IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDMH-B-D

PRODUCT NAME: DMC11 FREE RUNNING TESTS

DATE: MAY 1977

MAINTAINER: DIAGNOSTICS

AUTHOR: FAY BASHAW

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Digital.

1. ABSTRACT

The function of the DMC11 diagnostics is to verify that the option operates according to specifications. The diagnostics verfiy that there are no malfunctions and the all operations of the DMC11 are correct in its environment.

Parameters must be set up to alert the diagnostics to the DMC11 configuration. These parameters are contained in the STATUS TABLE and are generated in two ways: 1) Manual Input - the operator answers questions. 2) Autosizing - the program determines the parameters automatically.

DZDMH tests the DMC11-AR and DMC11-AL micro-processors (M8200-YA and M8200-YB), or the KMC11 micro-processor (M8204). Free running tests are performed. A line unit (M8201 or M8202) must be installed. DZDMH can be used as a heat test diagnostic by manufacturing.

Currently there are five off line diagnostics that are to be run in sequence to insure that if an error should occur it will be detected at an early stage.

NOTE: Additional diagnostics may be added in the future.

The five diagnostics are:

1. DZDMC [REV] Basic W/R and Micro-processor tests
2. DZDME [REV] DDCMP Line unit tests
3. DZDMF [REV] BITSTUFF Line unit tests
4. DZDMG [REV] CROM and Jump tests
5. DZDMH [REV] Free-running tests (Heat test tape)

2. REQUIREMENTS

2.1 EQUIPMENT

Any PDP11 family CPU (except an LSI-11) with minimum 8k memory
ASR 33 (or equilivalent)
DMC11-AR with DMC11-DA or DMC11-FA or
DMC11-AL with DMC11-MA or DMC11-MD

2.2 STOPAGE

Program will use all 8K of memory except where ABL and BOOTSTRAP LOADER reside. Locations 1500 thru 1640; contain the "STATUS TABLE" information which is generated at start of diagnostics by manual input (questions) or automatically (auto-sizing). This area is an overlay area and should not be altered by the operator.

3. LOADING PROCEEDURE

3.1 METHOD

All programs are in absolute format and are loaded using the ABSOLUTE LOADER. NOTE: if the diagnostics are on a media such as DISK ,MAGTAPE,DECTAPE, or CASSETTE: follow instructions for the monitor which has been provided on that specific media.

ABSOLUTE LOADER starting address *500

MEMORY	* SIZE
4k	17
8k	37
12k	57
16k	77
20k	117
24k	137
28k	157

3.1.1 Place address of ABS loader into switch register.
(also place 'HALT' SW up)

3.1.2 Depress 'LOAD ADDRESS' key on console and release.

3.1.3 Depress 'START KEY' on console and release (program should now be loading into CPU)

4. STARTING PROCEEDURE

a. Set switch register to 000200
b. Depress 'LOAD ADDRESS' key and release
c. Set SWR to zero for 'AUTO SIZING' or SWR bit0=1 for manual input (questions) or SWR bit7=1 to use existing parameters set up by a previous start or a previously run DMC11 diagnostic.
d. Depress 'START KEY' and release. The program will type Maindec Name and program name (if this was the first start up of the program) and also the following:

```
                 MAP OF DMC11 STATUS
                 -------------------

         PC     CSR     STAT1    STAT2    STAT3
         --     ---     -----    -----    -----

      001500  160010  145310   177777   000000
      001510  160020  145320   177777   000000
```

The program will type 'R' and proceed to run the diagnostic. The above is only an example. This would indicate the status table starting at add. 1500 in the program. In this example the table contains the information and status of two DMC11"S. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. For information of status table see section 8.4 for help.

If the diagnostic was started with SW00=1 indicating manual parameter input then the following shows an example of the questions asked and some example answers:

```
HOW MANY DMC11'S TO BE TESTED?1

01
CSR ADDRESS?160010
VECTOR ADDRESS?310
BR PRIORITY LEVEL?  (4,5,6,7)?5
DOES MICRO-PROCESSOR HAVE CRAM?  (Y OR N)N
WHICH LINE UNIT?  IF NONE TYPE "N",  IF  M8201  TYPE  "1",  IF
M9202 TYPE "2"?1
IS THE LOOP BACK CONNECTOR ON?Y
SWITCH PAC#1 (DDCMP LINE#)?377
SWITCH PAC#2 (BM873 BOOT ADD)?377
```

Following the questions the status map is printed out as described above, the information in the map reflects the answers to the questions. If the diagnostic was started with SW00=0 and SW07=0 (AUTO-SIZING) then no questions are asked and only the status-map is printed out. If AUTO-SIZING is used the status information must be verified to be correct (match the hardware), if it does not match the hardware the diagnostic must be restarted with SW00=1 and the questions answered.

4.1 CONTROL SWITCH SETTINGS

```
SW 15 Set:  Halt on error
SW 14 Set:  Loop on current tes:
SW 13 Set:  Inhibit error print out
SW 12 Set:  Inhibit type out/abell on error.
SW 11 Set:  Inhibit iterations.  (quick pass)
SW 10 Set:  Escape to next test on error
SW 09 Set:  Loop with current data
SW 08 Set:  Catch error and loop on it
SW 07 Set:  Use previous status table.
SW 06 Set:  Halt in    ROMCLK    routine    before    clocking
            micro-processor
SW 05 Set:  Reserved
SW 04 Set:  Reserved
SW 03 Set:  Reselect DMC11's desired active
SW 02 Set:  Lock on selected test
SW 01 Set:  Restart program at selected test
SW 00 Set:  Build new status table from questions.  (If SW07=0
              and  SW00=0  a  new  status  table  is  built by
              auto-sizing)
```

Switch 06 and 08-15 are dynamic and can be changed as needed while the diagnostic is running. Switches 00-03 and switch 07 are static, and are used only on starting or restarting the diagnostic.

4.1.2 SWITCH PEGISTER OPTIONS (at start up)

SW 01 RESTART PROGRAM AT SELECTED TEST. It is strongly suggested that at least one pass has been made before trying to select a test, the reason being is that the program has to clear areas and set up parameters. When this switch is used the diagnostic will ask TEST NO.? Answer by typing the number of the test desired and carrige return to begin execution at the selected test.

SW 02 LOCK ON SELECTED TEST. This switch when used with SW01 will cause the program to constantly loop on the selected test. Hitting any key on the console will let it advance to the next test and loop until a key is hit again. If SW02=0 when SW01 is used. The program will begin at the selected test and continue normal operations.

SW 03 RESELECT DMC11'S DESIRED ACTIVE. please note that a message is typed out for setting the switch register equal to DMC11's active. this means if the system has four DMC11s; bits 00,01,02,03 will be set in loc 'DMACTV' from the switch register. Using this switch(SW00) alters that location;therefore if four DMC11s are in the system ***DO NOT*** set switchs greater than SW 03 in the up position. this would be a fatal error. do not select more active DMC11s than there is information on in the status table.

METHOD: A: Load address 200
B: Start with SW 00=1
C: Program will type message
D: Set a switch for each DMC desired active. EXAMPLE: If you have 4 DMC's but only want to run the first and the last set SWR bits 0 and 3 = 1. PRESS CONTINUE
E: Number (IF VALID) will be in data lights (excluding 11/05)
F: Set with any other switch settings desired. PRESS CONTINUE.

4.1.3 DYNAMIC SWITCHES

ERROR SWITCHES

1. SW 12 Delete print out/bell on error.
2. SW 13 Delete error printout.
3. SW 15 Halt on the error.
4. SW Ø8 Goto beginning of the test(on error).
5. SW 1Ø Goto next test(on error).

SCOPE SWITCHES

1. SWØ6 Halt in ROMCLK routine before clocking micro-processor instruction. This allows the operator to scope a micro-processor instruction in the static state before it is clocked. Hit continue to resume running.
2. SWØ9 (if enabled by 'SCOP1') on an error; If an '*' is printed in front of the test no. (ex. *TEST NO. 1Ø) SWØ9 is incorporated in that test and therefore SWØ9 is usually the best switch for the scope loop (SW14=Ø, SW1Ø=Ø, SWØ9=1, SWØ8=Ø). If SWØ9 is not enabeled; and there is a HARD error (constant); SWØ8 is best. (SW14=1,Ø, SW1Ø=Ø, SWØ9=Ø, SWØ8=1). for intermittemt errors; SW14=1 will loop on test reguardless of error or not error. (SW14=1, SW1Ø=Ø, SWØ9=Ø, SWØ8=1,Ø)
3. SW11 Inhibit interations.
4. SW14 Loop on current test.

4.2 STARTING ADDRESS

Starting address is at ØØØ2ØØ there are no other starting addresses for the DMC11 diagnostics. (see section 4.Ø)

NOTE: If address ØØØØ42 is non-zero the program assumes it is under ACT11 or XXDP control and will act accordingly after all available DMC11's are tested the program will return to 'XXDP' or 'ACT-11'.

5. OPERATING PROCEDURE

When program is initially started messages as described in section 4.Ø will be printed, and program will begin running the diagnostic

5.2 PROGRAM AND/OR OPERATOR ACTION

The typical approach should be

1. Halt on error (via SW 15=1) when ever an error occurs.
2. Clear SW 15.
3. Set SW 14: (loop on this test)
4. Set SW 13: (inhibit error print out)

The TEST NUMBER and PC will be typed out and possibily an error message (this depends on the test) to give the operator an idea as to the source of the problem. If it is necessary to know more information concerning the error report; LOOK IN THE LISTING for that TEST NUMBER which was typed out and then NOTE THE PC of thE ERROR REPORT this way the EXACT FUNCTION of the test CAN BE DETERMINED.

6. ERRORS

As described previously there will always be a TEST NUMBER and PC typed out at the time of an error (providing SW 13=0 and SW 12=0), in most cases additional information will be supplied in the the error message to give the operator an indication of the error.

6.2 ERROR RECOVERY

If for some reason the DMC11 should 'HANG THE BUS' (gain control of bus so that console manual functions are inhibited) an init or power down/up is necessary for operator to regain control of cpu. If this should happen; look in location 'TSTNO' (address 1226)for the number of the test that was running at the time of the catastrophic error. In this way the operator will have an idea as to what the DMC11 was doing at the time of the error.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

See section 4. (PLEASE)
Status table should be verified reguardless of how program was started. Also it is important to use this listing along with the information printed on the TTY to completly isolate problems.

7.2 OPERATING RESTRICTIONS

The first time a DMC11 diagnostic is loaded into core and run the STATUS TABLE must be set up. This is done by manual input (SWØØ=1) or by autosizing (SWØØ=Ø and SWØ7=Ø). Thereafter however the status table need not be setup by subsequent restarts or even loading the next DMC diagnostic because the STATUS TABLE is overlayed. The current parameters in the STATUS TABLE are used when SWØ7=1 on start up.

7.3 HARDWARE CONFIGURATION RESTRICTIONS

DMC11(M82ØØ)- Jumper W1 must be in, and switch 7 of E76 must be in the OFF position.

KMC(M82Ø4)- Jumper W1 must be in.

LINE UNIT(M82Ø1)- Jumpers W1, W2, and W4 must be IN. Jumpers W3, and W5 must be OUT. SW8 of E26 must be in the ON POSITION.

LINE UNIT (M82Ø2)- Jumper W1 must be in. SW8 of E26 must be in the OFF position.

8. MISCELLANEOUS

8.1 EXECUTION TIME

All DMC11 device diagnostics will give an 'END PASS' message (providing no errors and sw12=Ø) within 4 mins. This is assuming SW11=1 (DELETE ITERATIONS) is set to give the fastest possible execution. The actual execution time depends greatly on the PDP11 CPU configuration and the amount of memory in the system.

8.2 PASS COMPLETE

NOTE: EVERY time the program is started; the tests will run as if SW11 (delete iterations) was up (=1). This is to 'VERIFY NO HARD ERRORS' as soon as possible. Therefore the first pass -EACH TIME PROGRAM IS STARTED- will be a 'QUICK PASS' until all DMC11's in system are tested. when the diagnostic has completed a pass the following is an example of the print out to be expected.

```
END PASS DZDMH CSR: 175ØØØ VEC: Ø3ØØ PASSES: ØØØØØ1
ERRORS: ØØØØØØ
```

NOTE: The pass count and error counts are cummulitive for each DMC11 that is running, and are set to zero only when the diagnostic is started. Therefore after an overnight run for example, the total passes and errors for each DMC11 since the diagnostic was started are reflected in PASSES: and ERRORS:.

8.4 KEY LOCATIONS

RETURN (1214) Contains the address where program will return when iteration count is reached or if loop on test is asserted.

NEXT (1216) Contains the address of the next test to be peformed.

TSTNO (1226) Contains the number of the test now being peformed.

RUN (1316) The bit in 'RUN' always points to the DMC11 currently being tested. EXAMPLE: (RUN) 1302/0000000001000000 Means that DMC11 no.06 is the DMC11 now running.

DMCR00-DMCR17
DMST00-DMST17
(1500)-(1640)

These locations contain the information needed to test up to 16 (decimal) DMC11s sequentialy. they contain the CSR,VECTOR and STATUS concerning the configuration of each DMC11.

DMACTV (1306) Each bit set in this location indicates that the associated DMC11 will be tested in turn. EXAMPLE: (DMACTV) 1276/0000000000011111 means that DMC11 no. 00,01,02,03,04 will be tested. EXAMPLE: (DMACTV) 1276/0000000000010001 Means that DMC11 no. 00,04 will be tested.

DMCSR (1404) Contains the CSR of the current DMC11 Under test.

8.4A 'STATUS TABLE' (1500-1640)

The table is filled by AUTO SIZING or by the manual parameter input (questions) as described Previously. Also if desired by user; the locations may be altered by hand (toggled in) to suit the specific configuration.

The example status map shown below contains information for two DMC11'S. the table can contain up to 16 DMC11'S. Following the map is a description of the bits for each map entry

MAP OF DMC11 STATUS

PC	CSR	STAT1	STAT2	STAT3
001500	160010	145310	177777	000000
001510	160020	016320	000000	000000

Each map entry contains 4 words which contain the status information for 1 DMC11. The PC shows where in core memory the first of the 4 words is. In the example above the first DMC'S status is in locations, 15ØØ, 15Ø2, 15Ø4, and 15Ø6. The second DMC status is located at 151Ø, 1512, 1514, and 1516. The information contained in each 4 word entry is defined as follows:

```
CSR:    Contains DMC11 CSR address

STAT1:  BITS ØØ-Ø8 IS DMC11 VECTOR ADDRESS
        BIT15=1 MICRO-PROCESSOR HAS CRAM
        BIT15=Ø MICRO-PROCESSOR HAS CROM
        BIT14=1 TURNAROUND CONNECTOR IS ON
        BIT14=Ø NO TURNAROUND CONNECTOR
        BIT13=Ø LINE UNIT IS AN M82Ø1
        BIT13=1 LINE UNIT IS AN M82Ø2
        BIT12=1 NO LINE UNIT
        BITS Ø9-11 IS DMC11 BR PRIORITY LEVEL

STAT2:  LOW BYTE IS SWITCH PAC#1 (DDCMP LINE NUMBER)
        HIGH BYTE IS SWITCH PAC#2 (BM873 BOOT ADD)

STAT3:  BITØ=1 PERFORM FREE RUNNING TESTS ON KMC
                (MUST BE SET MANUALLY, SEE TEST 1)
        BIT1=Ø DMC11-AR (LOW SPEED)
        BIT1=1 DMC11-AL (HIGH SPEED)
```

8.5 METHOD OF AUTO SIZING

8.5.1 FINDING THE CONTROL STATUS REGISTER.

The auto-sizing routine finds a DMC11 as follows: It starts at address 160000 and tests all address in increments of 10 up to and including address 167760. If the address does not time out, the following is done, the first CROM address is written to a 125252 then it is read back. If it contains a -1 or 125252 or a 626 or 16520 a DMC11 or KMC11 has been found, if not, the address is updated by 10 and the search continues. A -1 indicates a DMC11 with no CROM, a 125252 indicates a KMC11 with CRAM, a 626 indicates a DMC11-AL and a 16520 indicates a DMC11-AR. Further tests are performed at this point to determine which line unit, if any, is installed, if a loop-back connector is installed and various switch settings on the line unit. THIS IS WHY THE STATUS TABLE MUST BE VERIFIED BY THE USER AND IF ANY OF THE INFORMATION DOES NOT AGREE WITH THE HARDWARE THE DIAGNOSTIC MUST BE RESTARTED AND THE QUESTIONS MUST BE ANSWERED. All DMC11's in the system will be found by the auto-sizer. If it does not find a DMC11 the diagnostic must be restarted and the questions answered.

8.5.2 FINDING THE VECTOR AND BR LEVEL

The vector area (address 300-776) is filled with the instruction IOT and '.+2' (next address). The processor status is started at 7 and the DMC is programmed to interrupt. The PS is lowered by 1 until the DMC interrupts, a delay is made and if no interupt occures at PS level 3 (because of a bad DMC11) the program assumes vector address 300 at BR level 5 and the problem should be fixed in the diagnostic. Once the problem is fixed; the program should be re-setup again to get correct vector. If an interupt occured; the address to which the DMC11 interupted to is picked up and reported as the vector. NOTE! if the vector reported is not the vector set up by you; there is a problem and AUTO SIZING should not be done.

8.6 SOFTWARE SWITCH REGISTER

If the diagnostic is run on an 11/04 or other CPU without a switch register then a software switch register is used to allow user the same switch options as described previously. If the hardware switch register does not exist or if one does and it contains all ones (177777) this software switch register is used.

Control:

To obtain control at any allowable time during execution of the diagnostic the operator types a CTRL G on the console terminal keyboard. As soon as the CTRL G is recognized, by the diagnostic, the following message will be displayed:

```
    SWR=XXXXXX NEW?
```

Where XXXXXX is the current contents of the software switch register in octal. The software control routine will then await operator action. At which time the operator is required to type one or more of the legal characters: 1) 0 - 7, 2) line feed(<LF>), 3) carriage return(<CR>), or 4) control-U (CTRL U). No check is made for legality. If the input character is not a <LF>, <CR>, or CTRL U it is assumed to be an octal digit.

To change the contents of the SSR the operator simply types the new desired value in octal - leading zeros need not be typed. And terminates the input string with a <CR> or <LF> depending on the program action desired as described below. The input value will be truncated to the last 6 digits typed. At least one digit must be typed on any given input string prior to the terminator before a change to the SSR will occur.

When the input string is terminated with a <CR> the diagnostic will continue execution from the point at which it was interrupted. If a <CR> is the only thing typed the program will continue without changing the SSR. The <LF> differs from the <CR> by restarting the program as if it were restarted at address 200.

If a CTRL U is typed at any point in the input string prior to the terminator the input value will be disregarded and the prompt displayed (SWR = XXXXXX NEW?).

To set the SSR for the starting switches, first load the diagnostic, then hit CTRL G, then start the diagnostic.

```
  DOCUMENT
**************
 DZDMH  LST
**************
```

```
6          MAINDEC-11-DZDMH-F   DMC11 FREE RUNNING TESTS
           COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
           ---------------------------------------------------------------

1666       *************************** TEST 1 ***************************
           FREE RUNNING FLAG MODE DATA TEST
           TRANSMIT A MESSAGE AND VERIFY THE RECEIVED DATA
           LINE UNIT LOOP IS SET FOR THIS TEST.
           ALL FOLLOWING TESTS ARE FREE RUNNING AND ARE PERFORMED
           ONLY ON DMC'S WITH LINE UNITS. IF YOU WISH TO PERFORM
           THESE FREE RUNNING TESTS ON A KMC (NORMALLY THE FREE RUNNING TESTS
           WILL FAIL ON A KMC, THE TIMER IS TOO FAST) THEN YOU MUST
           MANUALLY SET BIT0 OF STAT3 IN THE STATUS MAP. ALSO THE KMC
           MUST HAVE THE MICRO-CODE LOADED BY PREVIOUSLY RUNNING
           DZDMG TEST 2 AND THEN LOADING AND STARTING DZDMH
            WITH SWITCH 7 = 1

1857       *************************** TEST 2 ***************************
           OVERUN TEST
           IN FREE RUNNING MODE SEND MESSAGE WITH NO RECEIVE
           BUFFER AVAILABLE, VERIFY THAT AN OVERRUN ERROR OCCURS

1937       *************************** TEST 3 ***************************
           LOST DATA TEST
           IN FREE RUNNING MODE SEND A MESSAGE LONGER THAN THE RECEIVE
           BUFFER, VERIFY THAT A LOST DATA ERROR OCCURS.

2003       *************************** TEST 4 ***************************
           TRANSMIT NON-EXISTENT MEMORY TEST
           IN FREE RUNNING MODE, LOAD A TRANSMIT BA THAT WILL TIME OUT
           VERIFY THAT A NON-EXISTENT MEMORY ERROR OCCURS

2066       *************************** TEST 5 ***************************
           RECEIVE NON-EXISTENT MEMORY TEST
           IN FREE RUNNING MODE, LOAD A RECEIVE BA THAT WILL TIME OUT
           VERIFY THAT A NON-EXISTENT MEMORY ERROR OCCURS

2132       *************************** TEST 6 ***************************
           PROCESSOR ERROR TEST
           IN FREE RUNNING MODE, DO A BASE TRANSFER REQUEST AFTER A
           BASE HAS BEEN SET UP, VERIFY THAT A PROCESSOR ERROR OCCURS.

2192       *************************** TEST 7 ***************************
           PROCESSOR ERROR TEST
           IN FREE RUNNING MODE DO A RQI WITH AN ILLEGAL 10 CODE
           VERIFY THAT A PROCESSOR ERROR OCCURS

2252       *************************** TEST 10 ***************************
           HALF DUPLEX TEST
           IN FREE RUNNING MODE, SET HALF DUPLEX AND L U LOOP
           SEND A MESSAGE AND VERIFY THAT THERE ARE NO DONES
```

```
2291      *************************** TEST 11 ***************************
          RESUME TEST
          THIS TEST SENDS AND RECEIVES A BUFFER AND SHUTS DOWN THE
          DMC. THEN A MASTER CLEAR IS ISSUED AND A BASE WITH RESUME
          BIT SET IS GIVEN, ANOTHER BUFFER IS SENT AND RECEIVED.
          DATA IS CHECKED.

2380      *************************** TEST 12 ***************************
          FREE RUNNING DATA TEST (INTERRUPT DRIVEN EXERCISER)
          THIS TEST REPEATEDLY QUEUES UP 7 RECEIVE BUFFERS AND
          7 TRANSMIT BUFFERS AND CHECKS DATA WHEN ALL 7 BUFFERS
          ARE RECEIVED. TRANSMIT COUNTS RANGE FROM 2 TO 104.
          DATA IS A BINARY COUNT PATTERN. THE RESUME FUNCTION
          IS CHECKED IN THIS TEST. THIS TEST USES THE TURNAROUND CONNECTOR
          IF IT IS PRESENT, OTHERWISE LINE UNIT LOOP IS SET.
```

```
                                        ;*MAINDEC-11-DZDMH-B   DMC11 FREE RUNNING TESTS
                                        ;*COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                                        ;*------------------------------------------------------------

                                        ;STARTING PROCEDURE
                                        ;LOAD PROGRAM
                                        ;LOAD ADDRESS 000200
                                        ;SWR=0  AUTOSIZE DMC11
                                        ;SW07=1  USE CURRENT DMC11 PARAMETERS
                                        ;SW00=1  INPUT NEW DMC11 PARAMETERS
                                        ;PRESS START
                                        ;PROGRAM WILL TYPE "MAINDEC-11-DZDMH-B   DMC11 FREE RUNNING TESTS"
                                        ;PROGRAM WILL TYPE STATUS MAP
                                        ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                                        ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
                                        ;AND THEN RESUME TESTING
                                        ;SUBSEQUENT RESTARTS WILL NOT TYPE PROGRAM TITLE



                                        ;SWITCH REGISTER OPTIONS
                                        ;-----------------------

        100000                          SW15=100000             ;=1,HALT ON ERROR
        040000                          SW14=40000              ;=1,LOOP ON CURRENT TEST
        020000                          SW13=20000              ;=1,INHIBIT ERROR TYPEOUT
        010000                          SW12=10000              ;=1,DELETE TYPEOUT/BELL ON ERROR.
        004000                          SW11=4000               ;=1,INHIBIT ITERAIIONS
        002000                          SW10=2000               ;=1,ESCAPE TO NEXT TEST ON ERROR
        001000                          SW09=1000               ;=1,LOOP WITH CURRENT DATA
        000400                          SW08=400                ;=1,LOOP ON ERROR
        000200                          SW07=200                ;=1,USE CURRENT DMC11 PARAMETERS, =0,AUTOSIZE DMC11
        000100                          SW06=100                ;=1, HALT BEFORE CLOCKING MICRO-PROCESSOR INSTRUCTION
        000040                          SW05=40
        000020                          SW04=20
        000010                          SW03=10                 ;RESELECT DMC11'S TO BE TESTED (ACTIVE)
        000004                          SW02=4                  ;LOCK ON TEST SELECT
        000002                          SW01=2                  ;RESTART PROGRAM AT SELECTED TEST
        000001                          SW00=1                  ;INPUT DMC11 PARAMETERS
```

```
                                        ;REGISTER DEFINITIONS
                                        ;--------------------

        000000                          R0=%0           ;GENERAL REGISTER
        000001                          R1=%1           ;GENERAL REGISTER
        000002                          R2=%2           ;GENERAL REGISTER
        000003                          R3=%3           ;GENERAL REGISTER
        000004                          R4=%4           ;GENERAL REGISTER
        000005                          R5=%5           ;GENERAL REGISTER
        000006                          SP=%6           ;PROCESSOR STACK POINTER
        000007                          PC=%7           ;PROGRAM COUNTER

                                        ;LOCATION EQUIVALENCIES
                                        ;----------------------

        177776                          PS=177776       ;PROCESSOR STATUS WORD
        001200                          STACK=1200      ;START OF PROCESSOR STACK

                                        ;INSTRUCTION DEFINITIONS
                                        ;-----------------------

        005746                          PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
        005726                          POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
        010046                          PUSHR0=10046    ;SAVE R0 ON STACK
        012600                          POPR0=12600     ;RESTORE R0 FROM STACK
        024646                          PUSH2SP=24646   ;DECREMENT STACK TWICE
        022626                          POP2SP=22626    ;INCREMENT STACK TWICE
                                        .EQUIV  EMT,HLT ;BASIC DEFINITION OF ERROR CALL

                                        ;BIT DEFINITIONS
                                        ;---------------

        100000                          BIT15=100000
        040000                          BIT14=40000
        020000                          BIT13=20000
        010000                          BIT12=10000
        004000                          BIT11=4000
        002000                          BIT10=2000
        001000                          BIT9=1000
        000400                          BIT8=400
        000200                          BIT7=200
        000100                          BIT6=100
        000040                          BIT5=40
        000020                          BIT4=20
        000010                          BIT3=10
        000004                          BIT2=4
        000002                          BIT1=2
        000001                          BIT0=1
```

```

                                ;!*****************************************************************
                                ;------------------------------------------------------------------
                                        ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
                                        ;THE STANDARD "TRAP CATCHER" IS PLACED
                                        ;BETWEEN ADDRESS 0 TO ADDRESS 776.
                                        ;IT LOOKS LIKE "PC+2 HALT".
                                ;------------------------------------------------------------------
                                ;!*****************************************************************

        000000                  .=0
                                ;STANDARD INTERRUPT VECTORS
                                ;--------------------------

        000024                  .=24
000024  005336                          .PFAIL                  ;POWER FAIL HANDLER
000026  000340                          340                     ;SERVICE AT LEVEL 7
000030  004750                          .HLT                    ;ERROR HANDLER
000032  000340                          340                     ;SERVICE AT LEVEL 7
000034  004716                          .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                          340                     ;SERVICE AT LEVEL 7
        000040                  .=40
000040  000000                          0                       ;SAVE FOR ACT-11 OR XXDP
000042  000000                          0                       ;RETURN ADDRESS IF UNDER ACT-11 OR XXDP
000044  000000                          0                       ;SAVE FOR ACT-11 OR XXDP
000046  003522                          $ENDAD                  ;FOR USE WITH ACT-11 OR XXDP
        000052                  .=52
000052  000000                          0                       ;ACT-11 PROGRAM CHARACTERISTICS

        000174                  .=174
000174  000000                  DISPREG:0                       ;SOFTWARE DISPLAY REGISTER
000176  000000                  SWREG:  0                       ;SOFTWARE SWITCH REGISTER

        000200                  .=200
000200  000137  002002                  JMP     .START          ;GO TO START OF PROGRAM


        001000                  .=1000
001000  025377  040515  047111  MTITLE: .ASCII  <377><12>/MAINDEC-11-DZDMH-B/<377>
001025     104  041515  030461          .ASCIZ  /DMC11 FREE RUNNING TESTS/<377>

        001200                  .=1200

                                ;INDIRECT POINTERS TO SWITCH REGISTER AND LIGHT DISPLAY
                                ;------------------------------------------------------

001200  177570                  DISPLAY:177570
001202  177570                  SWR:    177570
```

```

                                ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
                                ;---------------------------------------------------

001204  177560                  TKCSR:  177560                  ;TELETYPE KEYBOARD CONTROL REGISTER
001206  177562                  TKDBR:  177562                  ;TELETYPE KEYBOARD DATA BUFFER
001210  177564                  TPCSR:  177564                  ;TELEPRINTER CONTROL REGISTER
001212  177566                  TPDBR:  177566                  ;TELEPRINTER DATA BUFFER

                                ;PROGRAM CONTROL PARAMETERS
                                ;--------------------------

001214  000000                  RETURN: 0                       ;SCOPE ADDRESS FOR LOOP ON TEST
001216  000000                  NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
001220  000000                  LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT DATA
001222  000003                  ICOUNT: 3                       ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE
001224  000000                  LPCNT:  0                       ;NUMBER OF ITEREATIONS COMPLETED
001226  000000                  TSTNO:  0                       ;NUMBER OF TEST IN PROGRESS
001230  000000                  PASCNT: 0                       ;NUMBER OF PASSES COMPLETED
001232  000000                  ERRCNT: 0                       ;TOTAL NUMBER OF ERRORS
001234  000000                  LSTERR: 0                       ;PC OF LAST ERROR CALL

                                ;PROGRAM VARIABLES
                                ;-----------------

001236  000000                  STRTSW: 0                       ;SWITCHES AT START OF PROGRAM
001240  000000                  STAT:   0                       ;DM STATUS WORD STORAGE
001242  000000                  CLKX:   0
001244  000000                  MASKX:  0
001246  000000                  TEMP1:  0                       ;TEMPORARY STORAGE
001250  000000                  TEMP2:  0                       ;TEMPORARY STORAGE
001252  000000                  TEMP3:  0                       ;TEMPORARY STORAGE
001254  000000                  TEMP4:  0                       ;TEMPORARY STORAGE
001256  000000                  TEMP5:  0                       ;TEMPORARY STORAGE
001260  000000                  SAVR0:  0                       ;R0 STORAGE
001262  000000                  SAVR1:  0                       ;R1 STORAGE
001264  000000                  SAVR2:  0                       ;R2 STORAGE
001266  000000                  SAVR3:  0                       ;R3 STORAGE
001270  000000                  SAVR4:  0                       ;R4 STORAGE
001272  000000                  SAVR5:  0                       ;R5 STORAGE
001274  000000                  SAVSP:  0                       ;STACK POINTER STORAGE
001276  000000                  SAVPC:  0                       ;PROGRAM COUNTER STORAGE
001300  000000                  ZERO:   0
001302  000001                  ONE:    1
001304  000000                  MEMLIM: 0                       ;HIGHEST LOCATION FOR NPR'S
001306  000001                  DMACTV: .BLKW 1                 ;DMC11'S SELECTED ACTIVE.
001310  000001                  DMNUM:  .BLKW 1                 ;OCTAL NUMBER OF DMC11'S.
001312  000001                  SAVACT: .BLKW 1                 ;ORIGINAL ACTV  DEVICES
001314  000001                  SAVNUM: .BLKW 1                 ;WORKABLE NUMBER
001316  000000                  RUN:    0                       ;POINTER TO RUNNING DEVICE.
                                .EVEN
001320  001472                  CREAM:  DM.MAP-6                ;TABLE POINTER.
001322  001676                  MILK:   CNT.MAP-4               ;TABLE POINTER
```

```
                          ;PROGRAM CONTROL FLAGS
                          ;--------------------

001324     000            INIFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
001325     000            ERRFLG: .BYTE   0               ;ERROR OCCURED FLAG
001326     000            LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
001327     000            QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG,
                                                          ;ON FIRST PASS OF EACH DMC11 ITERATIONS WILL BE

                          .EVEN

                                  ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
                                  ;POINTERS TO SUBROUTINES CAN BE FOUND
                                  ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS

                          ;;*****************************************************************
                          ;----------------------------------------------------------------
001330                    .TRPTAB:
           104400         SCOPE=TRAP+0            ;CALL TO SCOPE LOOP AND ITERATION HANDLER
001330     003576                 .SCOPE
           104401         SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
001332     003736                 .SCOP1
           104402         TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
001334     003766                 .TYPE
           104403         INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
001336     004050                 .INSTR
           104404         INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
001340     004154                 .INSTER
           104405         PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
001342     004174                 .PARAM
           104406         SAV05=TRAP+6            ;CALL TO REGISTER SAVE ROUTINE
001344     004374                 .SAV05
           104407         RES05=TRAP+7            ;CALL TO REGISTER RESTORE ROUTINE
001346     004434                 .RES05
           104410         CONVRT=TRAP+10          ;CALL TO DATA OUTPUT ROUTINE
001350     004466                 .CONVRT
           104411         CNVRT=TRAP+11           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
001352     004472                 .CNVRT
           104412         MSTCLR=TRAP+12          ;CALL TO ISUE A MASTER CLEAR
001354     005466                 .MSTCLR
           104413         DELAY=TRAP+13           ;CALL TO DELAY
001356     005436                 .DELAY
           104414         ROMCLK=TRAP+14          ;CALL TO CLOCK ROM ONCE
001360     005504                 .ROMCLK
           104415         DATACLK=TRAP+15         ;CALL TO CLK DATA
001362     005552                 .DATACLK
           104416         TIMER=TRAP+16           ;CALL TO DELAY A CLOCK TICK
001364     005616                 .TIMER

                          ;----------------------------------------------------------------
                          ;;*****************************************************************
```

```
                          ;DMC11 CONTROL INDICATORS FOR CURRENT DMC11 UNDER TEST
                          ;----------------------------------------------------

001366     000000         STAT1:  0
001370     000000         STAT2:  0
001372     000000         STAT3:  0

                          ;DMC11 VECTOR AND REGISTER INDIRECT POINTERS
                          ;------------------------------------------

001374     000000         DMRVEC: 0               ;POINTER TO DMC11 RECEIVER INTERRUPT VECTOR
001376     000000         DMRLVL: 0               ;POINTER TO DMC11 RECEIVER INTERRUPT SERVICE PS
001400     000000         DMTVEC: 0               ;POINTER TO DMC11 TRANSMITTER INTERRUPT VECTOR
001402     000000         DMTLVL: 0               ;POINTER TO DMC11 TRANSMITTER INTERRUPT SERVICE PS
001404     000000         DMCSR:  0               ;POINTER TO DMC11 CONTROL STATUS REGISTER
001406     000000         DMCSRH: 0               ;POINTER TO DMC11 CONTROL STATUS REGISTER HIGH BYTE.
001410     000000         DMCTL:  0               ;POINTER TO DMC11 CONTOL OUT REGISTER
001412     000000         DMPO4:  0               ;POINTER TO DMC11 PORT REGISTER(SEL 4)
001414     000000         DMPO6:  0               ;POINTER TO DMC11 PORT REGISTER(SEL 6)

                          ;TEMP STORAGE
                          ;------------

001416     000000         TEMP:   0
           001460         .=.+40

                          ;DMC11 STATUS TABLE AND ADDRESS ASSIGNMENTS
                          ;-----------------------------------------

           001500         .=1500
001500                    DM.MAP:
001500     000001         DMCR00: .BLKW   1               ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 00
001502     000001         DMS100: .BLKW   1               ;VECTOR FOR DMC11 NUMBER 00
001504     000001         DMS200: .BLKW   1               ;DDCMP LINE# FOR DMC11 NUMBER 00
001506     000001         DMS300: .BLKW   1               ;3RD STATUS WORD

001510     000001         DMCR01: .BLKW   1               ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 01
001512     000001         DMS101: .BLKW   1               ;VECTOR FOR DMC11 NUMBER 01
001514     000001         DMS201: .BLKW   1               ;DDCMP LINE# FOR DMC11 NUMBER 01
001516     000001         DMS301: .BLKW   1               ;3RD STATUS WORD

001520     000001         DMCR02: .BLKW   1               ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 02
001522     000001         DMS102: .BLKW   1               ;VECTOR FOR DMC11 NUMBER 02
001524     000001         DMS202: .BLKW   1               ;DDCMP LINE# FOR DMC11 NUMBER 02
001526     000001         DMS302: .BLKW   1               ;3RD STATUS WORD

001530     000001         DMCR03: .BLKW   1               ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 03
001532     000001         DMS103: .BLKW   1               ;VECTOR FOR DMC11 NUMBER 03
001534     000001         DMS203: .BLKW   1               ;DDCMP LINE# FOR DMC11 NUMBER 03
001536     000001         DMS303: .BLKW   1               ;3RD STATUS WORD

001540     000001         DMCR04: .BLKW   1               ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 04
001542     000001         DMS104: .BLKW   1               ;VECTOR FOR DMC11 NUMBER 04
001544     000001         DMS204: .BLKW   1               ;DDCMP LINE# FOR DMC11 NUMBER 04
001546     000001         DMS304: .BLKW   1               ;3RD STATUS WORD
```

```
 001550  000001                    DMCR05: .BLKW    1              ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 05
 001552  000001                    DMS105: .BLKW    1              ;VECTOR FOR DMC11 NUMBER 05
 001554  000001                    DMS205: .BLKW    1              ;DDCMP LINE# FOR DMC11 NUMBER 05
 001556  000001                    DMS305: .BLKW    1              ;3RD STATUS WORD

 001560  000001                    DMCR06: .BLKW    1              ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 06
 001562  000001                    DMS106: .BLKW    1              ;VECTOR FOR DMC11 NUMBER 06
 001564  000001                    DMS206: .BLKW    1              ;DDCMP LINE# FOR DMC11 NUMBER 06
 001566  000001                    DMS306: .BLKW    1              ;3RD STATUS WORD

 001570  000001                    DMCR07: .BLKW    1              ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 07
 001572  000001                    DMS107: .BLKW    1              ;VECTOR FOR DMC11 NUMBER 07
 001574  000001                    DMS207: .BLKW    1              ;DDCMP LINE# FOR DMC11 NUMBER 07
 001576  000001                    DMS307: .BLKW    1              ;3RD STATUS WORD

 001600  000001                    DMCR10: .BLKW    1              ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 10
 001602  000001                    DMS110: .BLKW    1              ;VECTOR FOR DMC11 NUMBER 10
 001604  000001                    DMS210: .BLKW    1              ;DDCMP LINE# FOR DMC11 NUMBER 10
 001606  000001                    DMS310: .BLKW    1              ;3RD STATUS WORD

 001610  000001                    DMCR11: .BLKW    1              ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 11
 001612  000001                    DMS111: .BLKW    1              ;VECTOR FOR DMC11 NUMBER 11
 001614  000001                    DMS211: .BLKW    1              ;DDCMP LINE# FOR DMC11 NUMBER 11
 001616  000001                    DMS311: .BLKW    1              ;3RD STATUS WORD

 001620  000001                    DMCR12: .BLKW    1              ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 12
 001622  000001                    DMS112: .BLKW    1              ;VECTOR FOR DMC11 NUMBER 12
 001624  000001                    DMS212: .BLKW    1              ;DDCMP LINE# FOR DMC11 NUMBER 12
 001626  000001                    DMS312: .BLKW    1              ;3RD STATUS WORD

 001630  000001                    DMCR13: .BLKW    1              ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 13
 001632  000001                    DMS113: .BLKW    1              ;VECTOR FOR DMC11 NUMBER 13
 001634  000001                    DMS213: .BLKW    1              ;DDCMP LINE# FOR DMC11 NUMBER 13
 001636  000001                    DMS313: .BLKW    1              ;3RD STATUS WORD

 001640  000001                    DMCR14: .BLKW    1              ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 14
 001642  000001                    DMS114: .BLKW    1              ;VECTOR FOR DMC11 NUMBER 14
 001644  000001                    DMS214: .BLKW    1              ;DDCMP LINE# FOR DMC11 NUMBER 14
 001646  000001                    DMS314: .BLKW    1              ;3RD STATUS WORD

 001650  000001                    DMCR15: .BLKW    1              ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 15
 001652  000001                    DMS115: .BLKW    1              ;VECTOR FOR DMC11 NUMBER 15
 001654  000001                    DMS215: .BLKW    1              ;DDCMP LINE# FOR DMC11 NUMBER 15
 001656  000001                    DMS315: .BLKW    1              ;3RD STATUS WORD

 001660  000001                    DMCR16: .BLKW    1              ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 16
 001662  000001                    DMS116: .BLKW    1              ;VECTOR FOR DMC11 NUMBER 16
 001664  000001                    DMS216: .BLKW    1              ;DDCMP LINE# FOR DMC11 NUMBER 16
 001666  000001                    DMS316: .BLKW    1              ;3RD STATUS WORD

 001670  000001                    DMCR17: .BLKW    1              ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 17
 001672  000001                    DMS117: .BLKW    1              ;VECTOR FOR DMC11 NUMBER 17
 001674  000001                    DMS217: .BLKW    1              ;DDCMP LINE# FOR DMC11 NUMBER 17
 001676  000001                    DMS317: .BLKW    1              ;3RD STATUS WORD

 001700  000000                    DM.END: 000000
```

```

                                    ;DMC11 PASS COUNT AND ERROR COUNT TABLE
                                    ;--------------------------------------

 001702                             CNT.MAP:
 001702  000000                     PACT00: 0            ;PASS COUNT FOR DMC11 NUMBER 00
 001704  000000                     ERCT00: 0            ;ERROR COUNT FOR DMC11 NUMBER 00

 001706  000000                     PACT01: 0            ;PASS COUNT FOR DMC11 NUMBER 01
 001710  000000                     ERCT01: 0            ;ERROR COUNT FOR DMC11 NUMBER 01

 001712  000000                     PACT02: 0            ;PASS COUNT FOR DMC11 NUMBER 02
 001714  000000                     ERCT02: 0            ;ERROR COUNT FOR DMC11 NUMBER 02

 001716  000000                     PACT03: 0            ;PASS COUNT FOR DMC11 NUMBER 03
 001720  000000                     ERCT03: 0            ;ERROR COUNT FOR DMC11 NUMBER 03

 001722  000000                     PACT04: 0            ;PASS COUNT FOR DMC11 NUMBER 04
 001724  000000                     ERCT04: 0            ;ERROR COUNT FOR DMC11 NUMBER 04

 001726  000000                     PACT05: 0            ;PASS COUNT FOR DMC11 NUMBER 05
 001730  000000                     ERCT05: 0            ;ERROR COUNT FOR DMC11 NUMBER 05

 001732  000000                     PACT06: 0            ;PASS COUNT FOR DMC11 NUMBER 06
 001734  000000                     ERCT06: 0            ;ERROR COUNT FOR DMC11 NUMBER 06

 001736  000000                     PACT07: 0            ;PASS COUNT FOR DMC11 NUMBER 07
 001740  000000                     ERCT07: 0            ;ERROR COUNT FOR DMC11 NUMBER 07

 001742  000000                     PACT10: 0            ;PASS COUNT FOR DMC11 NUMBER 10
 001744  000000                     ERCT10: 0            ;ERROR COUNT FOR DMC11 NUMBER 10

 001746  000000                     PACT11: 0            ;PASS COUNT FOR DMC11 NUMBER 11
 001750  000000                     ERCT11: 0            ;ERROR COUNT FOR DMC11 NUMBER 11

 001752  000000                     PACT12: 0            ;PASS COUNT FOR DMC11 NUMBER 12
 001754  000000                     ERCT12: 0            ;ERROR COUNT FOR DMC11 NUMBER 12

 001756  000000                     PACT13: 0            ;PASS COUNT FOR DMC11 NUMBER 13
 001760  000000                     ERCT13: 0            ;ERROR COUNT FOR DMC11 NUMBER 13

 001762  000000                     PACT14: 0            ;PASS COUNT FOR DMC11 NUMBER 14
 001764  000000                     ERCT14: 0            ;ERROR COUNT FOR DMC11 NUMBER 14

 001766  000000                     PACT15: 0            ;PASS COUNT FOR DMC11 NUMBER 15
 001770  000000                     ERCT15: 0            ;ERROR COUNT FOR DMC11 NUMBER 15

 001772  000000                     PACT16: 0            ;PASS COUNT FOR DMC11 NUMBER 16
 001774  000000                     ERCT16: 0            ;ERROR COUNT FOR DMC11 NUMBER 16

 001776  000000                     PACT17: 0            ;PASS COUNT FOR DMC11 NUMBER 17
 002000  000000                     ERCT17: 0            ;ERROR COUNT FOR DMC11 NUMBER 17

```

```
                                   FORMAT OF STATUS TABLE
                                   ----------------------

            15  14  13  12  11  10  09  08  07  06  05  04  03  02  01  00
           -----------------------------------------------------------------
           I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
           I C   O   N   T   R   O   L       R   E   G   I   S   T   E   R I     CSR
           I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
           -----------------------------------------------------------------
           -----------------------------------------------------------------
           I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
           I * I * I * I * I * I * I * I *   *   V   E   C   T   O   R   * I     STAT1
           I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
           -----------------------------------------------------------------
           -----------------------------------------------------------------
           I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
           I *   B   M       A   D   D   * I *   L   I   N   E       #   * I     STAT2
           I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
           -----------------------------------------------------------------
           -----------------------------------------------------------------
           I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
           I   I   I   I   I   I   I   I   I   I   I   I   I   I   I * I * I     STAT3
           I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
           -----------------------------------------------------------------


                                   DEFINITION OF FORMAT
                                   --------------------

                        CSR:       CONTAINS DMC11 CSR ADDRESS

                        STAT1:     BITS 00-08 IS DMC11 VECTOR ADDRESS
                                   BIT15=1 MICRO-PROCESSOR HAS CRAM
                                   BIT15=0 MICRO-PROCESSOR HAS CROM
                                   BIT14=1 ???? TURNAROUND CONNECTOR IS ON
                                   BIT14=0 NO TURNAROUND CONNECTOR
                                   BIT13=0 LINE UNIT IS AN M8201
                                   BIT13=1 LINE UNTT IS AN M8202
                                   BIT12=1 NO LINE UNIT
                                   BITS 09-11 IS DMC11 BR PRIORITY LEVEL

                        STAT2:     LOW BYTE IS SWITCH PAC#1 (DDCMP LINE NUMBER)
                                   HIGH BYTE IS SWITCH PAC#2 (BM873 BOOT ADD)

                        STAT3:     BIT0=1 DO FREE RUNNING TESTS ON KMC
                                   (MUST BE SET TO A ONE MANUALLY [PROGRAM DZDMI ONLY])
                                   KMC MUST HAVE MICRO-CODE WRITTEN FROM RUNNING
                                   DZDMG TEST 2 FIRST
                                   BIT1=1 DMC11-AL LOCAL HIGH SPEED MICRO-CODE
                                   BIT1=0 DMC11-AR REMOTE LOW SPEED MICRO-CODE
```

```

                                                        ;PROGRAM INITIALIZATION
                                                        ;LOCK OUT INTERRUPTS
                                                        ;SET UP PROCESSOR STACK
                                                        ;SET UP POWER FAIL VECTOR
                                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
                                                        ;TYPE TITLE MESSAGE

002002  012737  000340  177776          .START: MOV     #340,PS         ;LOCK OUT INTERRUPTS
002010  012706  001200                          MOV     #STACK,SP       ;SET UP STACK
002014  012737  005336  000024                  MOV     #.PFAIL,@#24    ;SET UP POWER FAIL VECTOR
002022  013737  001310  001314                  MOV     DMNUM,SAVNUM    ;SAVE NUMBER OF DEVICES IN SYSTEM.
002030  005037  010016                          CLR     SWFLG           ;CLEAR SOFT TYPEOUT FLAG
002034  105037  001325                          CLRB    ERRFLG          ;CLEAR ERROR FLAG
002040  105037  001327                          CLRB    QV.FLG          ;ZERO QUICK VERIFY FLAG
002044  012737  001470  001320                  MOV     #DM.MAP-10,CREAM;GET MAP POINTER.
002052  012737  001676  001322                  MOV     #CNT.MAP-4,MILK ;GET PASS COUNT MAP POINTER
002060  012737  100000  001316                  MOV     #BIT15,RUN      ;POINT POINTER TO FIRST DEVICE.
002066  012700  001702                          MOV     #CNT.MAP,R0     ;PASS COUNT POINTER TO R0
002072  005023                          23$:    CLR     (R0)+           ;CLEAR TABLE
002074  022700  002002                          CMP     #CNT.MAP+100,R0 ;DONE YET?
002100  001374                                  BNE     23$             ;KEEP GOING
002102  005037  001234                          CLR     LSTERR          ;CLEAR LAST ERROR POINTER
002106  012737  000001  001226                  MOV     #1,TSTNO        ;SET UP FOR TEST 1
002114  012737  002002  001214                  MOV     #.START,RETURN  ;SET UP FOR POWER FAIL BEFORE
                                                                        ;TESTING STARTS
002122  013746  000006                          MOV     @#6,-(SP)       ;SAVE CURRENT VECTORS
002126  013746  000004                          MOV     @#4,-(SP)       ;
002132  012737  002166  000004                  MOV     #6$,@#4         ;SET UP FOR TIMEOUT
002140  012737  177570  001202                  MOV     #177570,SWR     ;SET SWR TO HARD SWR ADDRESS
002146  012737  177570  001200                  MOV     #177570,DISPLAY ;SET DISPLAY TO HARD SWR ADDRESS
002154  022777  177777  177020                  CMP     #-1,@SWR        ;REFERENCE HARDWARE SWITCH REGISTER
002162  001402                                  BEQ     6$+2            ;IF = -1 USE SOFT SWR ANYWAY
002164  000407                                  BR      7$              ;IF IT EXISTS AND NOT = -1 USE HARD SWR
002166  022626                          6$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
002170  012737  000176  001202                  MOV     #SWREG,SWR      ;POINTER TO SOFT SWR
002176  012737  000174  001200                  MOV     #DISPREG,DISPLAY;POINTER TO SOFT DISPLAY REG
002204  012637  000004                  7$:     MOV     (SP)+,@#4       ;RESTORE VECTORS
002210  012637  000006                          MOV     (SP)+,@#6       ;
002214  105737  001324                          TSTB    INIFLG          ;HAS INITIALIZATION BEEN PERFORMED
002220  001006                                  BNE     20$             ;BR IF YES
002222  022737  003522  000042                  CMP     #$ENDAD,@#42    ;IF ACT-11 AUTOMATIC MODE, DON'T TYPE ID
002230  001402                                  BEQ     20$
002232  104402  001000                          TYPE    ,MTITLE         ;TYPE TITLE MESSAGE
002236  004737  007606                  20$:    JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
002242  017737  176734  001236                  MOV     @SWR,STRTSW     ;STORE STARTING SWITCHES
002250  005737  000042                          TST     @#42            ;IS IT RUNNING IN AUTO MODE?
002254  001402                                  BEQ     .+6             ;BR IF NO
002256  005037  001236                          CLR     STRTSW          ;IF YES, CLEAR SWITCHES
002262  032737  000001  001236                  BIT     #SW00,STRTSW    ;IF SW00=1, QUESTIONS ARE ASKED.
002270  001012                                  BNE     17$             ;BR IF SW00=1
002272  105737  001236                          TSTB    STRTSW          ;BIT7=1??
002276  100007                                  BPL     17$             ;BR IF SW07=0
002300  005737  001306                          TST     DMACTV          ;ARE ANY DEVICES SELECTED?
002304  001006                                  BNE     16$             ;BR IF YES
002306  104402  007154                          TYPE,   NOACT           ;NO DEVICES SELECTED.
```

```
002312  000000                          HALT                    ;STOP THE SHOW
002314  000776                          BR      .-2             ;DISQUALIFY CONTINUE SWITCH
002316  004737  010512          17$:    JSR     PC,AUTO.SIZE    ;GO DO THE AUTO SIZE
002322  105737  001324          16$:    TSTB    INIFLG          ;FIRST TIME?
002326  001410                          BEQ     21$             ;BR IF YES
002330  105737  001236                  TSTB    STRTSW          ;IF USING SAME PARAMETERS DONT TYPE MAP
002334  100431                          BMI     1$
002336  032737  000006  001236          BIT     #BIT1!BIT2,STRTSW;IS TEST NO, OR LOCK SELECTED
002344  001403                          BEQ     24$             ;IF NO THEN TYPE STATUS
002346  000424                          BR      1$              ;IF YES DO NOT TYPE STATUS
002350  005137  001324          21$:    COM     INIFLG          ;SET FLAG
002354  104402  006224          24$:    TYPE    ,XHEAD          ;TYPE HEADER
002360  012704  001500                  MOV     #DM.MAP,R4      ;SET POINTER
002364  010437  001246          5$:     MOV     R4,TEMP1        ;SET ADDRESS
002370  012437  001250                  MOV     (R4)+,TEMP2     ;SET CSR
002374  001411                          BEQ     1$              ;ALL DONE IF ZERO
002376  012437  001252                  MOV     (R4)+,TEMP3     ;SET STAT1
002402  012437  001254                  MOV     (R4)+,TEMP4     ;SET STAT2
002406  012437  001256                  MOV     (R4)+,TEMP5     ;SET STAT3
002412  104410                          CONVRT                  ;TYPE OUT STATUS MAP
002414  007454                          XSTATO                  ;
002416  000762                          BR      5$
002420  012700  001500          1$:     MOV     #DM.MAP,R0      ;R0 POINTS TO STATUS TABLE

                                ;;*****************************************************************
                                ;;*AUTO SIZE TEST
                                ;;*THIS TEST VERIFYS THAT THE DMC11S AND/OR KMC11S ARE AT THE CORRECT FLOATING
                                ;;*ADDRESSES FOR YOUR SYSTEM, IF THIS TEST FAILS, IT IS NOT A HARDWARE ERROR,
                                ;;*CHECK THE ADDRESSES OF ALL FLOATING DEVICES (DJ,DH,DQ,DU,DUP,LK,DMC,DZ,KMC),
                                ;;*IF THERE ARE NO OTHER FLOATING DEVICES BEFORE THE DMC11, THE FIRST
                                ;;*DMC11 ADDRESS IS 760070, KMC11 IS 760110. NO DEVICE SHOULD EVER BE AT
                                ;;*ADDRESS 760000. THIS TEST MAY REQUIRE 2 OR MORE ATTEMPTS TO GET THE
                                ;;*RIGHT ADDRESSES. AFTER YOU HAVE CHANGED THE ADDRESS TO WHAT IT TOLD
                                ;;*YOU THE FIRST TIME, IT MAY COME BACK AND TELL YOU A DIFFERENT ADDRESS
                                ;;*THE NEXT TIME YOU RUN IT. PLEASE HAVE PATIENCE, THE FINAL ADDRESS
                                ;;*WILL BE CORRECT (AS LONG AS ALL DEVICES IN FRONT OF THE DMC'S ARE
                                ;;*CORRECT).
                                ;;*****************************************************************

002424  013746  000004                  MOV     @#4,-(SP)       ;SAVE LOC 4
002430  013746  000006                  MOV     @#6,-(SP)       ;SAVE LOC 6
002434  005037  000006                  CLR     @#6             ;CLEAR VEC+2
002440  005037  001252                  CLR     TEMP3           ;CLEAR FLAG
002444  005005                          CLR     R5              ;R5=0=DMC, R5=-1=KMC
002446  011037  001404          AUSTRT: MOV     (R0),DMCSR      ;GET NEXT DMC CSR
002452  001564                          BEQ     AUDONE          ;BR IF DONE
002454  005705                          TST     R5              ;DMC OR KMC?
002456  001005                          BNE     1$              ;BR IF KMC
002460  032760  100000  000002          BIT     #BIT15,2(R0)    ;CHECK FOR DMC CSR
002466  001061                          BNE     SKIP            ;SKIP IF NOT DMC
002470  000404                          BR      2$              ;ITS A DMC SO CONTINUE
002472  032760  100000  000002  1$:     BIT     #BIT15,2(R0)    ;CHECK FOR KMC CSR
002500  001454                          BEQ     SKIP            ;SKIP IF NOT KMC
002502  012737  002674  000004  2$:     MOV     #NODEV,@#4      ;SET UP FOR TIMEOUT
002510  005705                          TST     R5              ;DMC OR KMC?
002512  001003                          BNE     3$              ;BR IF KMC
```

```
002514  012703  000006                  MOV     #6,R3           ;R3 IS COUNT OF DEVICES BEFORE DMC
002520  000402                          BR      4$              ;GO ON
002522  012703  000010          3$:     MOV     #10,R3          ;R3 IS COUNT OF DEVICES BEFORE KMC
002526  012702  003010          4$:     MOV     #DEVTAB,R2      ;R2 IS DEVICE TABLE PONTER
002532  012701  160010                  MOV     #160010,R1      ;START WITH ADDRESS 160010
002536  005711                  FLOAT:  TST     (R1)            ;CHECK ADDRESS IN R1
002540  111204                          MOVB    (R2),R4         ;IF NO TIMEOUT, GET NEXT ADDRESS
002542  060401                          ADD     R4,R1           ;IN R1
002544  005201                          INC     R1              ;
002546  040401                          BIC     R4,R1           ;
002550  005703                          TST     R3              ;ANY MORE DEVICES TO CHECK FOR?
002552  001371                          BNE     FLOAT           ;BR IF YES
002554  012737  002700  000004          MOV     #ERR,@#4        ;OK ONLY DMC'S ARE LEFT, SET UP FOR TIMEOUT
002562  010137  003022                  MOV     R1,XLOC         ;SAVE FIRST DMC/KMC ADDRESS
002566  005705                  FY:     TST     R5              ;DMC OR KMC?
002570  001005                          BNE     1$              ;BR IF KMC
002572  032760  100000  000002          BIT     #BIT15,2(R0)    ;CHECK FOR DMC CSR
002600  001014                          BNE     SKIP            ;SKIP IF NOT DMC
002602  000404                          BR      2$              ;ITS A DMC SO CONTINUE
002604  032760  100000  000002  1$:     BIT     #BIT15,2(R0)    ;CHECK FOR KMC CSR
002612  001407                          BEQ     SKIP            ;SKIP IF NOT KMC
002614  005711                  2$:     TST     (R1)            ;CHECK DMC ADDRESS
002616  020137  001404                  CMP     R1,DMCSR        ;DOES IT MATCH
002622  001411                          BEQ     OK              ;BR IF YES
002624  062701  000010                  ADD     #10,R1          ;GET NEXT DMC ADDRESS
002630  000756                          BR      FY              ;DO IT AGAIN
002632  062700  000010          SKIP:   ADD     #10,R0          ;SKIP TO NEXT CSR IN TABLE
002636  011037  001404                  MOV     (R0),DMCSR      ;GET NEXT CSR
002642  001470                          BEQ     AUDONE          ;BR IF DONE
002644  000750                          BR      FY              ;ELSE CONTINUE
002646  062700  000010          OK:     ADD     #10,R0          ;SKIP TO NEXT DMC CSR
002652  062737  000010  003022          ADD     #10,XLOC        ;UPDATE EXPECTED DMC/KMC ADDRESS
002660  011037  001404                  MOV     (R0),DMCSR      ;GET NEXT DMC/KMC CSR
002664  001457                          BEQ     AUDONE          ;BR IF DONE
002666  013701  003022                  MOV     XLOC,R1         ;GET EXPECTED DMC/KMC ADDRESS
002672  000735                          BR      FY              ;CONTINUE
002674  122243                  NODEV:  CMPB    (R2)+,-(R3)     ;ON TIMEOUT, INC R2, DEC R3
002676  000002                          RTI                     ;RETURN
002700  005737  001252          ERR:    TST     TEMP3           ;CHECK FLAG IF = 0 TYPE HEADER
002704  001014                          BNE     1$              ;SKIP HEADER
002706  104402                          TYPE                    ;TYPEOUT HEADER MESSAGE
002710  007223                          CONERR                  ;CONFIGURATION ERROR!!!!
002712  012737  002700  001276          MOV     #ERR,SAVPC      ;SAVE PC FOR TYPEOUT
002720  104411                          CNVRT                   ;TYPE OUT ERROR PC
002722  002770                          ERRPC                   ;
002724  104402                          TYPE                    ;TYPE REST OF HEADER
002726  007277                          CNERR                   ;
002730  012737  177777  001252          MOV     #-1,TEMP3       ;SET FLAG SO IT ONLY GETS TYPED ONCE
002736  010137  001262          1$:     MOV     R1,SAVR1        ;SAVE R1 FOR TYPEOUT
002742  104410                          CONVRT
002744  002776                          CONTAB                  ;TYPE CSR VALUES
002746  005705                          TST     R5              ;DMC OR KMC ?
002750  001003                          BNE     3$              ;BR IF KMC
002752  104402                          TYPE
002754  007320                          DMCM
002756  000402                          BR      4$              ;CONTINUE
```

```
ØØ276Ø  1Ø44Ø2                    3$:     TYPE
ØØ2762  ØØ733Ø                            KMCM
ØØ2764  Ø22626                    4$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
ØØ2766  ØØØ727                            BR      OK              ;BR TO GET OUT
ØØ277Ø  ØØØØØ1                    ERRPC:  1
ØØ2772     ØØ6     ØØ2                    .BYTE   6,2
ØØ2774  Ø31276                            SAVPC
ØØ2776  ØØØØØ2                    CONTAB: 2
ØØ3ØØØ     ØØ6     ØØ4                    .BYTE   6,4
ØØ3ØØ2  ØØ3Ø22                            XLOC
ØØ3ØØ4     ØØ6     ØØ2                    .BYTE   6,2
ØØ3ØØ6  ØØ1444                            DMCSR
ØØ3Ø1Ø     ØØ7                    DEVTAB: .BYTE   7               ;DJ
ØØ3Ø11     Ø17                            .BYTE   17              ;DH
ØØ3Ø12     ØØ7                            .BYTE   7               ;DQ
ØØ3Ø13     ØØ7                            .BYTE   7               ;DU
ØØ3Ø14     ØØ7                            .BYTE   7               ;DUP
ØØ3Ø15     ØØ7                            .BYTE   7               ;LK
ØØ3Ø16     ØØ7                            .BYTE   7               ;DMC
ØØ3Ø17     ØØ7                            .BYTE   7               ;DZ
ØØ3Ø2Ø     ØØ7                            .BYTE   7               ;KMC
        ØØ3Ø22                    .EVEN
ØØ3Ø22  ØØØØØØ                    XLOC:   Ø
ØØ3Ø24  ØØ57Ø5                    AUDONE: TST     R5              ;DMC?
ØØ3Ø26  ØØ1ØØ5                            BNE     1$              ;BR IF KMC AND ALL DONE
ØØ3Ø3Ø  Ø127Ø5  177777                    MOV     #-1,R5          ;SET R5 TO -1 (KMC)
ØØ3Ø34  Ø127ØØ  ØØ15ØØ                    MOV     #DM.MAP,RØ      ;RESET RØ TO START OF TABLE
ØØ3Ø4Ø  ØØØ6Ø2                            BR      AUSTRT          ;GO DO KMC'S
ØØ3Ø42  Ø12637  ØØØØØ6            1$:     MOV     (SP)+,@#6       ;RESTORE LOC 6
ØØ3Ø46  Ø12637  ØØØØØ4                    MOV     (SP)+,@#4       ;RESTORE LOC 4
ØØ3Ø52  Ø32737  ØØØØ1Ø  ØØ1236            BIT     #SWØ3,STRTSW    ;SELECT SPECIFIC DEVICES??
ØØ3Ø6Ø  ØØ1422                            BEQ     3$              ;BR IF NO.
ØØ3Ø62  1Ø44Ø2  ØØ6144                    TYPE    ,MNEW           ;TYPE THE MESSAGE.
ØØ3Ø66  ØØ5ØØØ                            CLR     RØ              ;ZERO DATA LIGHTS
ØØ3Ø7Ø  ØØØØØØ                            HALT                    ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
ØØ3Ø72  Ø27737  1761Ø4  ØØ1312            CMP     @SWR,SAVACT     ;IS THE NUMBER VALID?
ØØ31ØØ  1Ø14Ø4                            BLOS    2$              ;BR IF NUMBER IS OK.
ØØ31Ø2  1Ø44Ø2  ØØ6ØØ5                    TYPE    ,MERR3          ;TELL USER OF INVALID NUMBER.
ØØ31Ø6  ØØØØØØ                            HALT                    ;STOP EVERY THING.
ØØ311Ø  ØØØ776                            BR      .-2             ;RESTART THE PROGRAM AGAIN.
ØØ3112  Ø17737  176Ø64  ØØ13Ø6    2$:     MOV     @SWR,DMACTV     ;GET NEW DEVICE PATTERN
ØØ312Ø  Ø137ØØ  ØØ13Ø6                    MOV     DMACTV,RØ       ;SHOW THE USER WHAT HE SELECTED.
ØØ3124  ØØØØØØ                            HALT                    ;CONTINUE DYNAMIC SWITCHES.
ØØ3126  Ø127ØØ  ØØØ3ØØ            3$:     MOV     #3ØØ,RØ         ;PREPARE TO CLEAR THE FLOATING
ØØ3132  Ø127Ø1  ØØØ3Ø2                    MOV     #3Ø2,R1         ;VECTOR AREA, 3ØØ-776
ØØ3136  Ø1Ø12Ø                    4$:     MOV     R1,(RØ)+        ;START PUTTING "PC+2 - HALT"
ØØ314Ø  ØØ5Ø21                            CLR     (R1)+           ;IN VECTOR AREA.
ØØ3142  Ø22Ø21                            CMP     (RØ)+,(R1)+     ;POP POINTERS
ØØ3144  Ø227ØØ  ØØ1ØØØ                    CMP     #1ØØØ,RØ        ;ALL DONE??
ØØ315Ø  ØØ1372                            BNE     4$              ;BR IF NO.

                                  ;TEST START AND RESTART
                                  ;---------------------

ØØ3152  Ø127Ø6  ØØ12ØØ            .BEGIN: MOV     #STACK,SP       ;SET UP STACK
ØØ3156  Ø13746  ØØØØØ6                    MOV     @#6,-(SP)       ;SAVE LOC 6
```

```
ØØ3162  Ø13746  ØØØØØ4                    MOV     @#4,-(SP)       ;SAVE LOC 4
ØØ3166  ØØ5ØØØ                            CLR     RØ              ;START AT Ø
ØØ317Ø  Ø12737  ØØ3234  ØØØØØ4            MOV     #2$,@#4         ;SET UP FOR TIME OUT
ØØ3176  ØØ5Ø37  ØØØØØ6                    CLR     @#6             ;TO AUTOSIZE MEMORY
ØØ32Ø2  ØØ572Ø                    6$:     TST     (RØ)+           ;CHECK ADDRESS IN RØ
ØØ32Ø4  Ø227ØØ  157776                    CMP     #157776,RØ      ;IS IT AT LEAST 28K
ØØ321Ø  ØØ1374                            BNE     6$              ;BR IF NO
ØØ3212  1627ØØ  ØØ7776                    SUB     #7776,RØ        ;SAVE 2K FOR MONITORS
ØØ3216  Ø1ØØ37  ØØ13Ø4            7$:     MOV     RØ,MEMLIM       ;STORE MEMORY LIMIT
ØØ3222  Ø12637  ØØØØØ4                    MOV     (SP)+,@#4       ;RESTORE LOC 4
ØØ3226  Ø12637  ØØØØØ6                    MOV     (SP)+,@#6       ;RESTORE LOC 6
ØØ3232  ØØØ413                            BR      1Ø$             ;CONTINUE
ØØ3234  Ø22626                    2$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
ØØ3236  1627ØØ  ØØØØØ4                    SUB     #4,RØ           ;GET LAST GOOD ADDRESS
ØØ3242  1627ØØ  ØØ7776                    SUB     #7776,RØ        ;SAVE 2K FOR MONITORS
ØØ3246  Ø227ØØ  Ø3ØØØØ                    CMP     #3ØØØØ,RØ       ;IS IT 8K?
ØØ3252  ØØ1361                            BNE     7$              ;BR IF NO
ØØ3254  Ø127ØØ  Ø374ØØ                    MOV     #374ØØ,RØ       ;IF 8K DON'T SAVE 2K
ØØ326Ø  ØØØ756                            BR      7$              ;
ØØ3262  Ø12737  ØØØ34Ø  177776    1Ø$:    MOV     #34Ø,PS         ;LOCK OUT INTERRUPTS
ØØ327Ø  Ø32737  ØØØØØ4  ØØ1236            BIT     #BIT2,STRTSW    ;CHECK FOR LOCK ON TEST
ØØ3276  ØØ1411                            BEQ     1$              ;BR IF NO LOCK DESIRED.
ØØ33ØØ  1Ø44Ø2  ØØ6Ø43                    TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
ØØ33Ø4  Ø12737  ØØØ24Ø  ØØ3612            MOV     #NOP,TTST       ;ADJUST SCOPE ROUTINE.
ØØ3312  Ø12737  ØØØ24Ø  ØØ3614            MOV     #NOP,TTST+2     ;SET UP TO LOCK
ØØ332Ø  ØØØ4Ø6                            BR      3$              ;CONTINUE ALONG.
ØØ3322  Ø13737  ØØ373Ø  ØØ3612    1$:     MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
ØØ333Ø  Ø13737  ØØ3732  ØØ3614            MOV     BRX,TTST+2      ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
ØØ3336  Ø12737  Ø1ØØ6Ø  ØØ1214    3$:     MOV     #CYCLE,RETURN   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
ØØ3344  Ø32737  ØØØØØ2  ØØ1236    4$:     BIT     #SWØ1,STRTSW    ;IS TEST NO. SELECTED?
ØØ3352  ØØ1ØØ2                            BNE     5$              ;BR IF YES
ØØ3354  1Ø44Ø2  ØØ5755                    TYPE    ,MR             ;TYPE R
ØØ336Ø  ØØØ177  17563Ø            5$:     JMP     @RETURN         ;START TESTING
```

```
                                                ;END OF PASS
                                                ;TYPE NAME OF TEST
                                                ;UPDATE PASS COUNT
                                                ;CHECK FOR EXIT TO ACT-11
                                                ;RESTART TEST

003364  000005                          .EOP:   RESET                   ;MAKE THE  WORLD CLEAN AGAIN.
003366  005037  001234                          CLR     LSTERR          ;CLEAR LAST ERROR PC
003372  105037  001325                          CLRB    ERRFLG          ;CLEAR ERROR FLAG
003376  005237  001230                          INC     PASCNT          ;UPDATE PASS COUNT
003402  013777  001230  175570                  MOV     PASCNT,@DISPLAY ;DISPLAY PASS COUNT
003410  104402  005733                          TYPE    ,MEPASS         ;TYPE END PASS
003414  104402  006072                          TYPE    ,MCSRX          ;TYPE CSR
003420  104411  003546                          CNVRT   ,XCSR           ;SHOW IT
003424  104402  006100                          TYPE    ,MVECX          ;TYPE VECTOR
003430  104411  003554                          CNVRT   ,XVEC           ;SHOW IT
003434  104402  006106                          TYPE    ,MPASSX         ;TYPE PASSES
003440  104411  003562                          CNVRT   ,XPASS          ;SHOW IT
003444  104402  006117                          TYPE    ,MERRX          ;TYPE ERRORS
003450  104411  003570                          CNVRT   ,XERR           ;SHOW IT
003454  013700  001322                          MOV     MILK,R0         ;GET POINTER TO PASS COUNT
003460  013720  001230                          MOV     PASCNT,(R0)+    ;STORE PASS COUNT FOR THIS DMC11
003464  013720  001232                          MOV     ERRCNT,(R0)+    ;STORE ERROR COUNT FOR THIS DMC11
003470  005337  001314                          DEC     SAVNUM          ;ARE ALL DEVICES TESTED?
003474  001017                                  BNE     RESTRT          ;BR IF NO.
003476  112737  000377  001327                  MOVB    #377,QV.FLG     ;SET THE QUICK VERIFY FLAG.
003504  013737  001310  001314                  MOV     DMNUM,SAVNUM    ;RESTORE THE COUNT
003512  013701  000042                          MOV     @#42,R1         ;CHECK FOR ACT-11 OR DDP
003516  001406                                  BEQ     RESTRT          ;IF NOT, CONTINUE TESTING
003520  000005                                  RESET                   ;STOP THE SHOW--CLEAR THE WORLD
003522                                  $ENDAD:
003522  004711                                  JSR     PC,(R1)
003524  000240                                  NOP
003526  000240                                  NOP
003530  000240                                  NOP
003532  000240                                  NOP
003534  012737  010060  001214          RESTRT: MOV     #CYCLE,RETURN
003542  000137  010060                          JMP     CYCLE
003546  000001                          XCSR:   1
003550     006     002                          .BYTE   6,2
003552  001404                                  DMCSR
003554  000001                          XVEC:   1
003556     004     002                          .BYTE   4,2
003560  001374                                  DMRVEC
003562  000001                          XPASS:  1
003564     006     002                          .BYTE   6,2
003566  001230                                  PASCNT
003570  000001                          XERR:   1
003572     006     002                          .BYTE   6,2
003574  001232                                  ERRCNT

                                                ;SCOPE LOOP AND INTERATION HANDLER
                                                ;---------------------------------

003576  004737  007606                  .SCOPE: JSR     PC,CKSWR        ;CKECK FOR SOFT SWR
003602  010016                                  MOV     R0,(SP)         ;SAVE R0 ON THE STACK
```

```
003604  032777  040000  175370                  BIT     #BIT14,@SWR     ;"LOOP ON THIS TEST"?
003612  001407                          TTST:   BEQ     1$              ;BR IF NO.  (IF LOCK SW01=1; THIS LOC =240)
003614  000437                                  BR      3$              ;GOTO 3$    (IF LOCK SW01=1; THIS LOC =240)
003616  005737  003734                          TST     DONE            ;WAS TKCSR DONE SET?
003622  001434                                  BEQ     3$              ;BR IF NO (LOCKED ON TEST)
003624  005037  003734                          CLR     DONE            ;YES, CLEAR FLAG
003630  000415                                  BR      2$              ;GO TO NEXT TEST
003632  032777  004000  175342          1$:     BIT     #SW11,@SWR      ;DELETE ITERATION?  (QUICK PASS)
003640  001011                                  BNE     2$              ;BR IF YES
003642  105737  001327                          TSTB    QV.FLG          ;HAVE PASSES BEECOMPLETED?
003646  001406                                  BEQ     2$              ;BR IF QUICK PASS.
003650  005237  001224                          INC     LPCNT           ;UPDATE ITERATION COUNTER
003654  023737  001224  001222                  CMP     LPCNT,ICOUNT    ;ARE ALL ITERATIONS DONE??
003662  101414                                  BLOS    3$              ;BR IF NOT YET
003664  105037  001325                  2$:     CLRB    ERRFLG          ;PREPARE FOR NEW TEST
003670  005037  001224                          CLR     LPCNT           ;START ICOUNTER AT 0
003674  005037  001220                          CLR     LOCK
003700  012737  000020  001222                  MOV     #20,ICOUNT      ;RESET ITERATIONS
003706  013737  001216  001214                  MOV     NEXT,RETURN     ;GET NEXT TEST
003714  011600                          3$:     MOV     (SP),R0         ;POP R0 OFF OF THE STACK
003716  022626                                  POP2SP                  ;FAKE AN "RTI"
003720  013701  001404                          MOV     DMCSR,R1        ;R1 CONTAINS BASE DMC ADDRESS
003724  000177  175264                          JMP     @RETURN         ;GO DO THE TEST
003730  001407                          BRW:    1407
003732  000437                          BRX:    437
003734  000000                          DONE:   0

                                                ;CHECK FOR FREEZE ON CURRENT DATA
                                                ;--------------------------------

003736  004737  007606                  .SCOP1: JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
003742  032777  001000  175232                  BIT     #SW09,@SWR      ;IS SW09=1(SET)?
003750  001405                                  BEQ     1$              ;BR IF NOT SET.
003752  005737  001220                          TST     LOCK
003756  001402                                  BEQ     1$
003760  013716  001220                          MOV     LOCK,(SP)       ;GOTO THE ADDRESS IN LOCK.
003764  000002                          1$:     RTI                     ;GO BACK.

                                                ;TELETYPE OUTPUT ROUTINE
                                                ;-----------------------

003766  010546                          .TYPE:  MOV     R5,-(SP)        ;SAVE R5 ON THE STACK.
003770  017605  000002                          MOV     @2(SP),R5       ;GET ADDRESS OF MESSAGE.
003774  062766  000002  000002                  ADD     #2,2(SP)        ;POP OVER ADDRESS.
004002  005737  010016                  4$:     TST     SWFLG           ;SOFT SWR MESSAGE?
004006  001004                                  BNE     1$              ;IF YES TYPE IT OUT REGARDLESS OF SW12
004010  032777  010000  175164                  BIT     #SW12,@SWR      ;INHIBIT ALL PRINT OUT??
004016  001012                                  BNE     3$              ;BR IF NO PRINT OUT WANTED (SW12=1)
004020  105715                          1$:     TSTB    (R5)            ;IS NUMBER MINUS? (MSB=1(BIT7))
004022  100002                                  BPL     2$              ;BR IF NUMBER IS PLUS
004024  104402  005617                          TYPE    ,MCRLF          ;TYPE A CR/LF!
004030  105777  175154                  2$:     TSTB    @TPCSR          ;TTY READY?
004034  100375                                  BPL     2$              ;BR IF NO.
004036  112577  175150                          MOVB    (R5)+,@TPDBR    ;PRINT CURRENT CHAR.
004042  001357                                  BNE     4$              ;IF NOT ZERO KEEP PRINTING!
004044  012605                          3$:     MOV     (SP)+,R5        ;END OF OUTPUT, RESTORE R5
```

```
004046  000002                          RTI                     ;GO HOME
                                        ;-------------------------

004050  010346                  .INSTR: MOV     R3,-(SP)        ;SAVE R3 ON STACK
004052  010446                          MOV     R4,-(SP)        ;SAVE R4 ON STACK
004054  017637  000004  004072          MOV     @4(SP),.MSG
004062  062766  000002  000004          ADD     #2,4(SP)
004070  104402                  .INST1: TYPE
004072  000000                  .MSG:   0
004074  012704  007502                  MOV     #INBUF,R4
004100  012703  000007                  MOV     #7,R3
004104  105777  175074          1$:     TSTB    @TKCSR
004110  100375                          BPL     1$
004112  117714  175070                  MOVB    @TKDBR,(R4)
004116  142714  000200                  BICB    #200,(R4)
004122  122427  000015                  CMPB    (R4)+,#15
004126  001417                          BEQ     INSTR2
004130  105777  175054          2$:     TSTB    @TPCSR
004134  100375                          BPL     2$
004136  017777  175044  175046          MOV     @TKDBR,@TPDBR
004144  005303                          DEC     R3
004146  001356                          BNE     1$
004150  012604                          MOV     (SP)+,R4
004152  012603                          MOV     (SP)+,R3
004154  104402  005666          .INSTE: TYPE    ,MQM
004160  010346                          MOV     R3,-(SP)
004162  010446                          MOV     R4,-(SP)
004164  000741                          BR      .INST1
004166  012604                  INSTR2: MOV     (SP)+,R4        ;RESTORE R4
004170  012603                          MOV     (SP)+,R3        ;RESTORE R3
004172  000002                          RTI

                                        ;CONVERT ASCII STRING TO OCTAL
                                        ;----------------------------

004174  010546                  .PARAM: MOV     R5,-(SP)
004176  010446                          MOV     R4,-(SP)
004200  016605  000004                  MOV     4(SP),R5
004204  012537  004364                  MOV     (R5)+,LOLIM
004210  012537  004366                  MOV     (R5)+,HILIM
004214  012537  004370                  MOV     (R5)+,DEVADR
004220  112537  004372                  MOVB    (R5)+,LOBITS
004224  112537  004373                  MOVB    (R5)+,ADRCNT
004230  010566  000004                  MOV     R5,4(SP)
004234  005005                  PARAM1: CLR     R5
004236  012704  007502                  MOV     #INBUF,R4
004242  122714  000015                  CMPB    #15,(R4)
004246  001420                          BEQ     PARERR
004250  121427  000060          1$:     CMPB    (R4),#60
004254  002415                          BLT     PARERR
004256  121427  000067                  CMPB    (R4),#67
004262  003012                          BGT     PARERR
004264  142714  000060                  BICB    #60,(R4)
004270  152405                          BISB    (R4)+,R5
004272  122714  000015                  CMPB    #15,(R4)
004276  001406                          BEQ     LIMITS
```

```
004300  006305                          ASL     R5
004302  006305                          ASL     R5
004304  006305                          ASL     R5
004306  000760                          BR      1$
004310  104404                  PARERR: INSTER
004312  000750                          BR      PARAM1

                                        ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
                                        ;--------------------------------------

004314  020537  004366          LIMITS: CMP     R5,HILIM
004320  101373                          BHI     PARERR
004322  020537  004364                  CMP     R5,LOLIM
004326  103770                          BLO     PARERR
004330  133705  004372                  BITB    LOBITS,R5
004334  001365                          BNE     PARERR

                                        ;STORE NUMBER AT SPECIFIED ADDRESS

004336  013704  004370                  MOV     DEVADR,R4
004342  010524                  1$:     MOV     R5,(R4)+
004344  062705  000002                  ADD     #2,R5
004350  105337  004373                  DECB    ADRCNT
004354  001372                          BNE     1$
004356  012604                          MOV     (SP)+,R4
004360  012605                          MOV     (SP)+,R5
004362  000002                          RTI
004364  000000                  LOLIM:  0
004366  000000                  HILIM:  0
004370  000000                  DEVADR: 0
004372  000000                  LOBITS: 0
        004373                  ADRCNT=LOBITS+1

                                        ;SAVE PC OF TEST THAT FAILED AND R0-R5
                                        ;-------------------------------------

004374  016637  000004  001276  .SAV05: MOV     4(SP),SAVPC     ;SAVE R7 (PC)

                                        ;SAVE R0-R5

004402  010537  001272          SV05:   MOV     R5,SAVR5        ;SAVE R5
004406  010437  001270                  MOV     R4,SAVR4        ;SAVE R4
004412  010337  001266                  MOV     R3,SAVR3        ;SAVE R3
004416  010237  001264                  MOV     R2,SAVR2        ;SAVE R2
004422  010137  001262                  MOV     R1,SAVR1        ;SAVE R1
004426  010037  001260                  MOV     R0,SAVR0        ;SAVE R0
004432  000002                          RTI                     ;LEAVE.

                                        ;RESTORE R0-R5

004434  013700  001260          .RES05: MOV     SAVR0,R0        ;RESTORE R0
004440  013701  001262                  MOV     SAVR1,R1        ;RESTORE R1
004444  013702  001264                  MOV     SAVR2,R2        ;RESTORE R2
004450  013703  001266                  MOV     SAVR3,R3        ;RESTORE R3
004454  013704  001270                  MOV     SAVR4,R4        ;RESTORE R4
004460  013705  001272                  MOV     SAVR5,R5        ;RESTORE R5
```

```
ØØ4464  ØØØØØ2                                  RTI                     ;LEAVE

                                                ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
                                                ;-------------------------------------------------------

ØØ4466  1Ø44Ø2  ØØ5672                  .CONVR: TYPE    ,MCRLF
ØØ4472  Ø1ØØ46                          .CNVRT: MOV     RØ,-(SP)
ØØ4474  Ø1Ø146                                  MOV     R1,-(SP)
ØØ4476  Ø1Ø346                                  MOV     R3,-(SP)
ØØ45ØØ  Ø1Ø446                                  MOV     R4,-(SP)
ØØ45Ø2  Ø1Ø546                                  MOV     R5,-(SP)
ØØ45Ø4  Ø176Ø1  ØØØØ12                          MOV     @12(SP),R1
ØØ451Ø  Ø62766  ØØØØØ2  ØØØØ12                  ADD     #2,12(SP)
ØØ4516  Ø12137  ØØ471Ø                          MOV     (R1)+,WRDCNT
ØØ4522  112137  ØØ4712                  1$:     MOVB    (R1)+,CHRCNT
ØØ4526  112137  ØØ4713                          MOVB    (R1)+,SPACNT
ØØ4532  Ø13137  ØØ4714                          MOV     @(R1)+,BINWRD
ØØ4536  122737  ØØØØØ3  ØØ4712                  CMPB    #3,CHRCNT
ØØ4544  ØØ1ØØ3                                  BNE     2$
ØØ4546  Ø42737  1774ØØ  ØØ4714                  BIC     #1774ØØ,BINWRD
ØØ4554  Ø137Ø4  ØØ4714                  2$:     MOV     BINWRD,R4
ØØ456Ø  1137Ø5  ØØ4712                          MOVB    CHRCNT,R5
ØØ4564  Ø127ØØ  ØØ1416                          MOV     #TEMP,RØ
ØØ457Ø  Ø1Ø4Ø3                          3$:     MOV     R4,R3
ØØ4572  Ø427Ø3  17777Ø                          BIC     #17777Ø,R3
ØØ4576  Ø627Ø3  ØØØØ6Ø                          ADD     #Ø6Ø,R3
ØØ46Ø2  11Ø32Ø                                  MOVB    R3,(RØ)+
ØØ46Ø4  ØØØ241                                  CLC
ØØ46Ø6  ØØ6ØØ4                                  ROR     R4
ØØ461Ø  ØØØ241                                  CLC
ØØ4612  ØØ6ØØ4                                  ROR     R4
ØØ4614  ØØØ241                                  CLC
ØØ4616  ØØ6ØØ4                                  ROR     R4
ØØ462Ø  ØØ53Ø5                                  DEC     R5
ØØ4622  ØØ1362                                  BNE     3$
ØØ4624  Ø127Ø3  ØØ7544                          MOV     #MDATA,R3
ØØ463Ø  114Ø23                          4$:     MOVB    -(RØ),(R3)+
ØØ4632  1Ø5337  ØØ4712                          DECB    CHRCNT
ØØ4636  ØØ1374                                  BNE     4$
ØØ464Ø  1Ø5737  ØØ4713                          TSTB    SPACNT
ØØ4644  ØØ14Ø5                                  BEQ     6$
ØØ4646  112723  ØØØØ4Ø                  5$:     MOVB    #Ø4Ø,(R3)+
ØØ4652  1Ø5337  ØØ4713                          DECB    SPACNT
ØØ4656  ØØ1373                                  BNE     5$
ØØ466Ø  1Ø5Ø13                          6$:     CLRB    (R3)
ØØ4662  1Ø44Ø2  ØØ7544                          TYPE    ,MDATA
ØØ4666  ØØ5337  ØØ471Ø                          DEC     WRDCNT
ØØ4672  ØØ1313                                  BNE     1$
ØØ4674  Ø126Ø5                                  MOV     (SP)+,R5
ØØ4676  Ø126Ø4                                  MOV     (SP)+,R4
ØØ47ØØ  Ø126Ø3                                  MOV     (SP)+,R3
ØØ47Ø2  Ø126Ø1                                  MOV     (SP)+,R1
ØØ47Ø4  Ø126ØØ                                  MOV     (SP)+,RØ
ØØ47Ø6  ØØØØØ2                                  RTI
ØØ471Ø  ØØØØØØ                          WRDCNT: Ø
ØØ4712  ØØØØØØ                          CHRCNT: Ø
```

```
        ØØ4713                          SPACNT=CHRCNT+1
ØØ4714  ØØØØØØ                          BINWRD: Ø


                                                ;TRAP DISPATCH SERVICE
                                                ;ARGUMENT OF TRAP IS EXTRACTED
                                                ;AND USED AS OFFSET TO OBTAIN POINTER
                                                ;TO SELECTED SUBROUTINE

ØØ4716  Ø11646                          .TRPSR: MOV     (SP),-(SP)          ;GET PC OF RETURN
ØØ472Ø  162716  ØØØØØ2                          SUB     #2,(SP)             ;=PC OF TRAP
ØØ4724  Ø17616  ØØØØØØ                          MOV     @(SP),(SP)          ;GET TRP
ØØ473Ø  ØØ631Ø                          TRPOK:  ASL     (SP)                ;MULTIPLY TRAP ARG BY 2
ØØ4732  Ø42716  177ØØ1                          BIC     #177ØØ1,(SP)        ;CLEAR UNWANTED BITS
ØØ4736  Ø62716  ØØ133Ø                          ADD     #.TRPTAB,(SP)       ;POINTER TO SUBROUTINE ADDRESS
ØØ4742  Ø17616  ØØØØØØ                          MOV     @(SP),(SP)          ;SUBROUTINE ADDRESS
ØØ4746  ØØØ136                                  JMP     @(SP)+              ;GO TO SUBROUTINE

                                                ;ERROR HANDLER
                                                ;-------------

ØØ475Ø  ØØ4737  ØØ76Ø6                  .HLT:   JSR     PC,CKSWR            ;CHECK FOR SOFT SWR
ØØ4754  Ø32777  Ø1ØØØØ  17422Ø                  BIT     #SW12,@SWR          ;BELL ON ERROR?
ØØ4762  ØØ14Ø6                                  BEQ     XBX                 ;BR IF NO BELL
ØØ4764  1Ø5777  17422Ø                          TSTB    @TPCSR              ;TTY READY.
ØØ477Ø  1ØØØØ3                                  BPL     XBX                 ;DON'T WAIT IF TTY NOT READY.
ØØ4772  112777  ØØØ2Ø7  174212                  MOVB    #2Ø7,@TPDBR         ;PUSH A BELL AT THE TTY.
ØØ5ØØØ  Ø32777  Ø2ØØØØ  174174          XBX:    BIT     #SW13,@SWR          ;DELETE ERROR PRINT OUT?
ØØ5ØØ6  ØØ11Ø5                                  BNE     HALTS               ;BR IF NO PRINT OUT WANTED.
ØØ5Ø1Ø  Ø21637  ØØ1234                          CMP     (SP),LSTERR         ;WAS THIS ERROR FOUND LAST TIME?
ØØ5Ø14  ØØ14Ø4                                  BEQ     1$                  ;BR IF YES
ØØ5Ø16  Ø11637  ØØ1234                          MOV     (SP),LSTERR         ;RECORD BEING HERE
ØØ5Ø22  1Ø5Ø37  ØØ1325                          CLRB    ERRFLG              ;PREPARE HEADER
ØØ5Ø26  1Ø44Ø6                          1$:     SAVØ5                       ;SAVE ALL PROC REGISTERS
ØØ5Ø3Ø  Ø116Ø5                                  MOV     (SP),R5             ;GET THE PC OF ERROR
ØØ5Ø32  1627Ø5  ØØØØØ2                          SUB     #2,R5               ;GET ADDRESS OF TRAP CALL
ØØ5Ø36  Ø115Ø4                                  MOV     (R5),R4             ;GET HLT INSTRUCTION
ØØ5Ø4Ø  ØØ63Ø4                                  ASL     R4                  ;MULT BY TWO
ØØ5Ø42  Ø615Ø4                                  ADD     (R5),R4             ;DOUBLE IT
ØØ5Ø44  ØØ63Ø4                                  ASL     R4                  ;MULT AGAIN
ØØ5Ø46  Ø427Ø4  177ØØ1                          BIC     #177ØØ1,R4          ;CLEAR JUNK
ØØ5Ø52  Ø627Ø4  Ø23414                          ADD     #.ERRTAB,R4         ;GET POINTER
ØØ5Ø56  Ø12437  ØØ5172                          MOV     (R4)+,ERRMSG        ;GET ERROR MESSAGE
ØØ5Ø62  Ø12437  ØØ52Ø4                          MOV     (R4)+,DATAHD        ;GET DATA HEADRER
ØØ5Ø66  Ø11437  ØØ5216                          MOV     (R4),DATABP         ;GET DATA TABLE
ØØ5Ø72  1Ø5737  ØØ1325                          TSTB    ERRFLG              ;TYPE HEADREER
ØØ5Ø76  ØØ14Ø3                                  BEQ     TYPMSG              ;BR IF YES
ØØ51ØØ  ØØ5737  ØØ5216                          TST     DATABP              ;DOES DATA TABLE EXIST?
ØØ51Ø4  ØØ1Ø4Ø                                  BNE     TYPDAT              ;BR IF YES.
ØØ51Ø6  1Ø44Ø2  ØØ5672                  TYPMSG: TYPE    ,MCRLF
ØØ5112  1Ø44Ø2  ØØ5672                          TYPE    ,MCRLF
ØØ5116  ØØ5737  ØØ122Ø                          TST     LOCK
ØØ5122  ØØ14Ø2                                  BEQ     1$
ØØ5124  1Ø44Ø2  ØØ6142                          TYPE    ,MASTEK
ØØ513Ø  1Ø44Ø2  Ø2613Ø                  1$:     TYPE    ,MTSTN
ØØ5134  1Ø4411  ØØ533Ø                          CNVRT   ,XTSTN              ;SHOW IT
```

```
005140  104402  006217                          TYPE    ,MERRPC         ;TYPE PC.
005144  104411  005322                          CNVRT   ,ERTAB0         ;SHOW IT
005150  104402  005672                          TYPE    ,MCRLF          ;GIVE A CR/LF
005154  112737  177777  001325                  MOVB    #-1,ERRFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
005162  005737  005172                          TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
005166  001402                                  BEQ     WRKO.FM         ;BR IF NO.
005170  104402                                  TYPE                    ;TYPE
005172  000000                          ERRMSG: 0                       ;    ERROR MESSAGE
005174                                  WRKO.FM:                        ;
005174  005737  005204                          TST     DATAHD          ;DATA HEADER?
005200  001402                                  BEQ     TYPDAT          ;BR IF NO
005202  104402                                  TYPE                    ;TYPE
005204  000000                          DATAHD: 0                       ;    DATA HEADER
005206  005737  005216                  TYPDAT: TST     DATABP          ;DATA TABLE?
005212  001402                                  BEQ     RESREG          ;BR IF NO.
005214  104410                                  CONVRT                  ;SHOW
005216  000000                          DATABP: 0                       ;    DATA TABLE
005220  104407                          RESREG: RES05                   ;RESTORE PROC REGISTERS
005222  022737  003522  000042          HALTS:  CMP     #$ENDAD,@#42    ;IF ACT-11 AUTOMATIC MODE, HALT!!
005230  001403                                  BEQ     1$
005232  005777  173744                          TST     @SWR            ;HALT ON ERROR?
005236  100005                                  BPL     EXITER          ;BR IF NO HALT ON ERROR
005240  010046                          1$:     PUSHR0                  ;SAVE RO
005242  016600  000002                          MOV     2(SP),R0        ;SHOW ERROR PC IN DATA LIGHTS
005246  000000                                  HALT                    ;HALT
005250  012600                                  POPR0                   ;GET RO
005252  005237  001232                  EXITER: INC     ERRCNT          ;UPDATE ERROR COUNT
005256  032777  000400  173716                  BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
005264  001007                                  BNE     1$              ;BR IF YES
005266  032777  002000  173706                  BIT     #SW10,@SWR      ;GOTO NEXT TEST?
005274  001411                                  BEQ     2$              ;BR IF NO
005276  013737  001216  001214                  MOV     NEXT,RETURN     ;SET FOR NEXT TEST
005304  012706  001200                  1$:     MOV     #STACK,SP       ;RESET SP
005310  013701  001404                          MOV     DMCSR,R1        ;SET UP R1
005314  000177  173674                          JMP     @RETURN         ;GOTO SPECIFIED TEST
005320  000002                          2$:     RTI                     ;RETURN
005322  000001                          ERTAB0: 1
005324     006     002                          .BYTE   6,2
005326  001276                                  SAVPC
005330  000001                          XTSTN:  1
005332     003     002                          .BYTE   3,2
005334  001226                                  TSTNO
                                                ;ENTER HERE ON POWER FAILURE
                                                ;---------------------------


005336                                  .PFAIL:
005336  012737  005350  000024                  MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
005344  000000                                  HALT                            ;HALT ON POWER DOWN NORMAL
005346  000777                                  BR      .

                                                ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

005350                                  RESTAR:
005350  012737  005336  000024                  MOV     #.PFAIL,24      ;SET UP FOR POWER FAILURE
005356  012706  001200                          MOV     #STACK,SP       ;RESET THE STACK POINTER
```

```
005362  013701  001404                          MOV     DMCSR,R1        ;RESTORE R1
005366  005037  001416                          CLR     TEMP            ;READY FOR TIMMER
005372  005237  001416                          INC     TEMP            ;PLUS ONE TO THE TIMER!
005376  001375                                  BNE     .-4             ;BR IF MORE TO GO
005400  104402  005675                          TYPE    ,MPFAIL         ;TYPE THE MESSAGE
005404  104411  005430                          CNVRT   ,PFTAB          ;TELL WHAT TEST TO RETURN TO.
005410  105037  001325                          CLRB    ERRFLG          ;START CLEAN
005414  005037  001234                          CLR     LSTERR          ;"""""""""""
005420  005011                                  CLR     (R1)            ;CLEAR MAINT BITS
005422  104412                                  MSTCLR                  ;START CLEAN UP OF DEVICE
005424  000177  173564                          JMP     @RETURN         ;START DOING THAT TEST AGAIN.
005430  000001                          PFTAB:  1
005432     003     002                  .BYTE   3,2
005434  001226                                  TSTNO

005436                                  .DELAY:
005436  012777  000020  173746                  MOV     #20,@DMP04
005444  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
005446  121111                                  121111                          ;POKE CLOCK DELAY BIT
005450                                  1$:
005450  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
005452  121224                                  121224                          ;PORT4_IBUS*11
005454  032777  000020  173730                  BIT     #BIT4,@DMP04    ;IS CLOCK BIT SET?
005462  001772                                  BEQ     1$              ;BR IF NO
005464  000002                                  RTI

005466                                  .MSTCLR:
005466  152777  000100  173712                  BISB    #BIT6,@DMCSRH   ;SET MASTER CLEAR
005474  142777  000300  173704                  BICB    #BIT6!BIT7,@DMCSRH  ;CLEAR MASTER CLEAR AND RUN
005502  000002                                  RTI                     ;RETURN

005504                                  .ROMCLK:
005504  152777  000002  173674                  BISB    #BIT1,@DMCSRH   ;SET ROMI
005512  013677  173676                          MOV     @(SP)+,@DMP06   ;LOAD INSTRUCTION IN SEL6
005516  062746  000002                          ADD     #2,-(SP)        ;ADJUST STACK
005522  032777  000100  173452                  BIT     #SW06,@SWR      ;HALT IF SW06 =1
005530  001401                                  BEQ     1$              ;BR IF SW06 =0
005532  000000                                  HALT                    ;HALT BEFORE CLOCKING INSTRUCTION
005534  152777  000003  173644          1$:     BISB    #BIT1!BIT0,@DMCSRH ;CLOCK INSTRUCTION
005542  142777  000007  173636                  BICB    #BIT2!BIT1!BIT0,@DMCSRH  ;CLEAR ROMO, ROMI, STEP
005550  000002                                  RTI

005552                                  .DATACLK:
005552  013637  001416                          MOV     @(SP)+,TEMP     ;PUT TICK COUNT IN TEMP
005556  062746  000002                          ADD     #2,-(SP)        ;ADJUST STACK
005562  152777  000020  173616          1$:     BISB    #BIT4,@DMCSRH   ;SET STEP LU
005570  027777  173610  173606                  CMP     @DMCSR,@DMCSR   ;WASTE TIME
005576  142777  000020  173602                  BICB    #BIT4,@DMCSRH   ;CLEAR STEP LU
005604  005337  001416                          DEC     TEMP            ;DEC TICK COUNT
005610  001364                                  BNE     1$              ;BR IF NOT DONE
005612  000002                                  RTI                     ;RETURN
005614  000001                          3$:     .BLKW 1

005616                                  .TIMER:
005616  013637  001416                          MOV     @(SP)+,TEMP     ;MOVE COUNT TO TEMP
005622  062746  000002                          ADD     #2,-(SP)        ;ADJUST STACK
```

```
     005626                              1$:
     005626  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
     005630  021364                              021364                  ;PORT4_IBUS* REG11
     005632  032777  000002  173552              BIT     #2,@DMPO4       ;IS PGM CLOCK BIT CLEAR?
     005640  001772                              BEQ     1$              ;BR IF YES
     005642                              2$:
     005642  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
     005644  021364                              021364                  ;PORT4_IBUS* REG11
     005646  032777  000002  173536              BIT     #2,@DMPO4       ;IS PGM CLOCK BIT SET?
     005654  001372                              BNE     2$              ;BR IF YES
     005656  005337  001416                      DEC     TEMP            ;DEC COUNT
     005662  001361                              BNE     1$              ;BR IF NOT DONE
     005664  000002                              RTI                     ;RETURN

     005666  020040  000077              MQM:    .ASCIZ  / ?/
(2)  005672  005015     000              MCRLF:  .ASCIZ  <15><12>
(2)  005675     377  053520  020122      MPFAIL: .ASCIZ  <377>/PWR FAILED, RESTART AT TEST /
(2)  005733     377  047105  020104      MEPASS: .ASCIZ  <377>/END PASS DZDMH  /
(2)  005755     377  000122              MR:     .ASCIZ  <377>/R/
(2)  005760  047377  020117  042504      MERR2:  .ASCIZ  <377>/NO DEVICES PRESENT./
(2)  006005     377  047111  052523      MERR3:  .ASCIZ  <377>/INSUFFICIENT DATA!/
(2)  006031     377  042524  052123      MTSTPC: .ASCIZ  <377>/TEST PC-/
(2)  006043     377  047514  045503      MLOCK:  .ASCIZ  <377>/LOCK ON SELECTED TEST/
(2)  006072  051503  035122  000040      MCSRX:  .ASCIZ   /CSR: /
(2)  006100  042526  035103  000040      MVECX:  .ASCIZ  /VEC: /
(2)  006106  040520  051523  051505      MPASSX: .ASCIZ  /PASSES: /
(2)  006117     105  051122  051117      MERRX:  .ASCIZ  /ERRORS: /
(2)  006130  042524  052123  047040      MTSTN:  .ASCIZ  /TEST NO: /
(2)  006142  000052                      MASTEK: .ASCIZ  /*/
(2)  006144  051777  052105  051440      MNEW:   .ASCIZ  <377>/SET SWITCH REG TO DMC11'S DESIRED ACTIVE./
(2)  006217     120  035103  000040      MERRPC: .ASCIZ  /PC: /
(2)  006224  020212  020040  020040      XHEAD:  .ASCIT  <212>/            MAP OF DMC11 STATUS/
(2)  006263     377  020040  020040              .ASCII  <377>/            -------------------/
(2)  006322  020212  050040  020103              .ASCII  <212>/  PC      CSR     STAT1    STAT2    STAT3/
(2)  006374  026777  026455  026455              .ASCIZ  <377>/------  ------  ------   ------   ------/
(2)  006452  044377  053517  046440      NUM:    .ASCIZ  <377>/HOW MANY DMC11'S TO BE TESTED?/
(2)  006510  041777  051123  040440      CSR:    .ASCIZ  <377>/CSR ADDRESS?/
(2)  006526  053377  041505  047524      VEC:    .ASCIZ  <377>/VECTOR ADDRESS?/
(2)  006547     377  051102  050040      PRIO:   .ASCIZ  <377>/BR PRIORITY LEVEL? (4,5,6,7)?/
(2)  006606  044777  020106  046504      CRAM:   .ASCIZ  <377>/IF DMC HAS CRAM (M8204) TYPE "Y", IF CROM (M8200) TYPE "N"
(2)  006704  053777  044510  044103      MODU:   .ASCIZ  <377>/WHICH LINE UNIT? IF NONE TYPE "N", IF M8201 TYPE "1", IF M
(2)  007016  051777  044527  041524      LINE:   .ASCIZ  <377>/SWITCH PAC#1 (DDCMP LINE #)?/
(2)  007054  051777  044527  041524      BM:     .ASCIZ  <377>/SWITCH PAC#2 (BM873 BOOT ADD)?/
(2)  007114  044777  020123  044124      CONN:   .ASCIZ  <377>/IS THE LOOP BACK CONNECTOR ON?/
(2)  007154  047377  020117  042504      NOACT:  .ASCIZ  <377>/NO DEVICES ARE SELECTED/
(2)  007205     377  051412  051127      SWMES:  .ASCIZ  <377><12>/SWR= /
(2)  007215     116  053505  020077      SWMES1: .ASCIZ  /NEW? /
(2)  007223     377  042377  041515      CONERR: .ASCIZ  <377><377>/DMC11 FOUND AT NON-STANDARD ADDRESS  PC: /
(2)  007277     377  054105  042520      CNERR:  .ASCIZ  <377>/EXPECTED  FOUND/
(2)  007320  024040  046504  024503      DMCM:   .ASCIZ  / (DMC) /
(2)  007330  024040  046513  024503      KMCM:   .ASCIZ  / (KMC) /
(2)  007340  042377  041515  030461      SPEED:  .ASCIZ  <377>/DMC11-AR(REMOTE,LOW SPEED) OR DMC11-AL(LOCAL,HIGH SPEED) T
(2)                                      .EVEN
(2)  007454  000005                      XSTAT0: 5
     007456     006     003                      .BYTE   6,3
     007460  001246                              TEMP1
```

```
     007462     006     003                      .BYTE   6,3
     007464  001250                              TEMP2
     007466     006     003                      .BYTE   6,3
     007470  001252                              TEMP3
     007472     006     003                      .BYTE   6,3
     007474  001254                              TEMP4
     007476     006     002                      .BYTE   6,2
     007500  001256                              TEMP5
                                         .EVEN


                                                 ;BUFFERS FOR INPUT-OUTPUT

     007502  000000                      INBUF:  0
             007544                      .=.+40
     007544  000000                      MDATA:  0
             007606                      .=.+40



                                                 ;ROUTINE USED TO CHANGE SOFTWARE SWITCH
                                                 ;REGISTER USING THE CONSOLE TERMINAL
                                                 ;-----------------------------------------

     007606  022737  000176  001202      CKSWR:  CMP     #SWREG,SWR      ;IS THE SOFT SWR BEING USED?
     007614  001077                              BNE     CKSWR5          ;BR IF NO
     007616  105777  171362                      TSTB    @TKCSR          ;IS DONE SET?
     007622  100003                              BPL     2$              ;GO ON IF NOT SET
     007624  012737  177777  003734              MOV     #-1,DONE        ;IF DONE SET, SET FLAG
     007632  022777  000007  171346      2$:     CMP     #7,@TKDBR       ;WAS CTRL G TYPED? (7 BIT ASCII)
     007640  001404                              BEQ     1$              ;BR IF YES
     007642  022777  000207  171336              CMP     #207,@TKDBR     ;WAS CTRL G TYPED? (8 BIT ASCII)
     007650  001061                              BNE     CKSWR5          ;BR IF NO
     007652  010246                      1$:     MOV     R2,-(SP)        ;STORE R2
     007654  010346                              MOV     R3,-(SP)        ;STORE R3
     007656  010446                              MOV     R4,-(SP)        ;STORE R4
     007660  012737  177777  010016              MOV     #-1,SWFLG       ;SET SOFT TYPE OUT FLAG
     007666  005002                      CKSWR1: CLR     R2              ;CLEAR NEW SWR CONTENTS
     007670  012704  177777                      MOV     #-1,R4          ;SET FLAG TO ALL ONES
     007674  104402  007205                      TYPE    ,SWMES          ;TYPE "SWR= "
     007700  104411                      CKSWR2: CNVRT                   ;TYPE OUT PRESENT CONTENTS
     007702  010052                              SOFTSW                  ;OF SOFT SWITCH REGISTER
     007704  104402  007215              CKSWR3: TYPE    ,SWMES1         ;TYPE "NEW? "
     007710  004737  010020              CKSWR4: JSR     PC,INCHAR       ;GET RESPONSE
     007714  022703  000015                      CMP     #15,R3          ;WAS IT A CR?
     007720  001424                              BEQ     5$              ;BR IF YES
     007722  022703  000012                      CMP     #12,R3          ;WAS IT A LF?
     007726  001416                              BEQ     4$              ;BR IF YES
     007730  022703  000025                      CMP     #25,R3          ;WAS IT CTRL U?
     007734  001754                              BEQ     CKSWR1          ;BR IF YES(START OVER)
     007736  022703  000007                      CMP     #7,R3           ;IF CNTL G GET NEXT CHAR
     007742  001762                              BEQ     CKSWR4
     007744  005004                              CLR     R4              ;IT MUST BE A DIGIT SO CLR FLAG
     007746  042703  177770                      BIC     #177770,R3      ;ONLY 0-7 ARE LEGAL SO MASK OFF BITS
     007752  006302                              ASL     R2              ;SHIFT R2 3 TIMES
     007754  006302                              ASL     R2
     007756  006302                              ASL     R2
     007760  050302                              BIS     R3,R2           ;ADD LAST DIGIT
```

```
007762  000757                              BR      CKSWR4             ;GET NEXT CHARACTER
007764  012766  002002  000006      4$:     MOV     #.START,6(SP)      ;LF WAS TYPED SO GO TO START
007772  005704                      5$:     TST     R4                 ;IS FLAG CLEAR?
007774  001002                              BNE     6$                 ;IF NOT DON'T CHANGE SOFT SWR
007776  010277  171200                      MOV     R2,@SWR            ;IF YES THEN WRITE NEW CONTENTS TO SOFT SWR
010002  005037  010016              6$:     CLR     SWFLG              ;CLEAR TYPEOUT FLAG
010006  012604                              MOV     (SP)+,R4           ;RESTORE R4
010010  012603                              MOV     (SP)+,R3           ;RESTORE R3
010012  012602                              MOV     (SP)+,R2           ;RESTORE R2
010014  000207                      CKSWR5: RTS     PC                 ;RETURN

010016  000000                      SWFLG:  0

010020  105777  171160              INCHAR: TSTB    @TKCSR
010024  100375                              BPL     .-4
010026  017703  171154                      MOV     @TKDBR,R3
010032  105777  171152                      TSTB    @TPCSR
010036  100375                              BPL     .-4
010040  010377  171146                      MOV     R3,@TPDBR
010044  042703  000200                      BIC     #BIT7,R3
010050  000207                              RTS     PC

010052  000001                      SOFTSW: 1
010054     006     002                      .BYTE   6,2
010056  000176                              SWREG
```

```

                                            ;
                                            ;ROUTINE USED TO "CYCLE" THROUGH UP TO 16 DMC11'S
                                            ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
                                            ;AND RUNS THE SPECIFIED DMC11'S.   THIS ROUTINE *MUST*
                                            ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
                                            ;SETUP NECESSARY.
                                            ;

010060  005737  001306              CYCLE:  TST     DMACTV             ;ARE ANY DMC11'S TO BE TESTED?
010064  001004                              BNE     1$                 ;BR IF OK.
010066  104402  007154                      TYPE    ,NOACT             ;NO DMC11'S SELECTED!!
010072  000000                              HALT                       ;STOP THE SHOW.
010074  000776                              BR      .-2                ;DISQUALIFY CONT. SW.
010076  000241                      1$:     CLC                        ;CLEAR PROC. CARRY BIT.
010100  006137  001316                      ROL     RUN                ;UPDATE POINTER
010104  005537  001316                      ADC     RUN                ;CATCH CARRY FROM RUN
010110  062737  000004  001322              ADD     #4,MILK            ;UPDATE POINTER
010116  062737  000010  001320              ADD     #10,CREAM          ;UPDATE ADDRESS POINTER.
010124  022737  001700  001320              CMP     #DM.MAP+200,CREAM
010132  001006                              BNE     2$                 ;KEEP GOING; NOT ALL TESTED FOR.
010134  012737  001500  001320              MOV     #DM.MAP,CREAM      ;RESET ADDRESS POINTER.
010142  012737  001702  001322              MOV     #CNT.MAP,MILK      ;RESET PASS COUNT POINTER
010150  033737  001316  001306      2$:     BIT     RUN,DMACTV         ;IS THIS ONE ACTIVE?
010156  001747                              BEQ     1$                 ;BR IF NO
010160  013700  001320                      MOV     CREAM,R0           ;GET ADDRESS POINTER
010164  013702  001322                      MOV     MILK,R2            ;GET PASS COUNT POINTER
010170  012037  001404                      MOV     (R0)+,DMCSR        ;LOAD SYSTEM CTRL. REG
010174  011037  001374                      MOV     (R0),DMRVEC        ;LOAD VECTOR
010200  042737  177000  001374              BIC     #177000,DMRVEC     ;CLEAR UNWANTED BITS
010206  012037  001366                      MOV     (R0)+,STAT1        ;LOAD STAT1
010212  012037  001370                      MOV     (R0)+,STAT2        ;LOAD STAT2
010216  012037  001372                      MOV     (R0)+,STAT3        ;LOAD STAT3
010222  012237  001230                      MOV     (R2)+,PASCNT       ;LOAD PASS COUNT
010226  012237  001232                      MOV     (R2)+,ERRCNT       ;LOAD ERROR COUNT
010232  012700  000002                      MOV     #2,R0              ;SAVE CORE THIS WAY!
010236  013737  001404  001406              MOV     DMCSR,DMCSRH
010244  005237  001406                      INC     DMCSRH
010250  013737  001406  001410              MOV     DMCSRH,DMCTL
010256  005237  001410                      INC     DMCTL
010262  013737  001410  001412              MOV     DMCTL,DMPO4
010270  060037  001412                      ADD     R0,DMPO4
010274  013737  001412  001414              MOV     DMPO4,DMPO6
010302  060037  001414                      ADD     R0,DMPO6

010306  013737  001374  001376              MOV     DMRVEC,DMRLVL      ;PTY LVL
010314  060037  001376                      ADD     R0,DMRLVL          ;
010320  013737  001376  001400              MOV     DMRLVL,DMTVEC      ;TX VEC
010326  060037  001400                      ADD     R0,DMTVEC          ;
010332  013737  001400  001402              MOV     DMTVEC,DMTLVL      ;TX LVL
010340  060037  001402                      ADD     R0,DMTLVL

010344  032737  000002  001236              BIT     #SW01,STRTSW       ;IS TEST NO. SELECTED
010352  001450                              BEQ     7$                 ;BR IF NO
010354                              4$:
010354  005737  000042                      TST     @#42               ;RUNNING IN AUTO MODE?
```

```
010360  001045                          BNE     7$              ;BR IF YES
010362  104402  005672                  TYPE    ,MCRLF
010366  104403                          INSTR                   ;GET TEST NO.
010370  036130                          MTSTN
010372  104405                          PARAM
010374  000001                          1
010376  001000                          1000
010400  001226                          TSTNO
010402     000                  .BYTE   0
010403     001                  .BYTE   1
010404  012700  012320                  MOV     #TST1,R0
010410  022710                  5$:     CMP     (PC)+,(R0)      ;CMP FIRST WORD TO 12737
010412  012737                          MOV     (PC)+,@(PC)+
010414  001020                          BNE     6$              ;BR IF NOT SAME
010416  023760  001226  000002          CMP     TSTNO,2(R0)     ;DOES TSTNO MATCH?
010424  001014                          BNE     6$              ;BR IF NO
010426  022760  001226  000004          CMP     #TSTNO,4(R0)    ;IS LAST WORD OK?
010434  001010                          BNE     6$              ;BR IF NO
010436  010037  001214                  MOV     R0,RETURN       ;IT IS A LEGAL TEST SO DO IT
010442  104402  005755                  TYPE    ,MR
010446  042737  000002  001236          BIC     #SW01,STRTSW
010454  000412                          BR      8$
010456  005720                  6$:     TST     (R0)+           ;POP R0
010460  020027  016224                  CMP     R0,#TLAST+10    ;AT END YET?
010464  001351                          BNE     5$              ;BR IF NO
010466  104402  005666                  TYPE    ,MQM            ;YES ILLEGAL TEST NO.
010472  000730                          BR      4$              ;TRY AGAIN

010474  012737  012320  001214  7$:     MOV     #TST1,RETURN    ;PREPARE RETURN ADDRESS
010502  013701  001404          8$:     MOV     DMCSR,R1        ;R1 = BASE DMC11 ADDRESS
010506  000177  170502                  JMP     @RETURN         ;GO START TESTING.


                                        ;ROUTINE USED TO "AUTO SIZE" THE DMC11
                                        ;CSR AND VECTOR.
                                        ;NOTE:  THE CSR MAY BE ANY WHERE IN THE FLOATING
                                        ;       ADDRESS RANGE (160000:164000)
                                        ;       AND THE VECTOR MAY BE ANY WHERE IN THE
                                        ;       FLOATING VECTOR RANGE (300:770)
                                        ;

010512                          AUTO.SIZE:
010512  000005                          RESET                   ;INSURE A BUS INIT.
010514  012702  001500          CSRMAP: MOV     #DM.MAP,R2      ;LOAD MAP POINTER.
010520  005022                  1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
010522  022702  001700                  CMP     #DM.END,R2      ;ALL DONE?
010526  001374                          BNE     1$              ;BR IF NO
010530  005037  001310                  CLR     DMNUM           ;SET OCTAL NUMBER OF DMC11'S TO 0
010534  012702  001500                  MOV     #DM.MAP,R2      ;R2 POINTS TO DMC MAP
010540  005037  001306                  CLR     DMACTV          ;CLEAR ACTIVE
010544  032737  000001  001236          BIT     #SW00,STRTSW    ;QUESTIONS?
010552  001002                          BNE     .+6             ;BR IF YES
010554  000137  011252                  JMP     7$              ;IF NO SKIP QUESTIONS
010560  012737  000001  001256          MOV     #1,TEMP5        ;START WITH 1
010566  104403                          INSTR
010570  006450                          NUM
```

```
010572  104405                          PARAM
010574  000001                          1
010576  000020                          16.
010600  001252                          TEMP3
010602     000                          .BYTE   0
010603     001                          .BYTE   1
010604  013737  001252  001310          MOV     TEMP3,DMNUM     ;DMNUM = HOW MANY
010612  104402  005672          12$:    TYPE    ,MCRLF
010616  104410                          CONVRT                  ;TYPE WHICH DMC IS BEING DONE
010620  012602                          WHICH                   ;TEMP5 IS WHICH DMC
010622  005237  001256                  INC     TEMP5
010626  104403                          INSTR
010630  006510                          CSR
010632  104405                          PARAM
010634  160000                          160000
010636  164000                          164000
010640  001254                          TEMP4
010642     000                          .BYTE   0
010643     001                          .BYTE   1
010644  013722  001254                  MOV     TEMP4,(R2)+     ;STORE CSR IN MAP
010650  104403                          INSTR
010652  036526                          VEC
010654  104405                          PARAM
010656  000000                          0
010660  000776                          776
010662  001254                          TEMP4
010664     000                          .BYTE   0
010665     001                          .BYTE   1
010666  013712  001254                  MOV     TEMP4,(R2)      ;STORE VECTOR IN MAP
010672  104402                  10$:    TYPE
010674  036547                          PRIO                    ;ASK WHAT BR LEVEL
010676  004737  012266                  JSR     PC,INTTY        ;GET RESPONSE
010702  022703  000024                  CMP     #24,R3          ;
010706  101014                          BHI     50$             ;BR IF LESS THAN 4
010710  022703  000027                  CMP     #27,R3          ;
010714  103411                          BLO     50$             ;BR IF GREATER THAN 7
010716  012704  000011                  MOV     #11,R4          ;R4 = NUMBER OF SHIFTS
010722  006303                          ASL     R3              ;SHIFT R3 LEFT
010724  005304                          DEC     R4              ;DEC SHIFT COUNT
010726  001375                          BNE     .-4             ;BR IF NOT DONE
010730  042703  170777                  BIC     #170777,R3      ;BIC UNWANTED BITS
010734  050312                          BIS     R3,(R2)         ;PUT BR LEVEL IN STATUS MAP
010736  000403                          BR      8$              ;CONTINUE
010740  104402                  50$:    TYPE
010742  005666                          MQM                     ;RESPONSE IS OUT OF LIMITS
010744  000752                          BR      10$             ;TRY AGAIN
010746  104402                  8$:     TYPE
010750  006606                          CRAM                    ;DOES DMC HAVE CRAM?
010752  004737  012266                  JSR     PC,INTTY        ;GET REPLY
010756  022703  000131                  CMP     #131,R3
010762  001427                          BEQ     9$              ;YES
010764  022703  000116                  CMP     #116,R3         ;NO
010770  001403                          BEQ     40$             ;NOT A Y OR N
010772  104402                          TYPE
010774  005666                          MQM                     ;TYPE "?"
010776  000763                          BR      8$              ;ASK AGAIN
```

```
011000  104402                          40$:    TYPE
011002  007340                                  SPEED                    ;DMC11-AR OR DMC11-AL?
011004  004737  012266                          JSR     PC,INTTY         ;GET RESPONSE
011010  022703  000122                          CMP     #122,R3          ;IS IT R
011014  001414                                  BEQ     16$              ;BR IF PEMOTE
011016  022703  000114                          CMP     #114,R3          ;IS IT L
011022  001403                                  BEQ     41$              ;BR IF LOCAL
011024  104402                                  TYPE
011026  005666                                  MQM
011030  000763                                  BR      40$              ;TRY AGAIN
011032  052762  000002  000004          41$:    BIS     #BIT1,4(R2)      ;SET BIT1 IN STAT3
011040  000402                                  BR      16$              ;CONTINUE
011042  052712  100000                  9$:     BIS     #BIT15,(R2)      ;SET BIT 15 IF CRAM
011046  104402                          16$:    TYPE
011050  006704                                  MODU                     ;ASK WHICH LINE UNIT
011052  004737  012266                          JSR     PC,INTTY         ;GET REPLY
011056  022703  000021                          CMP     #21,R3           ;"1"
011062  001417                                  BEQ     30$
011064  022703  000022                          CMP     #22,R3           ;"2"
011070  001412                                  BEQ     31$
011072  022703  000116                          CMP     #116,R3          ;"N"
011076  001403                                  BEQ     32$
011100  104402                                  TYPE
011102  005666                                  MQM                      ;IF NOT A 1,2 OR N TYPE "?"
011104  000760                                  BR      16$              ;TRY AGIAN
011106  052722  010000                  32$:    BIS     #BIT12,(R2)+     ;SET BIT 12 IN STAT2 IF NO LU
011112  022222                                  CMP     (R2)+,(R2)+      ;POP OVER STAT2 AND STAT3
011114  000447                                  BR      33$
011116  052712  020000                  31$:    BIS     #BIT13,(R2)      ;SET BIT 13 IN STAT2 IF M8202
011122  104402                          30$:    TYPE
011124  007114                                  CONN                     ;ASK IF LOOP-BACK IS ON
011126  004737  012266                          JSR     PC,INTTY         ;GET REPLY
011132  022703  000131                          CMP     #131,R3          ;Y
011136  001406                                  BEQ     17$
011140  022703  000116                          CMP     #116,R3          ;N
011144  001406                                  BEQ     18$
011146  104402                                  TYPE
011150  005666                                  MQM                      ;IF NOT Y OR N TYPE "?"
011152  000763                                  BR      30$              ;TRY AGAIN
011154  052722  040000                  17$:    BIS     #BIT14,(R2)+     ;TURNAROUND IS CONNECTED
011160  000402                                  BR      19$
011162  042722  040000                  18$:    BIC     #BIT14,(R2)+     ;NO TURNAROUND
011166                                  19$:
011166  104403                                  INSTR
011170  007016                                  LINE
011172  104405                                  PARAM
011174  000000                                  0
011176  000377                                  377
011200  001254                                  TEMP4
011202     000                                  .BYTE   0
011203     001                                  .BYTE   1
011204  113722  001254                          MOVB    TEMP4,(R2)+      ;STORE SWITCH PAC IN MAP
011210  104403                                  INSTR
011212  047054                                  BM
011214  104405                                  PARAM
011216  000000                                  0
```

```
011220  000377                                  377
011222  001254                                  TEMP4
011224     000                                  .BYTE   0
011225     001                                  .BYTE   1
011226  113722  001254                          MOVB    TEMP4,(R2)+      ;STORE SWITCH PAC IN MAP
011232  005722                                  TST     (R2)+            ;POP OVER STAT3
011234  005337  001252                  33$:    DEC     TEMP3            ;DEC DMC COUNT
011240  001402                                  BEQ     34$              ;BR IF DONE
011242  000137  010612                          JMP     12$              ;JUMP IF NOT
011246  000137  011702                  34$:    JMP     13$              ;CONTINUE
011252  012701  160000                  7$:     MOV     #160000,R1       ;SET FOR FIRST ADDRESS TO BE TESTED
011256  012737  011774  000004          MOV     #6$,@#4          ;SET FOR NON-EXISTANT DEVICE TIME OUT
011264  005011                          2$:     CLR     (R1)             ;CLEAR SEL0
011266  005711                                  TST     (R1)             ;IF DMC11 DMCSR S/B 0
011270  001172                                  BNE     3$               ;IF NO DEV ; TRAP TO 4, IF NO BIT 8 THEN NO DMC1
011272  005061  000006                          CLR     6(R1)            ;CLEAR SEL6
011276  005761  000006                          TST     6(R1)            ;IF DMC11 THEN DMRIC S/B =0!
011302  001165                                  BNE     3$               ;BR IF NOT DMC11
011304  012711  002000                          MOV     #BIT10,(R1)      ;SET ROMO
011310  005061  000004                          CLR     4(R1)            ;CLEAR SEL4
011314  012761  125252  000006                  MOV     #125252,6(R1)    ;WRITE THIS TO SEL6
011322  052711  020000                          BIS     #BIT13,(R1)      ;WRITE IT!
011326  022761  125252  000004                  CMP     #125252,4(R1)    ;WAS IT WRITTEN?
011334  001004                                  BNE     21$              ;IF NO IT IS NOT CRAM
011336  052762  100000  000002                  BIS     #BIT15,2(R2)     ;SET BIT15 IF CRAM
011344  000431                                  BR      22$
011346  012711  001000                  21$:    MOV     #BIT9,(R1)       ;SET ROMI
011352  012761  100417  000006                  MOV     #100417,6(R1)    ;PUT INSTRUCTION IN SEL6
011360  012711  001400                          MOV     #BIT9!BIT8,(R1)  ;CLOCK INSTRUCTION (MICRO PROC PC TO 0)
011364  012711  002000                          MOV     #BIT10,(R1)      ;SET ROMO
011370  022761  000626  000006                  CMP     #626,6(R1)       ;IS IT LOCAL CROM
011376  001411                                  BEQ     23$              ;BR IF YES
011400  022761  016520  000006                  CMP     #16520,6(R1)     ;IS IT REMOTE CROM?
011406  001410                                  BEQ     22$              ;BR IF YES
011410  022761  177777  000006                  CMP     #-1,6(R1)        ;NO CROM?
011416  001404                                  BEQ     22$              ;BR IF YES
011420  000516                                  BR      3$               ;NOT A DMC
011422  052762  000002  000006          23$:    BIS     #BIT1,6(R2)      ;SET BIT 1 IN STAT3
                                        ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DMC11 CSR ADDRESS.
011430  010122                          22$:    MOV     R1,(R2)+         ;STORE CSR IN CORE TABLE.
011432  012711  001000                  15$:    MOV     #BIT9,(R1)       ;CLEAR LINE UNIT LOOP
011436  005061  000004                          CLR     4(R1)            ;CLEAR PORT4
011442  012761  122113  000006                  MOV     #122113,6(R1)    ;LOAD INSTRUCTION (CLR DTR)
011450  052711  000400                          BIS     #BIT8,(R1)       ;CLOCK INSTRUCTION
011454  012761  021264  000006                  MOV     #021264,6(R1)    ;LOAD INSTRUCTION
011462  052711  000400                          BIS     #BIT8,(R1)       ;CLOCK INSTRUCTION
011466  122761  000377  000004                  CMPB    #377,4(R1)       ;IS IT ALL ONES?
011474  001003                                  BNE     .+10             ;BR IF NO
011476  052712  010000                          BIS     #BIT12,(R2)      ;IF YES, NO LINE UNIT, SET STATUS BIT
011502  000436                                  BR      20$
011504  032761  000002  000004                  BIT     #BIT1,4(R1)      ;IS SWITCH A ONE?
011512  001403                                  BEQ     .+10             ;BR IF M8201
011514  052712  060000                          BIS     #BIT13!BIT14,(R2) ;M8202 ASSUME CONNECTOR
011520  000427                                  BR      20$              ;CONNECTOR ON)
011522  032761  000010  000004                  BIT     #BIT3,4(R1)      ;IS MRDY SET
011530  001023                                  BNE     20$              ;BR IF M8201 NO CONNECTOR (ON LINE)
```

```
011532  012761  000100  000004            MOV     #BIT6,4(R1)        ;LOAD PORT4
011540  012761  122113  000006            MOV     #122113,6(R1)      ;LOAD INSTRUCTION
011546  052711  000400                    BIS     #BIT8,(R1)         ;CLOCK INSTRUCTION(SET DTR)
011552  012761  021264  000006            MOV     #021264,6(R1)      ;LOAD INSTRUCTION
011560  052711  000400                    BIS     #BTT8,(R1)         ;CLOCK INSTRUCTION(READ MODEM REG)
011564  032761  000010  000004            BIT     #BIT3,4(R1)        ;IS MRDY SET NOW?
011572  001402                            BEQ     20$                ;BR IF NO CONNECTOR
011574  052712  040000                    BIS     #BIT14,(R2)        ;SET STATUS BIT FOR CONNECTOR
011600  005722                    20$:    TST     (R2)+              ;POP POINTER
011602  012761  021324  000006            MOV     #021324,6(R1)      ;PUT INSTRUCTION IN PORT6
011610  012711  001400                    MOV     #BIT9!BIT8,(R1)    ;PORT4_LU 15
011614  156122  000004                    BISB    4(R1),(R2)+        ;STORE DDCMP LINE # IN TABLE
011620  012761  021344  000006            MOV     #021344,6(R1)      ;PORT6_INSTRUCTION
011626  012711  001400                    MOV     #BIT8!BIT9,(R1)    ;CLOCK INSTR.
011632  156122  000004                    BISB    4(R1),(R2)+        ;STORE BM873 ADD IN TABLE
011636  005722                            TST     (R2)+              ;POP OVER STAT3
011640  005011                            CLR     (R1)               ;CLEAR ROM1
011642  005237  001310                    INC     DMNUM              ;UPDATE DEVICE COUNTER
011646  022737  000020  001310            CMP     #20,DMNUM          ;ARE MAX. NO. OF DEV FOUND?
011654  001412                            BEQ     13$                ;YES DON'T LOOK FOR ANY MORE.
011656  005011                    3$:     CLR     (R1)               ;CLEAR BIT 10
011660  005061  000006                    CLR     6(R1)              ;CLEAR SEL 6
011664  062701  000010            14$:    ADD     #10,R1             ;UPDATE CSR POINTER ADDRESS
011670  022701  164000                    CMP     #164000,R1
011674  001402                            BEQ     13$                ;BR IF DONE
011676  000137  011264                    JMP     2$                 ;JUMP IF NOT
011702  005037  001306            13$:    CLR     DMACTV
011706  005737  001310                    TST     DMNUM              ;WERE ANY DMC11'S FOUND AT ALL?
011712  001423                            BEQ     5$                 ;ERROR AUTO SIZER FOUND NO DMC11'S IN THIS SYS.
011714  013701  001310                    MOV     DMNUM,R1
011720  010137  001314                    MOV     R1,SAVNUM          ;SAVE NUMBER OF DEVICES
011724  000241                    4$:     CLC
011726  006137  001306                    ROL     DMACTV             ;GENERATE ACTIVE REGISTER OF DEVICES.
011732  005237  001306                    INC     DMACTV             ;SET THE BIT
011736  005301                            DEC     R1
011740  001371                            BNE     4$                 ;BR IF MORE TO GENERATE
011742  012737  000006  000004            MOV     #6,@#4             ;RESTORE TRAP VECTOR
011750  013737  001306  001312            MOV     DMACTV,SAVACT      ;SAVE ACTIVE REGISTER
011756  000137  012010                    JMP     VECMAP             ;GO FIND THE VECTOR NOW.
011762  104402  005760            5$:     TYPE    ,MERR2             ;NOTIFY OPR THAT NO DMC11'S FOUND.
011766  005000                            CLR     R0                 ;MAKE DATA LIGHTS ZERO
011770  000000                            HALT                       ;STOP THE SHOW
011772  000776                            BR      .-2                ;DISABLE CONT. SW.
011774  012716  011664            6$:     MOV     #14$,(SP)          ;ENTERED BY NON-EXISTANT TIME-OUT.
012000  000002                            RTI                        ;RETURN TO MAINSTREAM

012002  000001                    WHICH:  1
012004     002     002                    .BYTE   2,2
012006  001256                            TEMP5

012010  032737  000001  001236    VECMAP: BIT     #SW00,STRTSW
012016  001114                            BNE     5$
012020  012737  000340  000022            MOV     #340,@#22          ;SET IOT TRAP PRIO TO 7
012026  012737  012202  000020            MOV     #4$,@#20           ;SET IOT TRAP VECTOR
012034  012702  001500                    MOV     #DM.MAP,R2         ;SET SOFTWARE POINTER
012040  012700  000300                    MOV     #300,R0            ;FLOATING VECTORS START HERE.
```

```
012044  012701  000302                    MOV     #302,R1            ;PC OF IOT INSTR.
012050  010120                    1$:     MOV     R1,(R0)+           ;START FILLING VECTOR AREA
012052  012721  000004                    MOV     #4,(R1)+           ;WITH .+2; IOT
012056  022021                            CMP     (R0)+,(R1)+        ;ADD 2 TO R0 +R1
012060  020127  001000                    CMP     R1,#1000
012064  101771                            BLOS    1$                 ;BR IF MORE TO FILL
012066  013737  001306  001246            MOV     DMACTV,TEMP1       ;STORE TEMPORALLY
012074  006037  001246            2$:     ROR     TEMP1              ;BRING OUT A BIT
012100  103063                            BCC     5$                 ;BR IF ALL DONE
012102  012704  000012                    MOV     #12,R4             ;R4 IS INDEX REGISTER
012106  016437  012252  177776            MOV     BRLVL(R4),PS       ;SET PS TO 7
012114  011201                            MOV     (R2),R1
012116  012761  000200  000004            MOV     #200,4(R1)
012124  012711  001000                    MOV     #BIT9,(R1)         ;SET ROMI
012130  012761  121111  000006            MOV     #121111,6(R1)      ;PUT INSTRUCTION IN PORT6
012136  012711  001400                    MOV     #BIT9!BIT8,(R1)    ;FORCE AN INTERRUPT
012142  105200                    7$:     INCB    R0                 ;STALL
012144  001376                            BNE     .-2                ;FOR TIME TO INTERUPT
012146  162704  000002                    SUB     #2,R4              ;GET NEXT LOWEST PS LEVEL
012152  001404                            BEQ     6$                 ;BR IF R4 = 0
012154  016437  012252  177776            MOV     BRLVL(R4),PS       ;MOVE NEXT LOWER LEVEL IN PS
012162  000767                            BR      7$                 ;BR TO DELAY
012164  052762  005300  000002    6$:     BIS     #5300,2(R2)        ;NO INTERUPT ASSUME 300 AT LEVEL 5 AND FIX DMC11
012172  005011                    3$:     CLR     (R1)               ;CLEAR ROMI
012174  062702  000010                    ADD     #10,R2             ;POP SOFTWARE POINTER
012200  000735                            BR      2$                 ;KEEP GOING
012202  051662  000002            4$:     BIS     (SP),2(R2)         ;GET VECTOR ADDRESS
012206  042762  000007  000002            BIC     #7,2(R2)           ;CLEAR JUNK
012214  016405  012254                    MOV     BRLVL+2(R4),R5     ;GET BR LEVEL OF DMC11
012220  006305                            ASL     R5                 ;SHIFT LEVEL 4 PLACES
012222  006305                            ASL     R5                 ;TO THE LEFT FOR THE
012224  006305                            ASL     R5                 ;STATUS TABLE
012226  006305                            ASL     R5
012230  042705  170777                    BIC     #170777,R5         ;CLEAR UNWANTED BITS
012234  050562  000002                    BIS     R5,2(R2)           ;PUT BR LEVEL IN STATUS TABLE
012240  022626                            CMP     (SP)+,(SP)+        ;POP IOT JUNK OFF STACK
012242  012716  012172                    MOV     #3$,(SP)           ;SET FOR RETURN
012246  000002                            RTI
012250  000207                    5$:     RTS     PC                 ;ALL DONE WITH "AUTO SIZING"

012252  000000                    BRLVL:  0       ;LEVEL 0
012254  000000                            0       ;LEVEL 0
012256  000200                            200     ;LEVEL 4
012260  000240                            240     ;LEVEL 5
012262  000300                            300     ;LEVEL 6
012264  000340                            340     ;LEVEL 7


012266  105777  166712            INTTY:  TSTB    @TKCSR             ;WAIT FOR DONE
012272  100375                            BPL     .-4
012274  017703  166706                    MOV     @TKDBR,R3          ;PUT CHAR IN R3
012300  105777  166704                    TSTB    @TPCSR             ;WAIT UNTIL PRINTER IS READY
012304  100375                            BPL     .-4
012306  010377  166700                    MOV     R3,@TPDBR          ;ECHO CHAR
012312  042703  000240                    BIC     #BIT7!BIT5,R3      ;MASK OFF LOWER CASE
012316  000207                            RTS     PC                 ;RETURN
```

```
 1663
```

```
 1664
 1665
 1666                                                          ;*************************** TEST 1 **************************
 1667                                                          ;*FREE RUNNING FLAG MODE DATA TEST
 1668                                                          ;*TRANSMIT A MESSAGE AND VERIFY THE RECEIVED DATA
 1669                                                          ;*LINE UNIT LOOP IS SET FOR THIS TEST.
 167Ø                                                          ;*ALL FOLLOWING TESTS ARE FREE RUNNING AND ARE PERFORMED
 1671                                                          ;*ONLY ON DMC'S WITH LINE UNITS. IF YOU WISH TO PERFORM
 1672                                                          ;*THESE FREE RUNNING TESTS ON A KMC (NORMALLY THE FREE RUNNING TESTS
 1673                                                          ;*WILL FAIL ON A KMC, THE TIMER IS TOO FAST) THEN YOU MUST
 1674                                                          ;*MANUALLY SET BITØ OF STAT3 IN THE STATUS MAP. ALSO THE KMC
 1675                                                          ;*MUST HAVE THE MICRO-CODE LOADED BY PREVIOUSLY RUNNING
 1676                                                          ;*DZDMG TEST 2 AND THEN LOADING AND STARTING DZDMH
 1677                                                          ;* WITH SWITCH 7 = 1
 1678                                                          ;**********************************************************
 1679
 168Ø                                                          ;  TEST 1
 1681                                                          ;--------------
 1682  Ø1232Ø  Ø12737  ØØØØØ1  ØØ1226                TST1:     MOV     #1,TSTNO
 1683  Ø12326  Ø12737  Ø134Ø4  ØØ1216                          MOV     #TST2,NEXT
 1684                                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
 1685  Ø12334  Ø32737  1ØØØØØ  ØØ1366                          BIT     #BIT15,STAT1     ;IS IT A DMC?
 1686  Ø12342  ØØ14Ø6                                          BEQ     .+16             ;BR IF YES
 1687  Ø12344  Ø32737  ØØØØØ1  ØØ1372                          BIT     #BITØ,STAT3      ;KMC WITH BITØ SET?
 1688  Ø12352  ØØ1ØØ2                                          BNE     .+6              ;BR IF YES
 1689  Ø12354  ØØØ137  Ø134Ø2                                  JMP     14$              ;SKIP TEST
 169Ø  Ø1236Ø  Ø32737  Ø1ØØØØ  ØØ1366                          BIT     #BIT12,STAT1     ;LU PRESENT?
 1691  Ø12366  ØØ1372                                          BNE     .-12             ;BR IF NO
 1692  Ø1237Ø  Ø137ØØ  Ø2137Ø                                  MOV     RCOUNT,RØ        ;CLEAR RECEIVER BUFFER
 1693  Ø12374  Ø627ØØ  ØØØØØ2                                  ADD     #2,RØ            ;CLEAR 2 MORE LOCATIONS
 1694  Ø124ØØ  Ø127Ø2  Ø21372                                  MOV     #RBUF,R2         ;CLEAR OUT RECEIVE BUFFER
 1695  Ø124Ø4  1Ø5Ø22                                10$:      CLRB    (R2)+            ;CLEAR BUFFER
 1696  Ø124Ø6  ØØ53ØØ                                          DEC     RØ               ;DONE YET!
 1697  Ø1241Ø  ØØ1375                                          BNE     1Ø$              ;NO
 1698  Ø12412  ØØ5Ø37  Ø21316                                  CLR     TFLAG            ;SET TFLAG TO Ø
 1699  Ø12416  ØØ5Ø37  Ø2132Ø                                  CLR     RFLAG            ;SET RFLAG TO Ø
 17ØØ  Ø12422  Ø12711  Ø4ØØØØ                                  MOV     #BIT14,(R1)      ;MASTER CLEAR
 17Ø1  Ø12426  Ø32737  1ØØØØØ  ØØ1366                          BIT     #BIT15,STAT1     ;CRAM?
 17Ø2  Ø12434  ØØ14Ø2                                          BEQ     .+6              ;BR IF NO
 17Ø3  Ø12436  Ø12711  1ØØØØØ                                  MOV     #BIT15,(R1)      ;IF CRAM SET RUN
 17Ø4  Ø12442  1Ø5227  ØØØØØØ                                  INCB    #Ø               ;DELAY
 17Ø5  Ø12446  ØØ1375                                          BNE     .-4              ;DELAY
 17Ø6  Ø1245Ø  ØØ5Ø37  ØØ1416                                  CLR     TEMP             ;GET SET TO DELAY
 17Ø7  Ø12454  ØØ5711                                1$:       TST     (R1)             ;RUN SET?
 17Ø8  Ø12456  1ØØ4Ø5                                          BMI     .+14             ;BR IF YES
 17Ø9  Ø1246Ø  ØØ5237  ØØ1416                                  INC     TEMP             ;INC DELAY
 171Ø  Ø12464  ØØ1373                                          BNE     1$               ;BR IF NOT DONE
 1711  Ø12466  1Ø4Ø14                                          HLT     14               ;ERROR RUN NOT SET
 1712  Ø1247Ø  ØØØ771                                          BR      1$               ;TRY AGAIN
 1713  Ø12472  Ø52711  ØØ4Ø43                                  BIS     #4Ø43,(R1)       ;BASE I, LU LOOP
 1714  Ø12476  ØØ5Ø37  ØØ1416                                  CLR     TEMP             ;GET SET TO DELAY
 1715  Ø125Ø2  1Ø5711                                2$:       TSTB    (R1)             ;RDI SET?
 1716  Ø125Ø4  1ØØ4Ø4                                          BMI     .+12             ;BR IF YES
 1717  Ø125Ø6  ØØ5237  ØØ1416                                  INC     TEMP             ;INC DELAY
 1718  Ø12512  ØØ1373                                          BNE     2$               ;BR IF NOT DONE
 1719  Ø12514  1Ø4Ø14                                          HLT     14               ;ERROR,RDI NOT SET
```

```
012516  012761  021444  000004            MOV     #BASE,4(R1)      ;SET UP BASE ADDRESS
012524  005061  000006                    CLR     6(R1)            ;CLEAR COUNT
012530  142711  000040                    BICB    #40,(R1)         ;CLEAR RQI
012534  005037  001416                    CLR     TEMP             ;GET SET TO DELAY
012540  105711                    3$:     TSTB    (R1)             ;IS RDI GONE?
012542  100020                            BPL     8$               ;BR IF YES
012544  005237  001416                    INC     TEMP             ;INC DELAY
012550  001373                            BNE     3$               ;BR IF NOT DONE
012552  105761  000002                    TSTB    2(R1)            ;IS THERE A CNTL O ERROR
012556  100011                            BPL     18$              ;BR IF NO
012560  016137  000004  001252            MOV     4(R1),TEMP3      ;SAVE SEL4 FOR TYPEOUT
012566  016137  000006  001254            MOV     6(R1),TEMP4      ;SAVE SEL6 FOR TYPEOUT
012574  104016                            HLT     16               ;CNTL O ERROR
012576  000137  013402                    JMP     14$              ;FATAL ERROR STOP
012602  104014                    18$:    HLT     14               ;ERROR RDI STILL SET
012604                            8$:
012604  152711  000041                    BISB    #41,(R1)         ;ASK FOR CNTL I
012610  105711                    64$:    TSTB    (R1)             ;WAIT FOR RDI
012612  100376                            BPL     64$              ;BR IF NOT SETY
012614  005061  000006                    CLR     6(R1)            ;SET FULL DUPLEX
012620  142711  000040                    BICB    #40,(R1)         ;CLEAR RQI
012624  105711                    65$:    TSTB    (R1)             ;RDI UP?
012626  100776                            BMI     65$              ;BR IF YES
012630  152711  000044                    BISB    #44,(R1)         ;REC BA/CC
012634  005037  001416                    CLR     TEMP             ;GET SET TO DELAY
012640  105711                    4$:     TSTB    (R1)             ;IS RDI SET?
012642  100404                            BMI     .+12             ;BR IF YES
012644  005237  001416                    INC     TEMP             ;INC DELAY
012650  001373                            BNE     4$               ;BR IF DELAY NOT DONE
012652  104014                            HLT     14               ;ERROR RDI NOT SET
012654  012761  021372  000004            MOV     #RBUF,4(R1)      ;LOAD REC BA
012662  013761  021370  000006            MOV     RCOUNT,6(R1)     ;LOAD REC COUNT
012670  142711  000040                    BICB    #40,(R1)         ;CLEAR RQI
012674  005037  001416                    CLR     TEMP             ;GET SET TO DELAY
012700  105711                    5$:     TSTB    (R1)             ;RDI GONE?
012702  100004                            BPL     .+12             ;BR IF YES
012704  005237  001416                    INC     TEMP             ;INC DELAY
012710  001373                            BNE     5$               ;BR IF NO DONE
012712  104014                            HLT     14               ;ERROR RDI STILL SET
012714  152711  000040                    BISB    #40,(R1)         ;XMIT BA/CC
012720  005037  001416                    CLR     TEMP             ;GET SET TO DELAY
012724  105711                    6$:     TSTB    (R1)             ;RDI SET?
012726  100404                            BMI     .+12             ;BR IF YES
012730  005237  001416                    INC     TEMP             ;INC DELAY
012734  001373                            BNE     6$               ;BR IF NOT DONE
012736  104014                            HLT     14               ;ERROR RDI NOT SET
012740  012761  021324  000004            MOV     #TBUF,4(R1)      ;LOAD XMIT BUFFER
012746  013761  021322  000006            MOV     TCOUNT,6(R1)     ;LOAD COUNT
012754  142711  000040                    BICB    #40,(R1)         ;CLEAR RQI
012760  005037  001416                    CLR     TEMP             ;GET SET TO DELAY
012764  105711                    7$:     TSTB    (R1)             ;RDI GONE?
012766  100004                            BPL     .+12             ;BR IF YES
012770  005237  001416                    INC     TEMP             ;INC DELAY
012774  001373                            BNE     7$               ;BR IF NOT DONE DELAY
012776  104014                            HLT     14               ;ERROR RDI STILL SET
013000  005037  001416            16$:    CLR     TEMP             ;GET SET TO DELAY
```

```
013004  012737  000022  001246            MOV     #22,TEMP1        ;GET SET FOR LONG DELAY
013012  105761  000002            11$:    TSTB    2(R1)            ;RDO SET?
013016  100407                            BMI     17$              ;BR IF YES
013020  005237  001416                    INC     TEMP             ;INC DELAY
013024  001372                            BNE     11$              ;BR IF DELAY NOT DONE
013026  005337  001246                    DEC     TEMP1            ;DEC DELAY COUNT
013032  001367                            BNE     11$              ;BR IF NOT DONE DELAY
013034  104014                            HLT     14               ;ERROR RDO NOT SET
013036  016137  000002  001250    17$:    MOV     2(R1),TEMP2      ;SAVE SEL2
013044  001001                            BNE     .+4              ;BR IF OK
013046  104014                            HLT     14               ;ERROR!!! SEL2 = 0!!!!!!
013050  032761  000004  000002            BIT     #BIT2,2(R1)      ;REC OR XMIT?
013056  001032                            BNE     13$              ;BR IF REC
013060  005737  021316            12$:    TST     TFLAG            ;FIRST TIME HERE?
013064  001401                            BEQ     .+4              ;BR IF YES
013066  104014                            HLT     14               ;ERROR MULTIPLE XMIT DONES
013070  012737  177777  021316            MOV     #-1,TFLAG        ;SET TFLAG TO -1
013076  132761  000001  000002            BITB    #BIT0,2(R1)      ;IS IT CONTROL O
013104  001401                            BEQ     .+4              ;BR IF NO
013106  104014                            HLT     14               ;XMIT ERROR
013110  022761  021324  000004            CMP     #TBUF,4(R1)      ;XMIT BA CORRECT?
013116  001401                            BEQ     .+4              ;BR IF YES
013120  104014                            HLT     14               ;XMIT BA ERROR
013122  023761  021322  000006            CMP     TCOUNT,6(R1)     ;COUNT OK?
013130  001401                            BEQ     .+4              ;BR IF YES
013132  104014                            HLT     14               ;XMIT COUNT ERROR
013134  142761  000207  000002            BICB    #207,2(R1)       ;CLEAR RDO AND BITS 0-2
013142  000453                            BR      15$              ;CONTINUE
013144  005737  021320            13$:    TST     RFLAG            ;FIRST TIME HERE?
013150  001401                            BEQ     .+4              ;BR IF YES
013152  104014                            HLT     14               ;ERROR MULTIPLE REC DONES
013154  012737  177777  021320            MOV     #-1,RFLAG        ;SET RFLAG TO -1
013162  132761  000001  000002            BITB    #BIT0,2(R1)      ;IS IT CNTL O
013170  001401                            BEQ     .+4              ;BR IF NO
013172  104014                            HLT     14               ;RECEIVE ERROR
013174  022761  021372  000004            CMP     #RBUF,4(R1)      ;REC BA CORRECT?
013202  001401                            BEQ     .+4              ;BR IF YES
013204  104014                            HLT     14               ;REC BA ERROR
013206  023761  021370  000006            CMP     RCOUNT,6(R1)     ;COUNT OK?
013214  001401                            BEQ     .+4              ;BR IF YES
013216  104014                            HLT     14               ;REC COUNT ERROR
013220  013700  021370                    MOV     RCOUNT,R0        ;GET SET TO CHECK DATA
013224  012702  021324                    MOV     #TBUF,R2         ;R2 POINTS TO GOOD DATA
013230  012703  021372                    MOV     #RBUF,R3         ;R3 POINTS TO RECEIVE DATA
013234  010337  001252            9$:     MOV     R3,TEMP3         ;SAVE ADDRESS FOR TYPEOUT
013240  112205                            MOVB    (R2)+,R5         ;R5 = XMIT DATA
013242  112304                            MOVB    (R3)+,R4         ;R4 = RECEIVE DATA
013244  120504                            CMPB    R5,R4            ;CHECK DATA
013246  001401                            BEQ     .+4              ;BR IF OK
013250  104013                            HLT     13               ;DATA ERROR
013252  005300                            DEC     R0               ;DEC COUNT
013254  001367                            BNE     9$               ;BR IF NOT DONE
013256  005713                            TST     (R3)             ;THIS SHOULD BE 0, ELSE
013260  001401                            BEQ     .+4              ;IT RECEIVED TO MUCH!!
013262  104014                            HLT     14               ;ERROR
013264  142761  000207  000002            BICB    #207,2(R1)       ;CLEAR RDO AND BITS 0-2
```

```
013272  005737  021320                  15$:    TST     RFLAG           ;REC DONE?
013276  001640                                  BEQ     16$             ;BR IF NO
013300  005737  021316                          TST     TFLAG           ;XMIT DONE?
013304  001635                                  BEQ     16$             ;BR IF NO
013306  004737  022502                          JSR     PC,SHUTDOWN     ;SHUTDOWN DMC
013312  012700  013340                          MOV     #25$,R0         ;POINTER TO EXPECTED SOFT COUNTS
013316  012701  021443                  21$:    MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
013322  012702  000010                          MOV     #10,R2          ;COUNT
013326  122021                          22$:    CMPB    (R0)+,(R1)+     ;COMPARE SOFT ERROR COUNTS
013330  001007                                  BNE     23$             ;IF ERROR BR 23$
013332  005302                                  DEC     R2              ;DEC COUNT
013334  001374                                  BNE     22$             ;CONTINUE CHECKING IF NOT DONE
013336  000421                                  BR      24$             ;ALL COUNTS OK, GET OUT
013340     000     000     000          25$:    .BYTE   0,0,0,0,0,0,0,0 ;EXPECTED ERROR COUNTS
013343     000     000     000
013346     000     000
013350  113737  021443  001250          23$:    MOVB    BASE+3,TEMP2
013356  113737  021445  001252                  MOVB    BASE+5,TEMP3
013364  113737  021447  001254                  MOVB    BASE+7,TEMP4
013372  113737  021451  001256                  MOVB    BASE+11,TEMP5
013400  104017                                  HLT     17
013402                                  24$:
013402  104400                          14$:    SCOPE                   ;SCOPE THIS TEST



                                                ;************************** TEST 2 **************************
                                                ;*OVERUN TEST
                                                ;*IN FREE RUNNING MODE SEND MESSAGE WITH NO RECEIVE
                                                ;*BUFFER AVAILABLE, VERIFY THAT AN OVERRUN ERROR OCCURS
                                                ;!***********************************************************


                                                ;  TEST 2
                                                ;---------------
013404  012737  000002  001226          TST2:   MOV     #2,TSTNO
013412  012737  013754  001216                  MOV     #TST3,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
013420  032737  100000  001366                  BIT     #BIT15,STAT1    ;IS IT A DMC?
013426  001406                                  BEQ     .+16            ;BR IF YES
013430  032737  000001  001372                  BIT     #BIT0,STAT3     ;KMC WITH BIT0 SET?
013436  001002                                  BNE     .+6             ;BR IF YES
013442  000137  013736                          JMP     10$             ;SKIP TEST
013444  032737  010000  001366                  BIT     #BIT12,STAT1    ;LU PRESENT?
013452  001372                                  BNE     .-12            ;BR IF NO
013454  004737  022040                          JSR     PC,BASELD                 ;LOAD DMC BASE ADDRESS
013460  004537  022450                          JSR     R5,XFRELD       ;LOAD XMIT BA/CC
013464  021324                                  TBUF                    ;BA
013466  000044                                  44                      ;CC
013470  012700  000010                          MOV     #10,R0          ;R0 = RETRANSMISSION COUNT
013474  012703  000015                          MOV     #15,R3          ;DELAY COUNT
013500  005037  001416                          CLR     TEMP            ;CLEAR DELAY COUNTER
013504  105761  000002                  1$:     TSTB    2(R1)           ;IS RDY O SET?
013510  100407                                  BMI     .+20            ;BR IF SET
013512  005237  001416                          INC     TEMP            ;INC DELAY COUNTER
013516  001372                                  BNE     1$              ;BR IF NOT DONE DELAY
013520  005303                                  DEC     R3              ;DEC DELAY COUNT
013522  001370                                  BNE     1$              ;BR IF DELAY NOT DONE
```

```
013524  104014                                  HLT     14              ;ERROR, RDY O NOT SET
013526  000503                                  BR      10$             ;GET OUT
013530  132761  000001  000002                  BITB    #BIT0,2(R1)     ;IS IT CNTL O?
013536  001002                                  BNE     11$             ;BR IF YES
013540  104014                                  HLT     14              ;ERROR, NOT CNTL O
013542  000475                                  BR      10$             ;CONTINUE
013544  012705  000004                  11$:    MOV     #BIT2,R5        ;PUT "EXPECTED" IN R5
013550  016104  000006                          MOV     6(R1),R4        ;PUT "FOUND" IN R4
013554  020504                                  CMP     R5,R4           ;IS ORUN SET?
013556  001404                                  BEQ     12$             ;BR IF YES
013560  022704  000001                          CMP     #1,R4           ;DATA CK ERROR?
013564  001465                                  BEQ     13$             ;BR IF YES
013566  104015                                  HLT     15              ;ERROR, ORUN NOT SET
013570  042761  000207  000002          12$:    BIC     #207,2(R1)      ;CLEAR RDO
013576  005037  001416                          CLR     TEMP            ;RESET DELAY
013602  005300                                  DEC     R0              ;DEC RETRANS COUNT
013604  001337                                  BNE     1$              ;COUNTINUE
013606  004737  022502                          JSR     PC,SHUTDOWN     ;SHUTDOWN DMC
013612  032737  020000  001366                  BIT     #BIT13,STAT1    ;IS IT AN M8201?
013620  001446                                  BEQ     10$             ;SKIP BASE CHECK IF YES
013622  012700  013664                          MOV     #25$,R0         ;POINTER TO EXPECTED SOFT COUNTS (LOW SPEED)
013626  032737  000002  001372                  BIT     #BIT1,STAT3     ;IS IT HIGH OR LOW
013634  001402                                  BEQ     21$             ;BR IF LOW
013636  012700  013674                          MOV     #26$,R0         ;POINTER TO EXPECTED SOFT COUNTS (HIGH SPEED)
013642  012701  021443                  21$:    MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
013646  012702  000010                          MOV     #10,R2          ;COUNT
013652  122021                          22$:    CMPB    (R0)+,(R1)+     ;COMPARE SOFT ERROR COUNTS
013654  001013                                  BNE     23$             ;IF ERROR BR 23$
013656  005302                                  DEC     R2              ;DEC COUNT
013660  001374                                  BNE     22$             ;CONTINUE CHECKING IF NOT DONE
013662  000425                                  BR      24$             ;ALL COUNTS OK, GET OUT
013664     000     000     000          25$:    .BYTE   0,0,0,100,0,0,0,0       ;EXPECTED ERROR COUNTS (LOW SPEED)
013667     100     000     000
013672     000     000

013674     000     000     077          26$:    .BYTE   0,0,77,100,0,0,0,0      ;EXPECTED ERROR COUNTS (HIGH SPEED)
013677     100     000     000
013702     000     000
013704  113737  021443  001250          23$:    MOVB    BASE+3,TEMP2
013712  113737  021445  001252                  MOVB    BASE+5,TEMP3
013720  113737  021447  001254                  MOVB    BASE+7,TEMP4
013726  113737  021451  001256                  MOVB    BASE+11,TEMP5
013734  104017                                  HLT     17
013736                                  24$:
013736  104400                          10$:    SCOPE                   ;SCOPE THIS TEST
013740  042761  000207  000002          13$:    BIC     #207,2(R1)      ;IGNOR THIS ERROR
013746  005037  001416                          CLR     TEMP            ;RESET DELAY
013752  000654                                  BR      1$              ;CONTINUE



                                                ;************************** TEST 3 **************************
                                                ;*LOST DATA TEST
                                                ;*IN FREE RUNNING MODE SEND A MESSAGE LONGER THAN THE RECEIVE
                                                ;*BUFFER, VERIFY THAT A LOST DATA ERROR OCCURS.
                                                ;!***********************************************************


                                                ;  TEST 3
```

```
1944                                                   ;---------------
1945  013754  012737  000003  001226         TST3:    MOV     #3,TSTNO
1946  013762  012737  014236  001216                  MOV     #TST4,NEXT
1947                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
1948  013770  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
1949  013772  032737  100000  001366                  BIT     #BIT15,STAT1    ;IS IT A DMC?
1950  014000  001406                                  BEQ     .+16            ;BR IF YES
1951  014002  032737  000001  001372                  BIT     #BIT0,STAT3     ;KMC WITH BIT0 SET?
1952  014010  001002                                  BNE     .+6             ;BR IF YES
1953  014012  000137  014234                          JMP     10$             ;SKIP TEST
1954  014016  032737  010000  001366                  BIT     #BIT12,STAT1    ;LU PRESENT?
1955  014024  001372                                  BNE     .-12            ;BR IF NO
1956  014026  004737  022040                          JSR     PC,BASELD               ;LOAD DMC BASE ADDRESS
1957  014032  004537  022416                          JSR     R5,RFRELD       ;LOAD RECEIVE BA/CC
1958  014036  021372                                  RBUF                    ;BA
1959  014040  000020                                  20                      ;CC
1960  014042  004537  022450                          JSR     R5,XFRELD       ;LOAD XMIT BA/CC
1961  014046  021324                                  TBUF                    ;BA
1962  014050  000044                                  44                      ;CC
1963  014052  012703  000015                          MOV     #15,R3          ;DELAY COUNT
1964  014056  005037  001416                          CLR     TEMP            ;CLEAR DELAY COUNTER
1965  014062  105761  000002         1$:      TSTB    2(R1)           ;IS RDY O SET?
1966  014066  100407                                  BMI     .+20            ;BR IF SET
1967  014070  005237  001416                          INC     TEMP            ;INC DELAY COUNTER
1968  014074  001372                                  BNE     1$              ;BR IF NOT DONE DELAY
1969  014076  005303                                  DEC     R3              ;DEC DELAY COUNT
1970  014100  001370                                  BNE     1$              ;BR IF DELAY NOT DONE
1971  014102  104014                                  HLT     14              ;ERROR, RDY O NOT SET
1972  014104  000453                                  BR      10$             ;GET OUT
1973  014106  132761  000001  000002                  BITB    #BIT0,2(R1)     ;IS IT CNTL O?
1974  014114  001002                                  BNE     11$             ;BR IF YES
1975  014116  104014                                  HLT     14              ;ERROR NOT CNTL O
1976  014120  000445                                  BR      10$             ;CONTINUE
1977  014122  012705  000020         11$:     MOV     #BIT4,R5        ;PUT "EXPECTED" IN R5
1978  014126  016104  000006                          MOV     6(R1),R4        ;PUT "FOUND" IN R4
1979  014132  020504                                  CMP     R5,R4           ;IS LOST DATA SET?
1980  014134  001401                                  BEQ     12$             ;BR IF YES
1981  014136  104015                                  HLT     15              ;ERROR, LOST DATA NOT SET
1982  014140  004737  022502         12$:     JSR     PC,SHUTDOWN     ;SHUTDOWN DMC
1983  014144  012700  014172                          MOV     #25$,R0         ;POINTER TO EXPECTED SOFT COUNTS
1984  014150  012701  021443         21$:     MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
1985  014154  012702  000010                          MOV     #10,R2          ;COUNT
1986  014160  122021                         22$:     CMPB    (R0)+,(R1)+     ;COMPARE SOFT ERROR COUNTS
1987  014162  001007                                  BNE     23$             ;IF ERROR BR 23$
1988  014164  005302                                  DEC     R2              ;DEC COUNT
1989  014166  001374                                  BNE     22$             ;CONTINUE CHECKING IF NOT DONE
1990  014170  000421                                  BR      24$             ;ALL COUNTS OK, GET OUT
1991  014172     000     000     000         25$:     .BYTE   0,0,0,0,0,0,0,0 ;EXPECTED ERROR COUNTS
1992  014175     000     000     000
1993  014200     000     000
1994  014202  113737  021443  001250         23$:     MOVB    BASE+3,TEMP2
1995  014210  113737  021445  001252                  MOVB    BASE+5,TEMP3
1996  014216  113737  021447  001254                  MOVB    BASE+7,TEMP4
1997  014224  113737  021451  001256                  MOVB    BASE+11,TEMP5
1998  014232  104017                                  HLT     17
1999  014234                                 24$:
```

```
2000  014234  104400                         10$:     SCOPE                   ;SCOPE THIS TEST
2001
2002
2003                                                   ;*************************** TEST 4 ***************************
2004                                                   ;*TRANSMIT NON-EXISTENT MEMORY TEST
2005                                                   ;*IN FREE RUNNING MODE, LOAD A TRANSMIT BA THAT WILL TIME OUT
2006                                                   ;*VERIFY THAT A NON-EXISTENT MEMORY ERROR OCCURS
2007                                                   ;;*************************************************************
2008
2009                                                   ;  TEST 4
2010                                                   ;---------------
2011  014236  012737  000004  001226         TST4:    MOV     #4,TSTNO
2012  014244  012737  014510  001216                  MOV     #TST5,NEXT
2013                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
2014  014252  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
2015  014254  032737  100000  001366                  BIT     #BIT15,STAT1    ;IS IT A DMC?
2016  014262  001406                                  BEQ     .+16            ;BR IF YES
2017  014264  032737  000001  001372                  BIT     #BIT0,STAT3     ;KMC WITH BIT0 SET?
2018  014272  001002                                  BNE     .+6             ;BR IF YES
2019  014274  000137  014506                          JMP     10$             ;SKIP TEST
2020  014300  032737  010000  001366                  BIT     #BIT12,STAT1    ;LU PRESENT?
2021  014306  001372                                  BNE     .-12            ;BR IF NO
2022  014310  004737  022040                          JSR     PC,BASELD               ;LOAD DMC BASE ADDRESS
2023  014314  004537  022450                          JSR     R5,XFRELD       ;LOAD XMIT BA/CC
2024  014320  177320                                  177320                  ;BA
2025  014322  140044                                  140044                  ;CC
2026  014324  012703  000015                          MOV     #15,R3          ;DELAY COUNT
2027  014330  005037  001416                          CLR     TEMP            ;CLEAR DELAY COUNTER
2028  014334  105761  000002         1$:      TSTB    2(R1)           ;IS RDY O SET?
2029  014340  100407                                  BMI     .+20            ;BR IF SET
2030  014342  005237  001416                          INC     TEMP            ;INC DELAY COUNTER
2031  014346  001372                                  BNE     1$              ;BR IF NOT DONE DELAY
2032  014350  005303                                  DEC     R3              ;DEC DELAY COUNT
2033  014352  001370                                  BNE     1$              ;BR IF DELAY NOT DONE
2034  014354  104014                                  HLT     14              ;ERROR, RDY O NOT SET
2035  014356  000453                                  BR      10$             ;GET OUT
2036  014360  132761  000001  000002                  BITB    #BIT0,2(R1)     ;IS IT CNTL O?
2037  014366  001002                                  BNE     11$             ;BR IF YES
2038  014370  104014                                  HLT     14              ;ERROR, NOT CNTL O
2039  014372  000445                                  BR      10$             ;CONTINUE
2040  014374  012705  000400         11$:     MOV     #BIT8,R5        ;PUT "EXPECTED" IN R5
2041  014400  016104  000006                          MOV     6(R1),R4        ;PUT "FOUND" IN R4
2042  014404  020504                                  CMP     R5,R4           ;IS NON-EX-MEM SET?
2043  014406  001401                                  BEQ     .+4             ;BR IF YES
2044  014410  104015                                  HLT     15              ;ERROR NON-EX-MEM NOT SET
2045  014412  004737  022502                          JSR     PC,SHUTDOWN     ;SHUTDOWN DMC
2046  014416  012700  014444                          MOV     #25$,R0         ;POINTER TO EXPECTED SOFT COUNTS
2047  014422  012701  021443         21$:     MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
2048  014426  012702  000010                          MOV     #10,R2          ;COUNT
2049  014432  122021                         22$:     CMPB    (R0)+,(R1)+     ;COMPARE SOFT ERROR COUNTS
2050  014434  001007                                  BNE     23$             ;IF ERROR BR 23$
2051  014436  005302                                  DEC     R2              ;DEC COUNT
2052  014440  001374                                  BNE     22$             ;CONTINUE CHECKING IF NOT DONE
2053  014442  000421                                  BR      24$             ;ALL COUNTS OK, GET OUT
2054  014444     000     000     000         25$:     .BYTE   0,0,0,0,0,0,0,0 ;EXPECTED ERROR COUNTS
2055  014447     000     000     000
```

```
014452     000     000
014454  113737  021443  001250        23$:    MOVB    BASE+3,TEMP2
014462  113737  021445  001252                MOVB    BASE+5,TEMP3
014470  113737  021447  001254                MOVB    BASE+7,TEMP4
014476  113737  021451  001256                MOVB    BASE+11,TEMP5
014504  104017                                HLT     17
014506                                24$:
014506  104400                        10$:    SCOPE                   ;SCOPE THIS TEST


                                              ;************************** TEST 5 **************************
                                              ;*RECEIVE NON-EXISTENT MEMORY TEST
                                              ;*IN FREE RUNNING MODE, LOAD A RECEIVE BA THAT WILL TIME OUT
                                              ;*VERIFY THAT A NON-EXISTENT MEMORY ERROR OCCURS
                                              ;!***********************************************************

                                              ;  TEST 5
                                              ;--------------
014510  012737  000005  001226        TST5:   MOV     #5,TSTNO
014516  012737  014772  001216                MOV     #TST6,NEXT
                                                                      ;R1 CONTAINS BASE DMC11 ADDRESS
014524  104412                                MSTCLR                  ;MASTER CLEAR DMC11
014526  032737  100000  001366                BIT     #BIT15,STAT1    ;IS IT A DMC?
014534  001406                                BEQ     .+16            ;BR IF YES
014536  032737  000001  001372                BIT     #BIT0,STAT3     ;KMC WITH BIT0 SET?
014544  001002                                BNE     .+6             ;BR IF YES
014546  000137  014770                        JMP     10$             ;SKIP TEST
014552  032737  010000  001366                BIT     #BIT12,STAT1    ;LU PRESENT?
014560  001372                                BNE     .-12            ;BR IF NO
014562  004737  022040                        JSR     PC,BASELD               ;LOAD DMC BASE ADDRESS
014566  004537  022416                        JSR     R5,RFRELD       ;LOAD RECEIVE BA/CC
014572  177320                                177320                  ;BA
014574  140044                                140044                  ;CC
014576  004537  022450                        JSR     R5,XFRELD       ;LOAD XMIT BA/CC
014602  021324                                TBUF                    ;BA
014604  000044                                44                      ;CC
014606  012703  000015                        MOV     #15,R3          ;DELAY COUNT
014612  005037  001416                        CLR     TEMP            ;CLEAR DELAY COUNTER
014616  105761  000002                1$:     TSTB    2(R1)           ;IS RDY O SET?
014622  100407                                BMI     .+20            ;BR IF SET
014624  005237  001416                        INC     TEMP            ;INC DELAY COUNTER
014630  001372                                BNE     1$              ;BR IF NOT DONE DELAY
014632  005303                                DEC     R3              ;DEC DELAY COUNT
014634  001370                                BNE     1$              ;BR IF DELAY NOT DONE
014636  104014                                HLT     14              ;ERROR, RDY O NOT SET
014640  000453                                BR      10$             ;GET OUT
014642  132761  000001  000002                BITB    #BIT0,2(R1)     ;IS IT CNTL O?
014650  001002                                BNE     11$             ;BR IF YES
014652  104014                                HLT     14              ;ERROR, NOT CNTL O
014654  000445                                BR      10$             ;CONTINUE
014656  012705  000400                11$:    MOV     #BIT8,R5        ;PUT "EXPECTED" IN R5
014662  016104  000006                        MOV     6(R1),R4        ;PUT "FOUND" IN R4
014666  020504                                CMP     R5,R4           ;IS NON-EX-MEM SET?
014670  001401                                BEQ     .+4             ;BR IF YES
014672  104015                                HLT     15              ;ERROR NON-EX-MEM NOT SET
014674  004737  022502                        JSR     PC,SHUTDOWN     ;SHUTDOWN DMC
```

```
014700  012700  014726                        MOV     #25$,R0         ;POINTER TO EXPECTED SOFT COUNTS
014704  012701  021443                21$:    MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
014710  012702  000010                        MOV     #10,R2          ;COUNT
014714  122021                        22$:    CMPB    (R0)+,(R1)+     ;COMPARE SOFT ERROR COUNTS
014716  001007                                BNE     23$             ;IF ERROR BR 23$
014720  005302                                DEC     R2              ;DEC COUNT
014722  001374                                BNE     22$             ;CONTINUE CHECKING IF NOT DONE
014724  000421                                BR      24$             ;ALL COUNTS OK, GET OUT
014726     000     000     000        25$:    .BYTE   0,0,0,0,0,0,0,0 ;EXPECTED ERROR COUNTS
014731     000     000     000
014734     000     000
014736  113737  021443  001250        23$:    MOVB    BASE+3,TEMP2
014744  113737  021445  001252                MOVB    BASE+5,TEMP3
014752  113737  021447  001254                MOVB    BASE+7,TEMP4
014760  113737  021451  001256                MOVB    BASE+11,TEMP5
014766  104017                                HLT     17
014770                                24$:
014770  104400                        10$:    SCOPE                   ;SCOPE THIS TEST


                                              ;************************** TEST 6 **************************
                                              ;*PROCESSOR ERROR TEST
                                              ;*IN FREE RUNNING MODE, DO A BASE TRANSFER REQUEST AFTER A
                                              ;*BASE HAS BEEN SET UP, VERIFY THAT A PROCESSOR ERROR OCCURS,
                                              ;!***********************************************************

                                              ;  TEST 6
                                              ;--------------
014772  012737  000006  001226        TST6:   MOV     #6,TSTNO
015000  012737  015234  001216                MOV     #TST7,NEXT
                                                                      ;R1 CONTAINS BASE DMC11 ADDRESS
015006  104412                                MSTCLR                  ;MASTER CLEAR DMC11
015010  032737  100000  001366                BIT     #BIT15,STAT1    ;IS IT A DMC?
015016  001406                                BEQ     .+16            ;BR IF YES
015020  032737  000001  001372                BIT     #BIT0,STAT3     ;KMC WITH BIT0 SET?
015026  001002                                BNE     .+6             ;BR IF YES
015030  000137  015232                        JMP     10$             ;SKIP TEST
015034  032737  010000  001366                BIT     #BIT12,STAT1    ;LU PRESENT?
015042  001372                                BNE     .-12            ;BR IF NO
015044  004737  022040                        JSR     PC,BASELD       ;LOAD BASE ADDRESS
015050  152711  000043                12$:    BISB    #43,(R1)        ;2ND BASE REQUEST
015054  105711                                TSTB    (R1)            ;RDI SET?
015056  100376                                BPL     .-2             ;BR IF NO
015060  142711  000040                        BICB    #40,(R1)        ;CLEAR RQI
015064  005037  001416                        CLR     TEMP            ;GET SET TO DELAY
015070  105761  000002                13$:    TSTB    2(R1)           ;RDO SET?
015074  100405                                BMI     14$             ;BR IF YES
015076  005237  001416                        INC     TEMP            ;INC DELAY
015102  001372                                BNE     13$             ;BR IF NOT DONE DELAY
015104  104014                                HLT     14              ;ERROR, RDO NOT SET
015106  000770                                BR      13$             ;TRY AGAIN
015110  132761  000001  000002        14$:    BITB    #BIT0,2(R1)     ;IS IS CNTL O?
015116  001002                                BNE     11$             ;BR IF YES
015120  104014                                HLT     14              ;ERROR NOT CNTL O
015122  000443                                BR      10$             ;CONTINUE
015124  012705  001000                11$:    MOV     #BIT9,R5        ;PUT "EXPECTED" IN R5
```

```
 2168  015130  016104  000006                     MOV     6(R1),R4           ;PUT "FOUND" IN R4
 2169  015134  020504                             CMP     R5,R4              ;IS PROC ERROR SET?
 2170  015136  001401                             BEQ     .+4                ;BR IF YES
 2171  015140  104015                             HLT     15                 ;ERROR, PROC ERROR NOT SET
 2172  015142  012700  015170                     MOV     #25$,R0            ;POINTER TO EXPECTED SOFT COUNTS
 2173  015146  012701  021443             21$:    MOV     #BASE+3,R1         ;POINTER TO ACTUAL COUNTS
 2174  015152  012702  000010                     MOV     #10,R2             ;COUNT
 2175  015156  122021                     22$:    CMPR    (R0)+,(P1)+        ;COMPARE SOFT ERROR COUNTS
 2176  015160  001007                             BNE     23$                ;IF ERROR BR 23$
 2177  015162  005302                             DEC     R2                 ;DEC COUNT
 2178  015164  001374                             BNE     22$                ;CONTINUE CHECKING IF NOT DONE
 2179  015166  000421                             BR      24$                ;ALL COUNTS OK, GET OUT
 2180  015170     000     000     000     25$:    .BYTE   0,0,0,0,0,0,0,0 ;EXPECTED ERROR COUNTS
 2181  015173     000     000     000
 2182  015176     000     000
 2183  015200  113737  021443  001250     23$:    MOVB    BASE+3,TEMP2
 2184  015206  113737  021445  001252             MOVB    BASE+5,TEMP3
 2185  015214  113737  021447  001254             MOVB    BASE+7,TEMP4
 2186  015222  113737  021451  001256             MOVB    BASE+11,TEMP5
 2187  015230  104017                             HLT     17
 2188  015232                             24$:
 2189  015232  104400                     10$:    SCOPE                      ;SCOPE THIS TEST
 2190
 2191
 2192                                             ;*************************** TEST 7 ***************************
 2193                                             ;*PROCESSOR ERROR TEST
 2194                                             ;*IN FREE RUNNING MODE DO A RQI WITH AN ILLEGAL 10 CODE
 2195                                             ;*VERIFY THAT A PROCESSOR ERROR OCCURS
 2196                                             ;**************************************************************
 2197
 2198                                             ;  TEST 7
 2199                                             ;--------------
 2200  015234  012737  000007  001226     TST7:   MOV     #7,TSTNO
 2201  015242  012737  015476  001216             MOV     #TST10,NEXT
 2202                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
 2203  015250  104412                             MSTCLR                     ;MASTER CLEAR DMC11
 2204  015252  032737  100000  001366             BIT     #BIT15,STAT1       ;IS IT A DMC?
 2205  015260  001406                             BEQ     .+16               ;BR IF YES
 2206  015262  032737  000001  001372             BIT     #BIT0,STAT3        ;KMC WITH BIT0 SET?
 2207  015270  001002                             BNE     .+6                ;BR IF YES
 2208  015272  000137  015474                     JMP     10$                ;SKIP TEST
 2209  015276  032737  010000  001366             BIT     #BIT12,STAT1       ;LU PRESENT?
 2210  015304  001372                             BNE     .-12               ;BR IF NO
 2211  015306  004737  022040                     JSR     PC,BASELD                  ;LOAD DMC BASE ADDRESS
 2212  015312  152711  000046                     BISB    #46,(R1)           ;RQI AND ILLEGAL CODE
 2213  015316  105711                             TSTB    (R1)               ;WAIT FOR RDI
 2214  015320  100376                             BPL     .-2                ;BR IF NO RDI
 2215  015322  142711  000040                     BICB    #40,(R1)           ;CLEAR RQI
 2216  015326  005037  001416                     CLR     TEMP               ;CLEAR COUNTER
 2217  015332  105761  000002             1$:     TSTB    2(R1)              ;RDY O SET?
 2218  015336  100405                             BMI     .+14               ;BR IF YES
 2219  015340  005237  001416                     INC     TEMP               ;BUMP COUNTER DELAY
 2220  015344  001372                             BNE     1$                 ;BR IF NOT DONE
 2221  015346  104014                             HLT     14                 ;ERROR NO RDY O
 2222  015350  000770                             BR      1$                 ;TRY AGAIN
 2223  015352  132761  000001  000002             BITB    #BIT0,2(R1)        ;IS IT CNTL O
```

```
 2224  015360  001002                             BNE     11$                ;BR IF YES
 2225  015362  104014                             HLT     14                 ;ERROR, NOT CNTL O
 2226  015364  000443                             BR      10$                ;CONTINUE
 2227  015366  012705  001000             11$:    MOV     #BIT9,R5           ;PUT "EXPECTED" IN R5
 2228  015372  016104  000006                     MOV     6(R1),R4           ;PUT "FOUND" IN R4
 2229  015376  020504                             CMP     R5,R4              ;IS PROC ERROR SET?
 2230  015400  001401                             BEQ     .+4                ;BR IF YES
 2231  015402  104015                             HLT     15                 ;ERROR PROC ERROR NOT SET
 2232  015404  012700  015432                     MOV     #25$,R0            ;POINTER TO EXPECTED SOFT COUNTS
 2233  015410  012701  021443             21$:    MOV     #BASE+3,R1         ;POINTER TO ACTUAL COUNTS
 2234  015414  012702  000010                     MOV     #10,R2             ;COUNT
 2235  015420  122021                     22$:    CMPB    (R0)+,(R1)+        ;COMPARE SOFT ERROR COUNTS
 2236  015422  001007                             BNE     23$                ;IF ERROR BR 23$
 2237  015424  005302                             DEC     R2                 ;DEC COUNT
 2238  015426  001374                             BNE     22$                ;CONTINUE CHECKING IF NOT DONE
 2239  015430  000421                             BR      24$                ;ALL COUNTS OK, GET OUT
 2240  015432     000     000     000     25$:    .BYTE   0,0,0,0,0,0,0,0 ;EXPECTED ERROR COUNTS
 2241  015435     000     000     000
 2242  015440     000     000
 2243  015442  113737  021443  001250     23$:    MOVB    BASE+3,TEMP2
 2244  015450  113737  021445  001252             MOVB    BASE+5,TEMP3
 2245  015456  113737  021447  001254             MOVB    BASE+7,TEMP4
 2246  015464  113737  021451  001256             MOVB    BASE+11,TEMP5
 2247  015472  104017                             HLT     17
 2248  015474                             24$:
 2249  015474  104400                     10$:    SCOPE                      ;SCOPE THIS TEST
 2250
 2251
 2252                                             ;*************************** TEST 10 ***************************
 2253                                             ;*HALF DUPLEX TEST
 2254                                             ;*IN FREE RUNNING MODE, SET HALF DUPLEX AND L U LOOP
 2255                                             ;*SEND A MESSAGE AND VERIFY THAT THERE ARE NO DONES
 2256                                             ;***************************************************************
 2257
 2258                                             ;  TEST 10
 2259                                             ;--------------
 2260  015476  012737  000010  001226     TST10:  MOV     #10,TSTNO
 2261  015504  012737  015632  001216             MOV     #TST11,NEXT
 2262                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
 2263  015512  104412                             MSTCLR                     ;MASTER CLEAR DMC11
 2264  015514  032737  100000  001366             BIT     #BIT15.STAT1       ;IS IT A DMC?
 2265  015522  001406                             BEQ     .+16               ;BR IF YES
 2266  015524  032737  000001  001372             BIT     #BIT0.STAT3        ;KMC WITH BIT0 SET?
 2267  015532  001002                             BNE     .+6                ;BR IF YES
 2268  015534  000137  015624                     JMP     10$                ;SKIP TEST
 2269  015540  032737  010000  001366             BIT     #BIT12,STAT1       ;LU PRESENT?
 2270  015546  001372                             BNE     .-12               ;BR IF NO
 2271  015550  004737  022156                     JSR     PC,BASELH          ;LOAD BASE AND HALF DUPLEX
 2272  015554  004537  022416                     JSR     R5,RFRELD          ;LOAD RECEIVE BUFFER
 2273  015560  021372                             RBUF                       ;BA
 2274  015562  000044                             44                         ;CC
 2275  015564  004537  022450                     JSR     R5,XFRELD          ;LOAD TRANSMIT BUFFER
 2276  015570  021324                             TBUF                       ;BA
 2277  015572  000044                             44                         ;CC
 2278  015574  012703  000003                     MOV     #3,R3              ;LOAD DELAY COUNT
 2279  015600  005037  001416                     CLR     TEMP               ;CLEAR DELAY
```

```
015604  105761  000002                  4$:     TSTB    2(R1)           ;IS DONE SET?
015610  100406                                  BMI     5$              ;BR IF YES (ERROR)
015612  005237  001416                          INC     TEMP            ;INC DELAY
015616  001372                                  BNE     4$              ;BR IF DELAY NOT DONE
015620  005303                                  DEC     R3              ;DEC DELAY COUNT
015622  001370                                  BNE     4$              ;BR IF DELAY NOT DONE
015624  104400                          10$:    SCOPE                   ;SCOPE THIS TEST
015626  104014                          5$:     HLT     14              ;ERROR DONE WITH HALF-DUPLEX
015630  000775                                  BR      10$             ;GET OUT


                                                ;*************************** TEST 11 ***************************
                                                ;*RESUME TEST
                                                ;*THIS TEST SENDS AND RECEIVES A BUFFER AND SHUTS DOWN THE
                                                ;*DMC. THEN A MASTER CLEAR IS ISSUED AND A BASE WITH RESUME
                                                ;*BIT SET IS GIVEN, ANOTHER BUFFER IS SENT AND RECEIVED.
                                                ;*DATA IS CHECKED.
                                                ;!*************************************************************

                                                ;  TEST 11
                                                ;--------------
015632  012737  000011  001226          TST11:  MOV     #11,TSTNO
015640  012737  016214  001216                  MOV     #TST12,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
015646  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
015650  032737  100000  001366                  BIT     #BIT15,STAT1    ;IS IT A DMC?
015656  001436                                  BEQ     .+16            ;BR IF YES
015660  032737  000001  001372                  BIT     #BIT0,STAT3     ;KMC WITH BIT0 SET?
015666  001002                                  BNE     .+6             ;BR IF YES
015670  000137  016212                          JMP     10$             ;SKIP TEST
015674  032737  010000  001366                  BIT     #BIT12,STAT1    ;LU PRESENT?
015702  001372                                  BNE     .-12            ;BR IF NO
015704  005037  020060                          CLR     RESUME          ;CLR RESUME FLAG
015710  005737  020060                  1$:     TST     RESUME          ;FIRST OR SECOND PASS?
015714  001003                                  BNE     2$              ;BR IF SECOND
015716  004737  022040                          JSR     PC,BASELD       ;BASE
015722  000402                                  BR      3$              ;CONTINUE
015724  004737  022276                  2$:     JSR     PC,RESUM        ;BASE WITH RESUME BIT
015730  004537  022416                  3$:     JSR     R5,RFRELD       ;RECEIVE BUFFER
015734  021372                                  RBUF                    ;BA
015736  000044                                  44                      ;CC
015740  004537  022450                          JSR     R5,XFRELD       ;XMIT BUFFER
015744  021324                                  TBUF                    ;BA
015746  000044                                  44                      ;CC
015750  012703  000030                          MOV     #30,R3          ;DELAY COUNT
015754  012700  000002                          MOV     #2,R0           ;NEED TWO DONES
015760  005037  001416                          CLR     TEMP            ;CLEAR DELAY COUNTER
015764  105761  000002                  4$:     TSTB    2(R1)           ;IS RDY O SET?
015770  100407                                  BMI     .+20            ;BR IF SET
015772  005237  001416                          INC     TEMP            ;INC DELAY COUNTER
015776  001372                                  BNE     4$              ;BR IF NOT DONE DELAY
016000  005303                                  DEC     R3              ;DEC DELAY COUNT
016002  001370                                  BNE     4$              ;BR IF DELAY NOT DONE
016004  104014                                  HLT     14              ;ERROR, RDY O NOT SET
016006  000531                                  BR      10$             ;GET OUT
016010  042761  000207  000002                  BIC     #207,2(R1)      ;CLEAR DONE
```

```
016016  005300                                  DEC     R0              ;TWO DONES YET?
016020  001361                                  BNE     4$              ;BR IF NOT
016022  012702  021324                          MOV     #TBUF,R2        ;ADDRESS OF GOOD DATA
016026  012703  021372                          MOV     #RBUF,R3        ;ADDRESS OF RECEIVED DATA
016032  012700  000044                          MOV     #44,R0          ;COUNT
016036  112205                          6$:     MOVB    (R2)+,R5        ;LOAD GOOD DATA
016040  112304                                  MOVB    (R3)+,R4        ;LOAD FOUND DATA
016042  120504                                  CMPB    R5,R4           ;COMPARE DATA
016044  001401                                  BEQ     7$              ;BR IF OK
016046  104012                                  HLT     12              ;DATA ERROR
016050  005300                          7$:     DEC     R0              ;DONE YET?
016052  001371                                  BNE     6$              ;BR IF NOT
016054  004737  022502                          JSR     PC,SHUTDOWN     ;SHUTDOWN DMC
016060  005737  020060                          TST     RESUME          ;
016064  001004                                  BNE     8$              ;BR IF ALL DONE
016066  012737  177777  020060                  MOV     #-1,RESUME      ;SET FLAG FOR SECOND PASS
016074  000705                                  BR      1$              ;CONTINUE
016076                                  8$:
016076  012700  016140                          MOV     #25$,R0         ;POINTER TO EXPECTED SOFT COUNTS (LOW SPEED)
016102  032737  000002  001372                  BIT     #BIT1,STAT3     ;IS IT HIGH OR LOW
016110  001402                                  BEQ     21$             ;BR IF LOW
016112  012700  016150                          MOV     #26$,R0         ;POINTER TO EXPECTED SOFT COUNTS (HIGH SPEED)
016116  012701  021443                  21$:    MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
016122  012702  000010                          MOV     #10,R2          ;COUNT
016126  122021                          22$:    CMPB    (R0)+,(R1)+     ;COMPARE SOFT ERROR COUNTS
016130  001013                                  BNE     23$             ;IF ERROR BR 23$
016132  005302                                  DEC     R2              ;DEC COUNT
016134  001374                                  BNE     22$             ;CONTINUE CHECKING IF NOT DONE
016136  000425                                  BR      24$             ;ALL COUNTS OK, GET OUT
016140     000     000     000          25$:    .BYTE   0,0,0,0,0,0,1,1 ;EXPECTED ERROR COUNTS (LOW SPEED)
016143     000     000     000
016146     001     001
016150     000     000     000          26$:    .BYTE   0,0,0,0,0,0,0,0 ;EXPECTED ERROR COUNTS (HIGH SPEED)
016153     000     000     000
016156     000     000
016160  113737  021443  001250          23$:    MOVB    BASE+3,TEMP2
~~016166  113737  021445  001252                  MOVB    BASE+5,TEMP3~~
016174  113737  021447  001254                  MOVB    BASE+7,TEMP4
016202  113737  021451  001256                  MOVB    BASE+11,TEMP5
016210  104017                                  HLT     17
016212                                  24$:
016212  104400                          10$:    SCOPE                   ;SCOPE THIS TEST


                                                ;*************************** TEST 12 ***************************
                                                ;*FREE RUNNING DATA TEST (INTERRUPT DRIVEN EXERCISER)
                                                ;*THIS TEST REPEATEDLY QUEUES UP 7 RECEIVE BUFFERS AND
                                                ;*7 TRANSMIT BUFFERS AND CHECKS DATA WHEN ALL 7 BUFFERS
                                                ;*ARE RECEIVED. TRANSMIT COUNTS RANGE FROM 2 TO 104.
                                                ;*DATA IS A BINARY COUNT PATTERN. THE RESUME FUNCTION
                                                ;*IS CHECKED IN THIS TEST. THIS TEST USES THE TURNAROUND CONNECTOR
                                                ;*IF IT IS PRESENT, OTHERWISE LINE UNIT LOOP IS SET.
                                                ;!*************************************************************

                                                ;  TEST 12
                                                ;--------------
```

```
016214  012737  000012  001226          TST12:  MOV     #12,TSTNO
016222  012737  003364  001216                  MOV     #.EOP,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
016230  104412                                  MSTCLR          ;MASTER CLEAR DMC11
016232  032737  100000  001366                  BIT     #BIT15,STAT1    ;IS IT A DMC?
016240  001406                                  BEQ     .+16            ;BR IF YES
016242  032737  000001  001372                  BIT     #BIT0,STAT3     ;KMC WITH BIT0 SET?
016250  001002                                  BNE     .+6             ;BR IF YES
016252  000137  017044                          JMP     ENDEX1          ;SKIP TEST
016256  032737  010000  001366                  BIT     #BIT12,STAT1    ;LU PRESENT?
016264  001372                                  BNE     .-12            ;BR IF NO
016266  012737  000340  177776                  MOV     #340,PS         ;LOCK OUT INTERRUPTS
016274  013700  001366                          MOV     STAT1,R0        ;GET BR LEVEL
016300  006200                                  ASR     R0              ;SHIFT RIGHT 4 TIMES
016302  006200                                  ASR     R0
016304  006200                                  ASR     R0
016306  006200                                  ASR     R0
016310  042700  177437                          BIC     #177437,R0      ;PUT BR LEVEL IN R0
016314  012777  017132  163052                  MOV     #IISR,@DMRVEC   ;LOAD INPUT VECTOR
016322  010077  163050                          MOV     R0,@DMRLVL      ;LOAD LEVEL
016326  012777  017422  163044                  MOV     #OISR,@DMTVEC   ;LOAD OUTPUT VECTOR
016334  010077  163042                          MOV     R0,@DMTLVL      ;LOAD LEVEL

                                                ;INITIALIZE ALL BUFFER LISTS AND COUNT LISTS

016340  012737  000104  021316                  MOV     #104,TFLAG      ;TFLAG CONTAINS COUNT
016346  012700  020064                          MOV     #XMITBA+2,R0    ;R0 POINTS TO BA LIST
016352  012703  020356                          MOV     #RBUFF,R3       ;R3 CONTAINS BUFFER ADDRESS
016356  010320                          1$:     MOV     R3,(R0)+        ;LOAD BA LIST WITH REC BA
016360  062703  000104                          ADD     #104,R3         ;UPDATE BUFFER ADDRESS
016364  022700  020102                          CMP     #XMITBA+20,R0   ;END OF REC BUFFERS?
016370  001372                                  BNE     1$              ;NO LOAD NEXT ONE
016372  012720  020120                  2$:     MOV     #TBUFF,(R0)+    ;LOAD BA LIST WITH XMIT BA
016376  022700  020120                          CMP     #XMITBA+36,R0   ;END OF XMIT BUFFERS?
016402  001373                                  BNE     2$              ;NO LOAD NEXT BUFFER
016404  012700  020232                          MOV     #RCNTAB+2,R0    ;R0 POINTS TO COUNT LIST
016410  013720  021316                  3$:     MOV     TFLAG,(R0)+     ;LOAD COUNT OF 104
016414  022700  020250                          CMP     #RCNTAB+20,R0   ;END OF REC COUNT LIST?
016420  001373                                  BNE     3$              ;BR IF NO
016422  012737  000005  021314                  MOV     #5,FLAG ;LOOP COUNT
016430  012711  040000                          MOV     #BIT14,(R1)     ;SET MASTER CLEAR
016434  032737  100000  001366                  BIT     #BIT15,STAT1    ;IOP?
016442  001402                                  BEQ     .+6             ;BR IF NO
016444  012711  100000                          MOV     #BIT15,(R1)     ;SET RUN ON IOP
016450  012700  177777                          MOV     #-1,R0          ;R0 IS INPUT DONE COUNTER
016454  005037  020060                  CLRTAB: CLR     RESUME          ;CLEAR RESUME FLAG
016460  012705  020266                          MOV     #RDNTAB,R5      ;GET READY TO CLEAR ALL RECEIVE
016464  005025                          2$:     CLR     (R5)+           ;BUFFERS
016466  022705  021312                          CMP     #RBUFFE,R5      ;END OF BUFFER?
016472  001374                                  BNE     2$              ;BR IF NO
016474  012704  020250                          MOV     #XCNTAB,R4      ;R4 POINTS TO XMIT COUNT LIST
016500  013724  021316                  4$:     MOV     TFLAG,(R4)+     ;LOAD XMIT CHAR COUNT
016504  022704  020266                          CMP     #XCNTAB+16,R4   ;DONE?
016510  001373                                  BNE     4$              ;BR IF NO
016512  005002                          5$:     CLR     R2              ;R2 IS OUTPUT DONE COUNTER
016514  005004                                  CLR     R4              ;R4 IS USED AS INDEX IN OISR
```

```
016516  005711                                  TST     (R1)            ;IS RUN SET?
016520  100376                                  BPL     .-2             ;WAIT FOR RUN
016522  152761  000100  000002                  BISB    #BIT6,2(R1)     ;SET IEO
016530  032737  040000  001366                  BIT     #BIT14,STAT1    ;LOOP BACK CONNECTOR?
016536  001002                                  BNE     .+6             ;BR IF YES
016540  052711  004000                          BIS     #BIT11,(R1)     ;SET LINE UNIT LOOP
016544  022737  000005  021314                  CMP     #5,FLAG         ;FIRST TIME?
016552  001003                                  BNE     1$              ;BR IF NOT
016554  052711  000143                          BIS     #143,(R1)       ;SET IEI,RQI,BASE I
016560  000402                                  BR      3$              ;CONTINUE
016562  052711  000144                  1$:     BIS     #144,(R1)       ;SET IEI, RQI, REC BA/CC
016566  005037  001416                  3$:     CLR     TEMP            ;SET UP FOR DELAY COUNT
016572  012737  000022  001250                  MOV     #22,TEMP2       ;GET SET FOR DELAY
016600  005037  177776                          CLR     PS              ;ALLOW INTERRUPTS
016604  022700  000020                  SCAN:   CMP     #20,R0          ;INPUT DONE?
016610  001402                                  BEQ     SCAN2           ;BR IF YES
016612  000137  017102                          JMP     SCAN1           ;BR IF NO
016616  022702  000034                  SCAN2:  CMP     #34,R2          ;XMIT DONE FOR ALL MESSAGES?
016622  001402                                  BEQ     8$              ;BR IF YES
016624  000137  017102                          JMP     SCAN1           ;BR IF NO
016630  022704  000034                  8$:     CMP     #34,R4          ;REC DONE FOR ALL MESSAGES?
016634  001402                                  BEQ     9$              ;BR IF YES
016636  000137  017102                          JMP     SCAN1           ;BR IF NO
016642                                  9$:
016642  012700  020266                          MOV     #RDNTAB,R0      ;GET FIRST REC BUFFER
016646  012002                          5$:     MOV     (R0)+,R2        ;R2 POINTS TO BUFFER
016650  005005                                  CLR     R5              ;R5=EXPECTED
016652  005003                                  CLR     R3              ;R3 = COUNT
016654  010237  001252                  6$:     MOV     R2,TEMP3        ;SAVE ADDRESS FOR TYPEOUT
016660  112204                                  MOVB    (R2)+,R4        ;GET RECEIVE DATA
016662  120504                                  CMPB    R5,R4           ;IS IT CORRECT?
016664  001401                                  BEQ     .+4             ;BR IF YES
016666  104013                                  HLT     13              ;DATA ERROR
016670  005205                                  INC     R5              ;NEXT CHARACTER
016672  005203                                  INC     R3              ;INC COUNT
016674  021003                                  CMP     (R0),R3         ;DONE YET?
016676  001366                                  BNE     6$              ;BR IF NO
016700  062700  000002                          ADD     #2,R0           ;GET NEXT REC BUFFER
016704  022700  020322                          CMP     #RDNTAB+34,R0   ;DONE YET?
016710  001356                                  BNE     5$              ;BR IF NO
016712  012700  000001                          MOV     #1,R0           ;SET R0 TO 1
016716  032737  000001  021314          4$:     BIT     #BIT0,FLAG      ;CHANGE CHAR COUNT FOR NEXT LOOP
016724  001003                                  BNE     1$              ;BR TO SUB 40
016726  005337  021316                          DEC     TFLAG           ;DEC BY ONE
016732  000403                                  BR      2$              ;CONTINUE
016734  162737  000040  021316          1$:     SUB     #40,TFLAG       ;SUBTRACT 40 FRON XMIT COUNT
016742  005337  021314                  2$:     DEC     FLAG            ;DEC LOOP COUNT
016746  001242                                  BNE     CLRTAB          ;GO DO IT AGAIN
016750  152711  000146                  ENDEX1: BISB    #146,(R1)       ;SHUT DOWN DMC
016754  005737  021314                  1$:     TST     FLAG            ;HAS INTERRUPT OCCURED?
016760  001775                                  BEQ     1$              ;BR IF NO
016762  012700  017024                          MOV     #10$,R0         ;R0 POINTS TO LO SPEED COUNTS
016766  032737  000002  001372                  BIT     #BIT1,STAT3     ;IS IT LO SPEED?
016774  001402                                  BEQ     2$              ;BR IF YES
016776  012700  017034                          MOV     #11$,R0         ;R0 POINTS TO HI COUNTS
017002  012701  021443                  2$:     MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
```

```
017006  012702  000010                        MOV     #10,R2          ;10 COUNTS TO CHECK
017012  122021                       3$:      CMPB    (R0)+,(R1)+     ;CHECK COUNT
017014  103414                                BLO     ENDEX2          ;BR IF ERROR
017016  005302                                DEC     R2              ;DEC COUNT
017020  001374                                BNE     3$              ;BR IF NOT DONE
017022  000410                                BR      ENDEX1          ;ALL OK GET OUT
017024     000     000     000       10$:     .BYTE   0,0,0,0,0,0,5,5 ;EXPECTED LO SPEED COUNTS
017027     000     000     000
017032     005     005
017034     000     000     005       11$:     .BYTE   0,0,5,0,0,0,5,5 ;EXPECTED HI SPEED COUNTS
017037     000     000     000
017042     005     005
017044  104400                       ENDEX1:  SCOPE                   ;SCOPE THIS TEST
017046  113737  021443  001250       ENDEX2:  MOVB    BASE+3,TEMP2    ;SAVE ALL ODD ADDRESSES
017054  113737  021445  001252                MOVB    BASE+5,TEMP3    ;FOR TYPEOUT
017062  113737  021447  001254                MOVB    BASE+7,TEMP4    ;
017070  113737  021451  001256                MOVB    BASE+11,TEMP5
017076  104017                                HLT     17              ;NON ZERO ERROR COUNT
017100  000761                                BR      ENDEX1          ;GET OUT
017102  005337  001416               SCAN1:   DEC     TEMP            ;DECREMENT DELAY COUNTER
017106  001402                                BEQ     1$              ;BR IF ZERO
017110  000137  016604                        JMP     SCAN            ;BR IF NOT DONE DELAY
017114  005337  001250               1$:      DEC     TEMP2           ;DEC DELAY COUNT
017120  001402                                BEQ     2$              ;BR IF DONE DELAY
017122  000137  016604                        JMP     SCAN            ;BR IF NOT DONE
017126  104001                       2$:      HLT     1               ;ERROR HUNG
017130  000745                                BR      ENDEX1          ;GET OUT

                                              ;INPUT INTERRUPT SERVICE ROUTINE

017132  022700  000017               IISR:    CMP     #17,R0          ;PROC. ERROR DONE?
017136  001421                                BEQ     12$             ;BR IF YES
017140  005737  020060                        TST     RESUME          ;IS THIS A RESUME INTERRUPT
017144  001432                                BEQ     8$              ;BR IF NO
017146  032711  000002                        BIT     #BIT1,(R1)      ;CNTL OR BASE?
017152  001407                                BEQ     13$             ;BR IF CNTL I
017154  012761  021440  000004                MOV     #BASE,4(R1)     ;LOAD BASE ADDRESS
017162  012761  010000  000006                MOV     #BIT12,6(R1)    ;WITH RESUME BIT SET
017170  000404                                BR      12$             ;CONTINUE
017172  005061  000006               13$:     CLR     6(R1)           ;SELECT FULL DUPLEX
017176  005037  020060                        CLR     RESUME          ;CLEAR RESUME FLAG
017202  142711  000040               12$:     BICB    #40,(R1)        ;CLEAR RQI
017206  105711                                TSTB    (R1)            ;IS RDI GONE?
017210  100776                                BMI     .-2             ;BR IF NO
017212  005737  020060                        TST     RESUME          ;BASE OR CNTL I?
017216  001403                                BEQ     14$             ;BR IF IT WAS CNTL I
017220  152711  000041                        BISB    #41,(R1)        ;ASK FOR CNTL I
017224  000002                                RTI                     ;RETURN
017226  105011                       14$:     CLRB    (R1)            ;CLEAR BSEL 0
017230  000002                                RTI                     ;RETURN
017232  005700                       8$:      TST     R0              ;FIRST TIME HERE?
017234  100006                                BPL     7$              ;LOAD BASE IF MINUS
017236  012761  021440  000004                MOV     #BASE,4(R1)     ;SET UP BASE ADDRESS
017244  005061  000006                        CLR     6(R1)           ;CLEAR COUNT
017250  000434                                BR      3$              ;CONTINUE
017252  001003                       7$:      BNE     1$              ;CNTL I FULL DUPLEX IF 0
```

```
017254  005061  000006                        CLR     6(R1)           ;SELECT FULL DUPLEX
017260  000430                                BR      3$              ;CONTINUE
017262  032700  000010               1$:      BIT     #BIT3,R0        ;XMIT?
017266  001013                                BNE     2$              ;BR IF YES
017270  000241                                CLC                     ;CLEAR CARRY
017272  006100                                ROL     R0              ;MAKE R0 EVEN
017274  016061  020062  000004                MOV     RECBA(R0),4(R1) ;LOAD REC BUFFER
017302  016061  020230  000006                MOV     RCNTAB(R0),6(R1);LOAD COUNT
017310  000241                                CLC                     ;CLEAR CARRY
017312  006000                                ROR     R0              ;GET R0 BACK
017314  000412                                BR      3$              ;CONTINUE
017316  000241                       2$:      CLC                     ;CLEAR CARRY
017320  006100                                ROL     R0              ;MAKE IT EVEN
017322  016061  020062  000004                MOV     XMITBA(R0),4(R1);LOAD XMIT BUFFER
017330  016061  020230  000006                MOV     RCNTAB(R0),6(R1);LOAD COUNT
017336  000241                                CLC                     ;CLEAR CARRY
017340  006000                                ROR     R0              ;PUT IT BACK
017342  142711  000040               3$:      BICB    #40,(R1)        ;CLEAR RQI
017346  105711                                TSTB    (R1)            ;WAIT FOR
017350  100776                                BMI     .-2             ;RDI TO GO AWAY
017352  005200                                INC     R0              ;INC COUNT
017354  001003                                BNE     6$              ;IF 0 ASK FOR CNTL I
017356  152711  000041                        BISB    #41,(R1)        ;ASK FOR CNTL I
017362  000002                                RTI                     ;RETURN
017364  022700  000017               6$:      CMP     #17,R0          ;DONE YET?
017370  001411                                BEQ     4$              ;BR IF YES
017372  032700  000010                        BIT     #BIT3,R0        ;XMIT?
017376  001003                                BNE     5$              ;BR IF YES
017400  152711  000044                        BISB    #44,(R1)        ;ASK FOR REC BA/CC
017404  000002                                RTI                     ;RETURN
017406  152711  000040               5$:      BISB    #40,(R1)        ;ASK FOR XMIT BA/CC
017412  000002                                RTI                     ;RETURN
017414  152711  000046               4$:      BISB    #46,(R1)        ;FORCE PROC. ERROR
017420  000002                                RTI                     ;RETURN

                                              ;OUTPUT INTERRUPT SERVICE ROUTINE

017422  032761  000001  000002       OISR:    BIT     #BIT0,2(R1)     ;IS THIS AN ERROR?
017430  001467                                BEQ     1$              ;BR IF NO
017432  005737  021314                        TST     FLAG            ;IS THIS SHUT DOWN INTERRUPT?
017436  001006                                BNE     9$              ;BR IF NO
017440  005237  021314                        INC     FLAG            ;YES MAKE FLAG NON-ZERO
017444  022761  001000  000006                CMP     #BIT9,6(R1)     ;SHUT DOWN BIT SET?
017452  001531                                BEQ     10$             ;YES ALL IS OK
017454  022700  000017               9$:      CMP     #17,R0          ;RESUME INTERRUPT?
017460  001041                                BNE     11$             ;BR IF NO
017462  022761  001000  000006                CMP     #BIT9,6(R1)     ;PROC. ERROR BIT SET?
017470  001035                                BNE     11$             ;BR IF NO
017472  005200                                INC     R0              ;BUMP COUNTER (TO 20)
017474  012711  040000                        MOV     #BIT14,(R1)     ;MASTER CLEAR DEVICE
017500  032737  100000  001366                BIT     #BIT15,STAT1    ;DMC OR KMC?
017506  001405                                BEQ     .+14            ;BR IF DMC
017510  012711  100000                        MOV     #BIT15,(R1)     ;SET RUN ON KMC
017514  105227  000000                        INCB    #0              ;DELAY ON KMC
017520  001375                                BNE     .-4
017522  012737  177777  020060                MOV     #-1,RESUME      ;SET RESUME FLAG
```

```
017530  005711                                TST     (R1)              ;RUN SET?
017532  100376                                BPL     .-2               ;BR IF NO
017534  012761  000100  000002                MOV     #BIT6,2(R1)       ;SET IEO
017542  032737  040000  001366                BIT     #BIT14,STAT1      ;LOOP BACK CONNECTOR?
017550  001002                                BNE     .+6               ;BR IF YES
017552  052711  004000                        BIS     #BIT11,(R1)       ;SET LINE UNIT LOOP
017556  052711  000143                        BIS     #143,(R1)         ;ASK FOR PORT (BASE REQUEST)
017562  000002                                RTI                       ;RETURN
017564  016137  000004  001252        11$:    MOV     4(R1),TEMP3       ;SAVE FOR ERROR TYPEOUT
017572  016137  000006  001254                MOV     6(R1),TEMP4       ;SAVE FOR ERROR TYPEOUT
017600  104016                                HLT     16                ;CNTL O ERROR
017602  022626                                CMP     (SP)+,(SP)+       ;ADJUST STACK
017604  000137  017044                        JMP     ENDEX1            ;GET OUT
017610  032761  000004  000002        1$:     BIT     #BIT2,2(R1)       ;RECEIVE?
017616  001053                                BNE     2$                ;BR IF YES
017620  022761  020120  000004                CMP     #TBUFF,4(R1)      ;IS XMIT BA CORRECT?
017626  001412                                BEQ     4$                ;BR IF OK
017630  022761  020121  000004                CMP     #TBUFF+1,4(R1)    ;IS XMIT BA CORRECT?
017636  001406                                BEQ     4$                ;BR IF YES
017640  012705  020120                        MOV     #TBUFF,R5         ;R5 = EXPECTED
017644  016137  000004  001252                MOV     4(R1),TEMP3       ;SAVE FOUND FOR TYPEOUT
017652  104002                                HLT     2                 ;XMIT BA ERROR
017654  005005                        4$:     CLR     R5                ;R5 IS INDEX REG
017656  026561  020250  000006        5$:     CMP     XCNTAB(R5),6(R1);IS CHAR COUNT OK?
017664  001406                                BEQ     6$                ;BR IF YES
017666  062705  000002                        ADD     #2,R5             ;INC INDEX
017672  022705  000016                        CMP     #16,R5            ;DONE LIST YET?
017676  001367                                BNE     5$                ;BR IF NO
017700  104003                                HLT     3                 ;XMIT COUNT ERROR
017702  016162  000004  020322        6$:     MOV     4(R1),XDNTAB(R2);STORE XMIT DONE BA
017710  062702  000002                        ADD     #2,R2             ;INC INDEX
017714  016162  000006  020322                MOV     6(R1),XDNTAB(R2);STORE XMIT DONE CC
017722  062702  000002                        ADD     #2,R2             ;INC INDEX
017726  142761  000207  000002                BICB    #207,2(R1)        ;CLEAR RDO
017734  000002                                RTI                       ;RETURN
017736  105011                        10$:    CLRB    (R1)              ;CLEAR SEL0
017740  105061  000002                        CLRB    2(R1)             ;CLEAR SEL2
017744  000002                                RTI                       ;RETURN
017746  012705  000002                2$:     MOV     #2,R5             ;SET UP R5 AS INDEX
017752  026561  020062  000004                CMP     RECBA(R5),4(R1) ;COMPARE WITH LIST OF CORRECT BA'S
017760  001406                                BEQ     3$                ;BR IF OK?
017762  062705  000002                        ADD     #2,R5             ;INCREMENT R5
017766  022705  000020                        CMP     #20,R5            ;END OF LIST?
017772  001367                                BNE     2$+4              ;BR IF NO
017774  104004                                HLT     4                 ;REC BA ERROR
017776  005005                        3$:     CLR     R5                ;R5 IS INDEX
020000  026561  020250  000006        7$:     CMP     XCNTAB(R5),6(R1);CHECK FOR CORRECT REC COUNT
020006  001406                                BEQ     8$                ;BR IF YES
020010  062705  000002                        ADD     #2,R5             ;INCREMENT R5
020014  022705  000016                        CMP     #16,R5            ;END OF LIST?
020020  001367                                BNE     7$                ;BR IF NOT
020022  104005                                HLT     5                 ;REC COUNT ERROR
020024  016164  000004  020266        8$:     MOV     4(R1),RDNTAB(R4)  ;STORE REC BA
020032  062704  000002                        ADD     #2,R4             ;INC INDEX
020036  016164  000006  020266                MOV     6(R1),RDNTAB(R4);STORE REC DONE CC
020044  062704  000002                        ADD     #2,R4             ;INC INDEX
```

```
020050  142761  000207  000002                BICB    #207,2(R1)        ;CLEAR RDO
020056  000002                                RTI                       ;RETURN


                                              ;BUFFERS

020060  000000                        RESUME: 0
020062                                RECBA:
020062  000017                        XMITBA: .BLKW   17        ;REC & XMIT BA LIST

020120                                TBUFF:                            ;TRANSMIT DATA
020120     000     001     002        .BYTE   0,1,2,3,4,5,6,7
020123     003     004     005
020126     006     007
020130     010     011     012        .BYTE   10,11,12,13,14,15,16,17
020133     013     014     015
020136     016     017
020140     020     021     022        .BYTE   20,21,22,23,24,25,26,27
020143     023     024     025
020146     026     027
020150     030     031     032        .BYTE   30,31,32,33,34,35,36,37
020153     033     034     035
020156     036     037
020160     040     041     042        .BYTE   40,41,42,43,44,45,46,47
020163     043     044     045
020166     046     047
020170     050     051     052        .BYTE   50,51,52,53,54,55,56,57
020173     053     054     055
020176     056     057
020200     060     061     062        .BYTE   60,61,62,63,64,65,66,67
020203     063     064     065
020206     066     067
020210     070     071     072        .BYTE   70,71,72,73,74,75,76,77
020213     073     074     075
020216     076     077
020220     100     101     102        .BYTE   100,101,102,103,104,105,106,107
020223     103     104     105
020226     106     107

020230  000010                        RCNTAB: .BLKW   10                ;RECEIVE COUNT TABLE
020250  000007                        XCNTAB: .BLKW   7                 ;TRANSMIT COUNT TABLE

020266  000016                        RDNTAB: .BLKW   16                ;RECEIVE DONE TABLE (BA/CC)
020322  000016                        XDNTAB: .BLKW   16                ;XMIT DONE TABLE (BA/CC)

020356                                RBUFF:                            ;RECEIVER BUFFERS
020356  000104                        RBUFF1:  .BLKB 104
020462  000104                        RBUFF2:  .BLKB 104
020566  000104                        RBUFF3:  .BLKB 104
020672  000104                        RBUFF4:  .BLKB 104
020776  000104                        RBUFF5:  .BLKB 104
021102  000104                        RBUFF6:  .BLKB 104
021206  000104                        RBUFF7:  .BLKB 104
021312  000000                        RBUFFE: 0                         ;END OF RECEIVER BUFFERS
                                 81800
                                 81900
```

```
 2728                                      82000
 2729                                      82100             ;BUFFER AREA
 2730                                      82200             ;===========
 2731                                      82300
 2732  021314  000000                      82400   FLAG:     0
 2733  021316  000000                      82500   TFLAG:    0
 2734  021320  000000                      82600   PFLAG:    0
 2735  021322  000044                      82700   TCOUNT:   44
 2736  021324  041101  042103  043105      82800   TBUF:     .ASCII/ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789/
 2737  021332  044107  045111  046113
 2738  021340  047115  050117  051121
 2739  021346  052123  053125  054127
 2740  021354  055131  030460  031462
 2741  021362  032464  033466  034470
 2742                                      82900   .EVEN
 2743  021370  000044                      83000   RCOUNT:   44
 2744  021372  021440                      83100   RBUF:     .=.+46
 2745                                      83200   .EVEN
 2746  021440  022040                      83300   BASE:     .=.+256.
 2747                                      00300
 2748                                      00400
 2749                                      00500   ;SUBROUTINES
 2750                                      00600   ;-----------
 2751                                      00700
 2752                                      00800
 2753  022040                              00900   BASELD:
 2754                                      01000             ;THIS SUBROUTINE LOADS THE DMC WITH A BASE ADDRESS
 2755                                      01100             ;AND PUTS DMC INTO FULL-DUPLEX MODE
 2756                                      01200
 2757  022040  012711  040000              01300             MOV     #BIT14,(R1)       ;MASTER CLEAR
 2758  022044  032737  100000  001366      01400             BIT     #BIT15,STAT1      ;CRAM?
 2759  022052  001402                      01500             BEQ     .+6               ;BR IF NO
 2760  022054  012711  100000              01600             MOV     #BIT15,(R1)       ;IF CRAM SET RUN
 2761  022060  105227  000000              01700             INCB    #0                ;DELAY
 2762  022064  001375                      01800             BNE     .-4               ;BR IF NOT DONE DELAY
 2763  022066  005711                      01900   1$:       TST     (R1)              ;IS RUN SET?
 2764  022070  100376                      02000             BPL     1$                ;BR IF NO
 2765  022072  052711  004000              02100             BIS     #BIT11,(R1)       ;SET LU LOOP
 2766  022076  152711  000043              02200             BISB    #43,(R1)          ;BASE REQUEST
 2767  022102  105711                      02300   2$:       TSTB    (R1)              ;RDY I SET?
 2768  022104  100376                      02400             BPL     2$                ;BR IF NO
 2769  022106  012761  021440  000004      02500             MOV     #BASE,4(R1)       ;LOAD BASE ADDRESS
 2770  022114  005061  000006              02600             CLR     6(R1)             ;CLEAR CC
 2771  022120  142711  000040              02700             BICB    #40,(R1)          ;CLEAR RQI
 2772  022124  105711                      02800   3$:       TSTB    (R1)              ;RDY I CLEAR?
 2773  022126  100776                      02900             BMI     3$                ;BR IF NO
 2774  022130  152711  000041                                BISB    #41,(R1)          ;ASK FOR CNTL I
 2775  022134  105711                              64$:      TSTB    (R1)              ;WAIT FOR RDI
 2776  022136  100376                                        BPL     64$               ;BR IF NOT SETY
 2777  022140  005061  000006                                CLR     6(R1)             ;SET FULL DUPLEX
 2778  022144  142711  000040                                BICB    #40,(R1)          ;CLEAR RQI
 2779  022150  105711                              65$:      TSTB    (R1)              ;RDI UP?
 2780  022152  100776                                        BMI     65$               ;BR IF YES
 2781  022154  000207                      03100             RTS     PC                ;RETURN
 2782                                      03200
 2783  022156                              03300   BASELH:
```

```
 2784                                      03400             ;THIS SUBROUTINE LOADS THE DMC WITH A BASE ADDRESS
 2785                                      03500             ;AND PUTS DMC INTO HALF-DUPLEX MODE
 2786                                      03600
 2787  022156  012711  040000              03700             MOV     #BIT14,(R1)       ;MASTER CLEAR
 2788  022162  032737  100000  001366      03800             BIT     #BIT15,STAT1      ;CRAM?
 2789  022170  001402                      03900             BEQ     .+6               ;BR IF NO
 2790  022172  012711  100000              04000             MOV     #BIT15,(R1)       ;IF CRAM SET RUN
 2791  022176  105227  000000              04100             INCB    #0                ;DELAY
 2792  022202  001375                      04200             BNE     .-4               ;BR IF NOT DONE DELAY
 2793  022204  005711                      04300   1$:       TST     (R1)              ;IS RUN SET?
 2794  022206  100376                      04400             BPL     1$                ;BR IF NO
 2795  022210  052711  004000              04500             BIS     #BIT11,(R1)       ;SET LU LOOP
 2796  022214  152711  000043              04600             BISB    #43,(R1)          ;BASE REQUEST
 2797  022220  105711                      04700   2$:       TSTB    (R1)              ;RDY I SET?
 2798  022222  100376                      04800             BPL     2$                ;BR IF NO
 2799  022224  012761  021440  000004      04900             MOV     #BASE,4(R1)       ;LOAD BASE ADDRESS
 2800  022232  005061  000006              05000             CLR     6(R1)             ;CLEAR CC
 2801  022236  142711  000040              05100             BICB    #40,(R1)          ;CLEAR RQI
 2802  022242  105711                      05200   3$:       TSTB    (R1)              ;RDY I CLEAR?
 2803  022244  100776                      05300             BMI     3$                ;BR IF NO
 2804  022246  152711  000041                                BISB    #41,(R1)          ;ASK FOR CNTL I
 2805  022252  105711                              64$:      TSTB    (R1)              ;WAIT FOR RDI
 2806  022254  100376                                        BPL     64$               ;BR IF NOT SETY
 2807  022256  012761  002000  000006                        MOV     #BIT10,6(R1)      ;SET HALF DUPLEX
 2808  022264  142711  000040                                BICB    #40,(R1)          ;CLEAR RQI
 2809  022270  105711                              65$:      TSTB    (R1)              ;RDI UP?
 2810  022272  100776                                        BMI     65$               ;BR IF YES
 2811  022274  000207                      05500             RTS     PC                ;RETURN
 2812                                      05600
 2813  022276                              05700   RESUM:
 2814                                      05800             ;THIS SUBROUTINE LOADS THE DMC WITH A BASE ADDRESS
 2815                                      05900             ;WITH RESUME BIT SET AND PUTS DMC INTO FULL-DUPLEX MODE
 2816                                      06000
 2817  022276  012711  040000              06100             MOV     #BIT14,(R1)       ;MASTER CLEAR
 2818  022302  032737  100000  001366      06200             BIT     #BIT15,STAT1      ;CRAM?
 2819  022310  001402                      06300             BEQ     .+6               ;BR IF NO
 2820  022312  012711  100000              06400             MOV     #BIT15,(R1)       ;IF CRAM SET RUN
 2821  022316  105227  000000              06500             INCB    #0                ;DELAY
 2822  022322  001375                      06600             BNE     .-4               ;BR IF NOT DONE DELAY
 2823  022324  005711                      06700   1$:       TST     (R1)              ;IS RUN SET?
 2824  022326  100376                      06800             BPL     1$                ;BR IF NO
 2825  022330  052711  004000              06900             BIS     #BIT11,(R1)       ;SET LU LOOP
 2826  022334  152711  000043              07000             BISB    #43,(R1)          ;BASE REQUEST
 2827  022340  105711                      07100   2$:       TSTB    (R1)              ;RDY I SET?
 2828  022342  100376                      07200             BPL     2$                ;BR IF NO
 2829  022344  012761  021440  000004      07300             MOV     #BASE,4(R1)       ;LOAD BASE ADDRESS
 2830  022352  012761  010000  000006      07400             MOV     #BIT12,6(R1)      ;SET RESUME BIT
 2831  022360  142711  000040              07500             BICB    #40,(R1)          ;CLEAR RQI
 2832  022364  105711                      07600   3$:       TSTB    (R1)              ;RDY I CLEAR?
 2833  022366  100776                      07700             BMI     3$                ;BR IF NO
 2834  022370  152711  000041                                BISB    #41,(R1)          ;ASK FOR CNTL I
 2835  022374  105711                              64$:      TSTB    (R1)              ;WAIT FOR RDI
 2836  022376  100376                                        BPL     64$               ;BR IF NOT SETY
 2837  022400  005061  000006                                CLR     6(R1)             ;SET FULL DUPLEX
 2838  022404  142711  000040                                BICB    #40,(R1)          ;CLEAR RQI
 2839  022410  105711                              65$:      TSTB    (R1)              ;RDI UP?
```

```
  2840  022412  100776                                     BMI     65$             ;BR IF YES
  2841  022414  000207                     07900           RTS     PC              ;RETURN
  2842                                     08000
  2843  022416                             08100   RFRELD:
  2844                                     08200           ;THIS SUBROUTINE LOADS THE DMC WITH A RECEIVE BA/CC
  2845                                     08300
  2846  022416  152711  000044             08400           BISB    #44,(R1)        ;REC BA/CC REQUEST
  2847  022422  105711                     08500   1$:     TSTB    (R1)            ;RDY I SET?
  2848  022424  100376                     08600           BPL     1$              ;BR IF NO
  2849  022426  012561  000004             08700           MOV     (R5)+,4(R1)     ;LOAD REC BA
  2850  022432  012561  000006             08800           MOV     (R5)+,6(R1)     ;LOAD REC CC
  2851  022436  142711  000040             08900           BICB    #40,(R1)        ;CLEAR RQI
  2852  022442  105711                     09000   2$:     TSTB    (R1)            ;IS RDY I CLEAR
  2853  022444  100776                     09100           BMI     2$              ;BR IF NO
  2854  022446  000205                     09200           RTS     R5              ;RETURN
  2855                                     09300
  2856  022450                             09400   XFRELD:
  2857                                     09500           ;THIS SUBROUTINE LOADS THE DMC WITH A TRANSMIT BA/CC
  2858                                     09600
  2859  022450  152711  000040             09700           BISB    #40,(R1)        ;XMIT BA/CC REQUEST
  2860  022454  105711                     09800   1$:     TSTB    (R1)            ;RDY I SET?
  2861  022456  100376                     09900           BPL     1$              ;BR IF NO
  2862  022460  012561  000004             10000           MOV     (R5)+,4(R1)     ;LOAD XMIT BA
  2863  022464  012561  000006             10100           MOV     (R5)+,6(R1)     ;LOAD XMIT CC
  2864  022470  142711  000040             10200           BICB    #40,(R1)        ;CLEAR RQI
  2865  022474  105711                     10300   2$:     TSTB    (R1)            ;IS RDY I CLEAR
  2866  022476  100776                     10400           BMI     2$              ;BR IF NO
  2867  022500  000205                     10500           RTS     R5              ;RETURN
  2868                                     10600
  2869                                     10700
  2870  022502                             10800   SHUTDOWN:
  2871                                     10900           ;THIS SUBROUTINE FORCES THE DMC TO UPDATE THE BASE TABLE
  2872                                     11000
  2873  022502  042761  000207  000002     11100           BIC     #207,2(R1)      ;CLEAR ANY OUTPUT DONES
  2874  022510  152711  000046             11200           BISB    #46,(R1)        ;ASK FOR ILLEGAL REQUEST
  2875  022514  105711                     11300   1$:     TSTB    (R1)            ;RDI SET?
  2876  022516  100376                     11400           BPL     1$              ;BR IF NO
  2877  022520  142711  000040             11500           BICB    #40,(R1)        ;CLEAR RQI
  2878  022524  105761  000002             11600   2$:     TSTB    2(R1)           ;OUTPUT DONE SET?
  2879  022530  100375                     11700           BPL     2$              ;BR IF NOT
  2880  022532  000207                     11800           RTS     PC              ;RETURN
  2881                                     11900
  2882                                     00300
        022534  052377  040522  051516     00400   EM2:    .ASCIZ  <377>/TRANSMIT BA ERROR/
        022557     377  051124  047101     00500   EM3:    .ASCIZ  <377>/TRANSMIT COUNT ERROR/
        022605     377  042522  042503     00600   EM4:    .ASCIZ  <377>/RECEIVE BA ERROR/
        022627     377  042522  042503     00700   EM5:    .ASCIZ  <377>/RECEIVE COUNT ERROR/
        022654  051377  041505  044505     00800   EM11:   .ASCIZ  <377>/RECEIVE DATA ERROR/
        022700  043377  042522  020105     00900   EM12:   .ASCIZ  <377>/FREE RUNNING ERROR/
        022724  041777  047117  051124     01000   EM13:   .ASCIZ  <377>/CONTROL OUT ERROR/
        022747     377  047111  042524     01100   EM14:   .ASCIZ  <377>/INTERNAL DDCMP ERROR COUNTS NON ZERO/
                                           01200
        023015     377  054105  042520     01300   DH1:    .ASCIZ  <377>/EXPECTED  FOUND  ADDRESS/
        023047     377  054105  042520     01400   DH2:    .ASCIZ  <377>/EXPECTED  FOUND/
        023070  020377  042523  032114     01500   DH3:    .ASCIZ  <377>/ SEL4      SEL6/
        023111     377  040502  042523     01600   DH4:    .ASCIZ  <377>/BASE+3 THRU BASE+12 /
```

```
        023137     377  046504  030503     01700   DH5:    .ASCIZ  <377>/DMC11 IS HUNG/
                                           01800   .EVEN
                                           01900
        023156  000003                     02000   DT1:    3
        023160     006     004             02100           .BYTE   6,4
        023162  001264                     02200           SAVR2
        023164     006     004             02300           .BYTE   6,4
        023166  001270                     02400           SAVR4
        023170     004     002             02500           .BYTE   4,2
        023172  001260                     02600           SAVR0
        023174  000003                     02700   DT2:    3
        023176     006     004             02800           .BYTE   6,4
        023200  001272                     02900           SAVR5
        023202     006     004             03000           .BYTE   6,4
        023204  001270                     03100           SAVR4
        023206     004     002             03200           .BYTE   4,2
        023210  001264                     03300           SAVR2
        023212  000003                     03400   DT3:    3
        023214     006     004             03500           .BYTE   6,4
        023216  001272                     03600           SAVR5
        023220     006     004             03700           .BYTE   6,4
        023222  001270                     03800           SAVR4
        023224     004     002             03900           .BYTE   4,2
        023226  001252                     04000           TEMP3
        023230  000002                     04100   DT4:    2
        023232     003     007             04200           .BYTE   3,7
        023234  001272                     04300           SAVR5
        023236     003     002             04400           .BYTE   3,2
        023240  001270                     04500           SAVR4
        023242  000002                     04600   DT5:    2
        023244     006     004             04700           .BYTE   6,4
        023246  001272                     04800           SAVR5
        023250     006     002             04900           .BYTE   6,2
        023252  001270                     05000           SAVR4
        023254  000003                     05100   DT6:    3
        023256     003     010             05200           .BYTE   3,10
        023260  001272                     05300           SAVR5
        023262     003     004             05400           .BYTE   3,4
        023264  001270                     05500           SAVR4
        023266     004     002             05600           .BYTE   4,2
        023270  021314                     05700           FLAG
        023272  000003                     05800   DT7:    3
        023274     003     010             05900           .BYTE   3,10
        023276  001272                     06000           SAVR5
        023300     003     004             06100           .BYTE   3,4
        023302  001270                     06200           SAVR4
        023304     004     002             06300           .BYTE   4,2
        023306  001264                     06400           SAVR2
        023310  000003                     06500   DT10:   3
        023312     003     007             06600           .BYTE   3,7
        023314  001272                     06700           SAVR5
        023316     003     004             06800           .BYTE   3,4
        023320  001270                     06900           SAVR4
        023322     006     002             07000           .BYTE   6,2
        023324  001252                     07100           TEMP3
        023326  000002                     07200   DT11:   2
        023330     006     004          07300           .BYTE   6,4
        023332  001252                  07400           TEMP3
        023334     006     002          07500           .BYTE   6,2
        023336  001254                  07600           TEMP4
        023340  000010                  07700   DT12:   10
        023342     003     002          07800           .BYTE   3,2
        023344  001250                  07900           TEMP2
        023346     003     002          08000           .BYTE   3,2
        023350  021444                  08100           BASE+4
        023352     003     002          08200           .BYTE   3,2
        023354  001252                  08300           TEMP3
        023356     003     002          08400           .BYTE   3,2
        023360  021446                  08500           BASE+6
        023362     003     002          08600           .BYTE   3,2
        023364  001254                  08700           TEMP4
        023366     003     002          08800           .BYTE   3,2
        023370  021450                  08900           BASE+10
        023372     003     002          09000           .BYTE   3,2
        023374  001256                  09100           TEMP5
        023376     003     002          09200           .BYTE   3,2
        023400  021452                  09300           BASE+12
        023402  000002                  09400   DT13:   2
        023404     006     004          09500           .BYTE   6,4
        023406  001272                  09600           SAVR5
        023410     006     002          09700     .     .BYTE   6,2
        023412  001270                  09800           SAVR4
                                        09900
        023414                          10000   .ERRTAB:
        023414  000000                  10100           0
        023416  000000                  10200           0
        023420  000000                  10300           0
        023422  022700                  10400   EM12
        023424  023137                  10500   DH5     ;HLT    1
        023426  000000                  10600   0
        023430  022534                  10700   EM2
        023432  023047                  10800   DH2     ;HLT    2
        023434  023402                  10900   DT13
        023436  022557                  11000   EM3
        023440  000000                  11100   0       ;HLT    3
        023442  000000                  11200   0
        023444  022605                  11300   EM4
        023446  000000                  11400   0       ;HLT    4
        023450  000000                  11500   0
        023452  022627                  11600   EM5
        023454  000000                  11700   0
        023456  000000                  11800   0
        023460  022605                  11900   EM4
        023462  023047                  12000   DH2     ;HLT    6
        023464  023242                  12100   DT5
        023466  022627                  12200   EM5
        023470  023047                  12300   DH2     ;HLT    7
        023472  023230                  12400   DT4
        023474  000000                  12500   0
        023476  023015                  12600   DH1     ;HLT    10
        023500  023254                  12700   DT6
        023502  000000                  12800   0
```

```
        023504  023015                  12900   DH1     ;HLT    11
        023506  023272                  13000   DT7
        023510  000000                  13100   0
        023512  023047                  13200   DH2     ;HLT    12
        023514  023230                  13300   DT4
        023516  022654                  13400   EM11
        023520  023015                  13500   DH1     ;HLT    13
        023522  023310                  13600   DT10
        023524  022700                  13700   EM12
        023526  000000                  13800   0       ;HLT    14
        023530  000000                  13900   0
        023532  022700                  14000   EM12
        023534  023047                  14100   DH2     ;HLT    15
        023536  023242                  14200   DT5
        023540  022724                  14300   EM13
        023542  023070                  14400   DH3     ;HLT    16
        023544  023326                  14500   DT11
        023546  022747                  14600   EM14
        023550  023111                  14700   DH4     ;HLT    17
        023552  023340                  14800   DT12
                                        14900
                                        15000
        023554                          15100   COPY[illegible]:
                000001                  15600   .END
```

```
ADRCNT= 004373         879*    915*    924#
AUDONE  003024         569     608     613     659*
AUSTRT  002446         568#    663
AUTO,S  010512         526     1368#
BASE    021440         1720    1838    1848    1849    1850    1851    1912    1925    1926    1927    1928    1984    1994
                       1995    1996    1997    2047    2057    2058    2059    2060    2113    2123    2124    2125    2126
                       2173    2183    2184    2185    2186    2233    2243    2244    2245    2246    2358    2371    2372
                       2373    2374    2503    2517    2518    2519    2520    2540    2556    2746#   2769    2799    2829
                       2882
BASELD  022040         1875    1956    2022    2085    2151    2211    2315    2753#
BASELH  022156         2271    2783#
BINWRD  004714         965*    968*    969     1006*
BIT0  = 000001         95#     1155    1156    1687    1793    1808    1870    1890    1951    1973    2017    2036    2080
                       2102    2146    2163    2206    2223    2266    2307    2398    2489    2597
BIT1  = 000002         94#     531     1149    1155    1156    1449    1532    1545    1909    2355    2500    2538
BIT10 = 002000         85#     1513    1524    2807
BIT11 = 004000         84#     2453    2621    2765    2795    2825
BIT12 = 010000         83#     1464    1543    1690    1873    1954    2020    2083    2149    2209    2269    2310    2401
                       2541    2830
BIT13 = 020000         82#     1467    1516    1547    1906
BIT14 = 040000         81#     781     1478    1480    1547    1558    1700    2432    2451    2609    2619    2757    2787
                       2817
BIT15 = 100000         80#     485     572     575     596     599     1451    1519    1685    1701    1703    1868    1949
                       2015    2078    2144    2204    2264    2305    2396    2433    2435    2610    2612    2758    2760
                       2788    2790    2818    2820
BIT2  = 000004         93#     531     712     1156    1787    1894    2629
BIT3  = 000010         92#     1549    1556    2562    2586
BIT4  = 000020         91#     1139    1162    1164    1977
BIT5  = 000040         90#     1661
BIT6  = 000100         89#     1144    1145    1551    2450    2618
BIT7  = 000200         88#     1145    1265    1661
BIT8  = 000400         87#     1523    1538    1540    1553    1555    1561    1564    1622    2040    2106
BIT9  = 001000         86#     1521    1523    1535    1561    1564    1620    1622    2167    2227    2602    2606
BM      007054         1187#   1492
BRLVL   012252         1617    1627    1635    1647#
BRW     003730         718     804#
BRX     003732         719     805#
CHRCNT  004712         963*    966     970     986*    1004#   1005
CKSWR   007606         512     779     811     1026    1212#
CKSWR1  007666         1225#   1237
CKSWR2  007700         1228#
CKSWR3  007704         1230#
CKSWR4  007710         1231#   1239    1246
CKSWR5  010014         1213    1220    1255#
CLKX    001242         171#
CLRTAB  016454         2437#   2495
CNERR   007277         626     1187#
CNT,MA  001702         196     364#    484     486     488     1293
CNVRT = 104411         233#    623     738     740     742     744     1060    1062    1122    1228
CONERR  007223         621     1187#
CONN    007114         1187#   1469
CONTAB  002776         630     643#
CONVRT= 104410         231#    543     629     1076    1391
CORMAX  023554         2882#
CRAM    006606         1187#   1430
CREAM   001320         195#    483*    1289*   1290    1292*   1296
```

```
CSR     006510         1187#   1395
CSRMAP  010514         1370#
CYCLE   010060         720     761     762     1280#
DATABP  005216         1049*   1052    1074    1077#
DATACL= 104415         241#
DATAHD  005204         1048*   1070    1073#
DELAY = 104413         237#
DEVADR  004370         877*    912     922#
DEVTAB  003010         583     648#
DH1     023015         2882#
DH2     023047         2882#
DH3     023070         2882#
DH4     023111         2882#
DH5     023137         2882#
DISPLA  001200         142#    498*    504*    735*
DISPRE  000174         128#    504
DMACTV  001306         189#    521     676*    677     1280    1294    1376*   1577*   1583*   1584*   1588    1613
DMCM    007320         634     1187#
DMCR00  001500         279#
DMCR01  001510         284#
DMCR02  001520         289#
DMCR03  001530         294#
DMCR04  001540         299#
DMCR05  001550         304#
DMCR06  001560         309#
DMCR07  001570         314#
DMCR10  001600         319#
DMCR11  001610         324#
DMCR12  001620         329#
DMCR13  001630         334#
DMCR14  001640         339#
DMCR15  001650         344#
DMCR16  001660         349#
DMCR17  001670         354#
DMCSR   001404         262#    568*    602     607*    612*    647     765     802     1094    1117    1163    1298*   1307
                       1356
DMCSRH  001406         263#    1144*   1145*   1149*   1155*   1156*   1162*   1164*   1307*   1308*   1309
DMCTL   001410         264#    1309*   1310*   1311
DMNUM   001310         190#    479     751     1374*   1389*   1568*   1569    1578    1580
DMPO4   001412         265#    1133*   1139    1176    1181    1311*   1312*   1313
DMPO6   001414         266#    1150*   1313*   1314*
DMRLVL  001376         259#    1316*   1317*   1318    2411*
DMRVEC  001374         258#    768     1299*   1300*   1316    2410*
DMS100  001502         280#
DMS101  001512         285#
DMS102  001522         290#
DMS103  001532         295#
DMS104  001542         300#
DMS105  001552         305#
DMS106  001562         310#
DMS107  001572         315#
DMS110  001602         320#
DMS111  001612         325#
DMS112  001622         330#
DMS113  001632         335#
DMS114  001642         340#
```

```
DMS115  001652          345#
DMS116  001662          350#
DMS117  001672          355#
DMS200  001504          281#
DMS201  001514          286#
DMS202  001524          291#
DMS203  001534          296#
DMS204  001544          301#
DMS205  001554          306#
DMS206  001564          311#
DMS207  001574          316#
DMS210  001604          321#
DMS211  001614          326#
DMS212  001624          331#
DMS213  001634          336#
DMS214  001644          341#
DMS215  001654          346#
DMS216  001664          351#
DMS217  001674          356#
DMS300  001506          282#
DMS301  001516          287#
DMS302  001526          292#
DMS303  001536          297#
DMS304  001546          302#
DMS305  001556          307#
DMS306  001566          312#
DMS307  001576          317#
DMS310  001606          322#
DMS311  001616          327#
DMS312  001626          332#
DMS313  001636          337#
DMS314  001646          342#
DMS315  001656          347#
DMS316  001666          352#
DMS317  001676          357#
DMTLVL  001402          261#   1320*   1321*   2413*
DMTVEC  001400          260#   1318*   1319*   1320    2412*
DM,END  001700          359#   1372
DM,MAP  001500          195     278#    483     536     546     662    1290    1292    1370    1375    1605
DONE    003734          784     786*    806#   1216*
DT1     023156         2882#
DT10    023310         2882#
DT11    023326         2882#
DT12    023340         2882#
DT13    023402         2882#
DT2     023174         2882#
DT3     023212         2882#
DT4     023230         2882#
DT5     023242         2882#
DT6     023254         2882#
DT7     023272         2882#
EM11    022654         2882#
EM12    022700         2882#
EM13    022724         2882#
EM14    022747         2882#
EM2     022534         2882#
```

```
EM3     022557         2882#
EM4     022605         2882#
EM5     022627         2882#
ENDEX   016750         2496#
ENDEX1  017044         2400    2509    2516#   2522    2530    2628
ENDEX2  017046         2506    2517#
ERCT00  001704          366#
ERCT01  001710          369#
ERCT02  001714          372#
ERCT03  001720          375#
ERCT04  001724          378#
ERCT05  001730          381#
ERCT06  001734          384#
ERCT07  001740          387#
ERCT10  001744          390#
ERCT11  001750          393#
ERCT12  001754          396#
ERCT13  001760          399#
ERCT14  001764          402#
ERCT15  001770          405#
ERCT16  001774          408#
ERCT17  002000          411#
ERR     002700          592     618#    622
ERRCNT  001232          163#    747     774    1087*   1305*
ERRFLG  001325          202#    481*    733*    795*   1037*   1050    1064*   1123*
ERRMSG  005172         1047*   1065    1068#
ERRPC   002770          624     640#
ERTAB0  005322         1062    1097#
EXIT  = 000205           96#
EXITER  005252         1092    1087#
FLAG    021314         2431*   2454    2489    2494*   2497    2599    2601*   2732#   2882
FLOAT   002536          585#    591
FY      002566          594#    605     609     615
HALTS   005222         1033    1079#
HILIM   004366          876*    903     921#
ICOUNT  001222          159#    793     798*
IISR    017132         2410    2534#
INBUF   007502          846     882    1202#
INCHAR  010020         1231    1259#
INIFLG  001324          201#    507     527     534*
INSTER= 104404          223#    897
INSTR = 104403          221#   1329    1381    1394    1403    1482    1491
INSTR2  004166          853     865#
INTTY   012266         1414    1431    1441    1454    1470    1655#
KMCM    007330          637    1187#
LIMITS  004314          892     903#
LINE    007016         1187#   1483
LOBITS  004372          878*    907     923#    924
LOCK    001220          158#    797*    814     816    1056
LOKFLG  001326          203#
LOLIM   004364          875*    905     920#
LPCNI   001224          160#    792*    793     796*
LSTERR  001234          164#    490*    732*   1034    1036*   1124*
MASKX   001244          172#
MASTEK  006142         1058    1187#
MCRLF   005672          831     954    1054    1055    1063    1187#   1328    1390
```

```
DZDMH   MACY11 30(1046)  11-JUL-77  12:32  PAGE 65                                                           PAGE: 0079
DZDMH.P11     16-MAY-77 09:54             CROSS REFERENCE TABLE -- USER SYMBOLS

MCSPX   006072             737     1187#
MDATA   007544             984      994     1204#
MEMLIM  001304             188#     700#
MEPASS  005733             736     1187#
MERRPC  006217            1061     1187#
MERRX   006117             743     1187#
MERR2   005760            1187#    1590
MERR3   006005             673     1187#
MILK    001322             196#     484*     745     1288*    1293*    1297
MLOCK   006043             714     1187#
MNEW    006144             668     1187#
MODU    006704            1187#    1453
MPASSX  006106             741     1187#
MPFAIL  005675            1121     1187#
MOM     005666             861     1187#    1352     1427     1437     1447     1462     1476
MR      005755             723     1187#    1346
MRESET= 004000              96#
MSTCLR= 104412             235#    1126     1948     2014     2077     2143     2203     2263     2304     2395
MTITLE  001000             136#     511
MTSTN   006130            1059     1187#    1330
MTSTPC  006031            1187#
MVECX   006100             739     1187#
NEXT    001216             157#     799     1092     1683*    1866*    1946*    2012*    2075*    2141*    2201*    2261*    2302*    2393*
NOACT   007154             523     1187#    1282
NODEV   002674             577      616#
NUM     006450            1187#    1382
OISP    017422            2412     2597#
OK      002646             603      610#     639
ONE     001302             187#
PACT00  001702             365#
PACT01  001706             368#
PACT02  001712             371#
PACT03  001716             374#
PACT04  001722             377#
PACT05  001726             380#
PACT06  001732             383#
PACT07  001736             386#
PACT10  001742             389#
PACT11  001746             392#
PACT12  001752             395#
PACT13  001756             398#
PACT14  001762             401#
PACT15  001766             404#
PACT16  001772             407#
PACT17  001776             410#
PARAM = 104405             225#    1331     1383     1396     1405     1484     1493
PARAM1  004234             881#     898
PARBIT= 040000              96#
PARERR  004310             884      886      888      897#     904      906      908
PASCNT  001230             162#     734*     735      746      771     1304*
PERFOR= 004537              96#
PFTAB   005430            1122     1128#
POPR0 = 012600              72#    1096
POP1SP= 005726              70#
POP2SP= 022626              74#     801
PRIO    006547            1187#    1413


DZDMH   MACY11 30(1046)  11-JUL-77  12:32  PAGE 66                                                           PAGE: 0080
DZDMH.P11     16-MAY-77 09:54             CROSS REFERENCE TABLE -- USER SYMBOLS

PS    = 177776              63#     476*     711*    1617*    1627*    2403*    2461*
PUSHR0= 010046              71#    1083
PUSH1S= 005746              69#
PUSH2S= 024646              73#
QV.FLG  001327             204#     482*     750*     790
RBUF    021372            1694     1750     1811     1819     1958     2273     2319     2339     2744#
RBUFF   020356            2419     2717#
RBUFFE  021312            2440     2725#
RBUFF1  020356            2718#
RBUFF2  020462            2719#
RBUFF3  020566            2720#
RBUFF4  020672            2721#
RBUFF5  020776            2722#
RBUFF6  021102            2723#
RBUFF7  021206            2724#
RCNTAB  020230            2427     2429     2567     2574     2711#
RCOUNT  021370            1692     1751     1814     1817     2743#
RDNTAB  020266            2438     2472     2486     2668*    2670*    2714#
RECBA   020062            2566     2655     2679#
RESREG  005220            1075     1078#
RESTAP  005350            1108     1114#
RESTRT  003534             749      753      761#
RESUM   022276            2317     2813#
RESUME  020060            2312*    2313     2349     2351*    2437*    2536     2544*    2548     2615*    2678#
RES05 = 104407             229#    1078
RETURN  001214             156#     492*     720*     724      761*     799*     803     1092*    1095     1127     1345*    1355*    1357
RFLAG   021320            1699*    1804     1807*    1832     2734#
RFRELD  022416            1957     2086     2272     2318     2843#
ROMCLK= 104414             239#    1134     1137     1174     1179
RUN     001316             193#     485*    1286*    1287*    1294
SAVACT  001312             191#     671     1588*
SAVNUM  001314             192#     479*     748*     751*    1581*
SAVPC   001276             195#     622*     642      929*    1099
SAVR0   001260             178#     938*     943     2882
SAVR1   001262             179#     628*     937*     944
SAVR2   001264             180#     936*     945     2882
SAVR3   001266             181#     935*     946
SAVR4   001270             182#     934*     947     2882
SAVR5   001272             183#     933*     948     2882
SAVSP   001274             184#
SAV05 = 104406             227#    1038
SCAN    016604            2462#    2525     2528
SCAN1   017102            2464     2467     2470     2523#
SCAN2   016616            2463     2465#
SCOPE = 104400             215#    1854     1931     2000     2063     2129     2189     2249     2286     2377     2516
SCOP1 = 104401             217#
SHUTDO  022502            1836     1905     1982     2045     2111     2348     2870#
SKIP    002632             573      576      597      600      606#
SOFTSW  010052            1229     1268#
SPACNT= 004713             964*     988      991*    1005*
SPEED   007340            1187#    1440
STACK = 001200              64#     477      690     1093     1116
STAT    001240             170#
STAT1   001365             251#    1301*    1685     1690     1701     1868     1873     1906     1949     1954     2015     2020     2078
                          2083     2144     2149     2204     2209     2264     2269     2305     2310     2396     2401     2404     2433
                          2451     2610     2619     2758     2789     2818
```

```
STAT2   001372             252#  1302*
STAT3   001372             253#  1303*   1687   1870   1909   1951   2017   2080   2146   2206   2266   2307   2355
                          2398   2500
STRTSW  001236             169#   513*   516*   517    519    529    531    666    712    721   1323   1347*  1377
                          1601
SV05    004402             933#
SWFLG   010016             480*   825   1224*  1251*  1257#
SWMES   007205            1187#  1227
SWMES1  007215            1187#  1230
SWR     001202             143#   497*   499    503*   513    671    676    781    788    812    827   1027   1032
                          1081   1088   1090   1152   1212   1250*
SWREG   000176             129#   503   1212   1270
SW00  = 000001              45#   517   1377   1601
SW01  = 000002              44#   721   1323   1347
SW02  = 000004              43#
SW03  = 000010              42#   666
SW04  = 000020              41#
SW05  = 000040              40#
SW06  = 000100              39#  1152
SW07  = 000200              38#
SW08  = 000400              37#  1088
SW09  = 001000              36#   812
SW10  = 002000              35#  1090
SW11  = 004000              34#   788
SW12  = 010000              33#   827   1027
SW13  = 020000              32#  1032
SW14  = 040000              31#
SW15  = 100000              30#
TBUF    021324            1766   1796   1818   1877   1961   2090   2276   2322   2336   2736#
TBUFF   020120            2424   2631   2633   2635   2682#
TCOUNT  021322            1767   1799   2735#
TEMP    001416             271#   971   1118*  1119*  1160*  1165*  1171*  1183*  1706*  1709*  1714*  1717*  1723*
                          1726*  1744*  1747*  1753*  1756*  1760*  1763*  1769*  1772*  1775*  1779*  1881*  1884*
                          1902*  1933*  1964*  1967*  2027*  2030*  2093*  2096*  2156*  2159*  2216*  2219*  2279*
                          2282*  2326*  2329*  2459*  2523*
TEMP1   001246             173#   537*  1189   1613*  1614*  1776*  1781*
TEMP2   001250             174#   538*  1191   1784*  1848*  1925*  1994*  2057*  2123*  2183*  2243*  2371*  2460*
                          2517*  2526*  2882
TEMP3   001252             175#   540*   566*   618    627*  1193   1386   1389   1501*  1730*  1820*  1849*  1926*
                          1995*  2058*  2124*  2184*  2244*  2372*  2476*  2518*  2624*  2636*  2882
TEMP4   001254             176#   541*  1195   1399   1402   1408   1411   1487   1490   1496   1499   1731*  1850*
                          1927*  1996*  2059*  2125*  2185*  2245*  2373*  2519*  2625*  2882
TEMP5   001256             177#   542*  1197   1380*  1393*  1599   1851*  1928*  1997*  2060*  2126*  2186*  2246*
                          2374*  2520*  2882
TFLAG   021316            1698*  1789   1792*  1834   2417*  2428   2443   2491*  2493*  2733#
TIMER = 104416             243#
TKCSR   001204             148#   848   1214   1259   1655
TKDBR   001206             149#   850    856   1217   1219   1261   1657
TLAST = 016214            1350   2747#
TPCSR   001210             150#   832    854   1029   1262   1658
TPDBR   001212             151#   834*   856*  1031*  1264*  1660*
TRPOK   004730            1017#
TSTNO   001226             161#   491*  1102   1130   1334   1341   1343   1682*  1865*  1945*  2011*  2074*  2140*
                          2200*  2260*  2301*  2392*
TST1    012320            1337   1355   1682#
TST10   015476            2201   2260#
```

```
TST11   015632            2261   2301#
TST12   016214            2302   2392#  2747
TST2    013404            1683   1865#
TST3    013754            1866   1945#
TST4    014236            1946   2011#
TST5    014510            2012   2074#
TST6    014772            2075   2140#
TST7    015234            2141   2200#
TTST    003612             715*   716*   718*   719*   782#
TWOSYN= 010000              96#
TYPDAT  005206            1053   1071   1074#
TYPE  = 104402             219#   511    523    535    620    625    633    636    668    673    714    723    736
                           737    739    741    743    831    844    861    954    994   1054   1055   1058   1059
                          1061   1063   1067   1072   1121   1227   1230   1282   1328   1346   1352   1390   1412
                          1426   1429   1436   1439   1446   1452   1461   1468   1475   1590
TYPMSG  005106            1051   1054#
VEC     006526            1187#  1404
VECMAP  012010            1589   1601#
WHICH   012002            1392   1597#
WRDCNT  004710             962*   995*  1003#
WRKO.F  005174            1066   1069#
XBX     005000            1028   1030   1032#
XCNTAB  020250            2442   2444   2639   2662   2712#
XCSR    003546             738    763#
XDNTAB  020322            2645*  2647*  2715#
XERR    003570             744    772#
XFRELD  022450            1876   1960   2023   2089   2275   2321   2856#
XHEAD   006224             535   1187#
XLOC    003022             593*   611*   614    645    658#
XMITBA  020062            2418   2422   2425   2573   2680#
XPASS   003562             742    769#
XSTATQ  007454             544   1187#
XTSTN   005330            1060   1100#
XVEC    003554             740    766#
X0    = 000110            2683#  2686#  2689#  2692#  2695#  2698#  2701#  2704#  2707#  2710#
X1    = 000101            2683#  2686#  2689#  2692#  2695#  2698#  2701#  2704#  2707#
X2    = 000102            2683#  2686#  2689#  2692#  2695#  2698#  2701#  2704#  2707#
X3    = 000103            2683#  2686#  2689#  2692#  2695#  2698#  2701#  2704#  2707#
X4    = 000104            2683#  2686#  2689#  2692#  2695#  2698#  2701#  2704#  2707#
X5    = 000105            2683#  2686#  2689#  2692#  2695#  2698#  2701#  2704#  2707#
X6    = 000106            2683#  2686#  2689#  2692#  2695#  2698#  2701#  2704#  2707#
X7    = 000107            2683#  2686#  2689#  2692#  2695#  2698#  2701#  2704#  2707#
ZERO    001300             186#
$COD  = ****** U             1
$CRAP = 177777               1#  1664#  1667   1678#  1855#  1858   1861#  1935#  1938   1941#  2001#  2004   2007#
                          2064#  2067   2070#  2130#  2133   2136#  2190#  2193   2196#  2250#  2253   2256#  2289#
                          2292   2297#  2378#  2381   2388#
$ENDAD  003522             123    509    755#  1079
$N    = 000012               1#  1664   1678   1680   1685#  1855   1861   1863   1868#  1935   1941   1943   1948
                          1949#  2001   2007   2009   2014   2015#  2064   2070   2072   2077   2078#  2130   2136
                          2138   2143   2144#  2190   2196   2198   2203   2204#  2250   2256   2258   2263   2264#
                          2289   2297   2299   2304   2305#  2378   2388   2390   2395   2396#  2747#
$S    = 000014               1#  1683   1685#  1866   1868#  1946   1949#  2012   2015#  2075   2078#  2141   2144#
                          2201   2204#  2261   2264#  2302   2305#  2396#
$Y    = 000017               1#   207#   215    217#   219#   221#   223#   225#   227#   229#   231#   233#   235#
                           237#   239#   241#   243#   245#
```

```
DZDMH   MACY11 30(1046)  11-JUL-77  12:32  PAGE 69                                                         PAGE: 0083
DZDMH.P11     16-MAY-77 09:54             CROSS REFERENCE TABLE -- USER SYMBOLS

.       = 023554              108#    109     112#    119#    124#    127#    131#    135#    137#    189#    190#    191#    192#
                              272#    277#    279#    280#    281#    282#    284#    285#    286#    287#    289#    290#    291#
                              292#    294#    295#    296#    297#    299#    300#    301#    302#    304#    305#    306#    307#
                              309#    310#    311#    312#    314#    315#    316#    317#    319#    320#    321#    322#    324#
                              325#    326#    327#    329#    330#    331#    332#    334#    335#    336#    337#    339#    340#
                              341#    342#    344#    345#    346#    347#    349#    350#    351#    352#    354#    355#    356#
                              357#    515     525     657#    675     1110    1120    1168#   1203#   1205#   1260    1263    1284
                              1378    1422    1542    1546    1593    1624    1656    1659    1686    1688    1691    1702    1705
                              1708    1716    1746    1755    1762    1771    1785    1790    1794    1797    1800    1805    1809
                              1812    1815    1824    1829    1869    1871    1874    1883    1950    1952    1955    1966    2016
                              2018    2021    2029    2043    2079    2081    2084    2095    2109    2145    2147    2150    2154
                              2170    2205    2207    2210    2214    2218    2230    2265    2267    2270    2306    2308    2311
                              2328    2397    2399    2402    2434    2449    2452    2479    2547    2579    2611    2614    2617
                              2620    2680#   2711#   2712#   2714#   2715#   2718#   2719#   2720#   2721#   2722#   2723#   2724#
                              2744#   2746#   2759    2762    2789    2792    2819    2822
.BEGIN  003152                690#
.CNVRT  004472                234     955#
.CONVR  004466                232     954#
.DATAC  005552                242     1159#
.DELAY  005436                238     1132#
.EOP    003364                731#    2393
.ERRTA  023414                1046    2882#
.HLT    004750                115     1026#
.INSTE  004154                224     861#
.INSTR  004050                222     840#
.INSTI  004070                844#    864
.MSG    004072                842#    845#
.MSTCL  005466                236     1143#
.PARAM  004174                226     872#
.PFAIL  005336                113     478     1107#   1115
.RES05  004434                230     943#
.POMCL  005504                240     1148#
.SAV05  004374                228     929#
.SCOPE  003576                216     779#
.SCOP1  003736                218     811#
.START  002002                132     476#    492     1247
.TIMER  005616                244     1170#
.TRPSP  004716                117     1014#
.TRPTA  001333                214#    1019
.TYPE   003766                220     822#
```

```
DZDMH   MACY11 30(1046)  11-JUL-77  12:32  PAGE 71                                                         PAGE: 0084
DZDMH.P11     16-MAY-77 09:54             CROSS REFERENCE TABLE -- MACRO NAMES

DMEND       1#    725
DMFPNT      1#
HLT        75#    1711    1719    1732    1734    1749    1758    1765    1774    1783    1786    1791    1795    1798    1801
          1806    1810    1813    1816    1825    1830    1852    1888    1892    1900    1929    1971    1975    1981    1998
          2034    2038    2044    2061    2100    2104    2110    2127    2161    2165    2171    2187    2221    2225    2231
          2247    2287    2333    2345    2375    2480    2521    2529    2626    2637    2644    2660    2667
$AUTO       1#     547
$BASEC      1#    1837    1908    1983    2046    2112    2172    2232    2353
$BUFFE      1#    1199
$BYTE       1#    2683    2686    2689    2692    2695    2698    2701    2704    2707
$CKDAT      1#    2472
$COMP       1#
$CYCLE      1#    1271
$DATAF      1#    1664
$EOP        1#     725
$EXER       1#    2378
$FD         1#    1736    2774    2834
$FINI       1#    2747
$GETPA      1#
$HALF       1#    2250
$HD         1#    2804
$HEADE      1#
$LSTDA      1#    1935
$MARHI      1#
$MOCK       1#
$MSG        1#    1187
$NONEX      1#    2001    2064
$ORUN       1#    1855
$PFAIL      1#    1103
$PROC       1#    2130
$PROC1      1#    2190
$QUEST      1#    1381    1394    1403    1482    1491
$RAMCL      1#    1131
$RCLK       1#    1134    1137    1174    1179
$RESUM      1#    2289
$SCOPE      1#     775
$SETUP      1#    1868    1949    2015    2078
$SIMBC      1#
$SKIPT      1#    1685    1868    1949    2015    2078    2144    2204    2264    2305    2396
$SOFTC      1#    1207
$TRPDE      1#     215     217     219     221     223     225     227     229     231     233     235     237     239     241
           243
$TSTN       1#    1680    1863    1943    2009    2072    2138    2198    2258    2299    2390
$VARIA      1#     131
$XZ         1#    1664    1678    1855    1861    1935    1941    2001    2007    2064    2070    2130    2136    2190    2196
          2250    2256    2289    2297    2378    2388

. ABS.  023554      000

 ERRORS DETECTED:  0

 DZDMH,DZDMH/SOL/CRF_IPLUTL,DZDMH
 RUN-TIME: 8 12 1 SECONDS
 RUN-TIME RATIO: 199/21=9.1
 CORE USED:  24K  (47 PAGES)
```