051576          T$$SEG= 010000
L10063  021352          NXTFOR  047652          RLBA    002132          TERMLI  047400          T$$SOF= 010074
L10064  021504          OCTMSG  043376          RLCS    002130          TERMTA  043362          T$$SRV= 010016
L10065  021646          OKHDR   013244          RLDA    002134          TEST.M  031020          T$$SW = 010011
L10066  022006          OPI   = 002000          RLMP    002136          TIMFLG  024350 G        T$$TES= 010072
L10067  022160          OPIERR  003710          RSTACK  053540 G        TIM.CO  024202 G        T.CNTL  002260
L10070  022506          OPIMES  003643          SAVEDO= 026050          TIM.OP  040006          T.SIZE  011360
L10071  023204          O$APTS= 000000          SEARCH  045546          TMPFNC  002262          T1      013702 G
L10072  023476          O$AU  = 000001          SECMSK  002206          TMP0    002226          T10     015042 G
L10073  023554          O$BGNR= 000000          SEEK  = 000006          TMP1    002230          T11     015146 G
L10074  023656          O$BGNS= 000001          SEGSTA  024414 G        TMP2    002232          T12     015246 G
MAJ.IN  024176 G        O$DU  = 000001          SEKINT  004301          TOO.MA  043342          T13     015336 G
MAJ.LO  051606          O$GNSW= 000001          SEKMES  004250          TRPFLG  002200          T14     015436 G
MAJ.US  024200 G        O$POIN= 000001          SET.MA  031272          TRPHAN  013544          T15     015546 G
MAN.TI  001244          PARSES  047106          SFTPRM  023624 G        TST.AB  033330          T16     015620 G
MAP16   054314 G        PAR.LA  043074          SHIFT   054376 G        TST.TO  025532          T17     015656 G
MASK.B  033216          PASS.C  024130 G        SIGN  = 000004          TYPEC   043736          T18     016002 G
MASK.W  033214          PFLG    002176          SIMBCC  013372          TYPEPC  037632          T19     016142 G
MAXCYL  002252          PRINTC  044720          SIZE  - 000004          TYPFLA  047254          T2      013776 G
MAXSEC  002246          PRINTF  050226          SKEND   002574          TYPLIN  043634          T20     016302 G
MDHEDR  002000 G        PRIOR = 000004          SKLST   002474          TYPNUM  043216          T21     016506 G
MEM.SI  030554          PRI00 = 000000 G        SMSG    023702          TYPSTR  043654          T22     016536 G
MERLMT  011356          PRI01 = 000040 G        SPEC.U  031006          TYP.ER  037462          T23     016742 G
```

```
T24     017026 G        T38     021354 G        UNI.MA  031010          WTCRDY  013624          XXDP.D  030464
T25     017172 G        T39     021506 G        USER.P  024374 G        WTDRDY  013560          XXX     011636
T26     017222 G        T4      014166 G        USER.T  024376 G        XEQDIA  053424 G        X$ALWA= 000000
T27     017374 G        T40     021650 G        UUT     002124          XEQSUB  053412 G        X$FALS= 000040
T28     017462 G        T41     022010 G        VALID.  024634          XEQ.CL  033134          X$OFFS= 000400
T29     017610 G        T42     022162 G        VAL.LA  025454          XEQ.CM  030444          X$TRUE= 000020
T3      014072 G        T43     022510 G        VAL.SW  031120          XEQ.IN  032616          $BREG   031160
T30     017632 G        T44     023206 G        VECMSG  023606          XEQ.LA  026706          $ENDAD  053376 G
T31     017712 G        T5      014262 G        VECT  = 000002          XEQ.OP  032710          $SAV2   054442 G
T32     020056 G        T6      014402 G        WCKINT  004106          XEQ.PR  026110          $SAV3   054456 G
T33     020214 G        T7      014504 G        WCKMES  004046          XEQ.TE  032754          $SAV4   054474 G
T34     020532 G        T8      014572 G        WHY     002256          XPOLY   002210          $SAV5   054514 G
T35     020570 G        T9      014716 G        WIDTH   040402          XTIME   052276 G        .     = 054720
T36     020634 G        UNITST  002126          WRCHK = 000002          XTIMEN  053122
T37     020760 G        UNIT.D  024132 G        WRITE = 000012          XTIMST  052320

. ABS.  054720     000


 ERRORS DETECTED:  0

 DSKZ:CZRLAB,DSKZ:CZRLAB=CZRLAB/ML,CZRLAB.P11,CZRLAB.SUP
 RUN-TIME: 41 39 1 SECONDS
 RUN-TIME RATIO: 251/82=3.0
 CORE USED:  16K  (31 PAGES)
```