DMC11

DDCMP MODE LINE UNIT

MD-11-DZDME-B

EP-DZDME-B-DL-A

FICHE 1 OF 1

AUG 1977

digital

MADE IN USA

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZDME-B-D
PRODUCT NAME:	DDCMP MODE LINE UNIT TESTS
DATE:	MAY 1977
MAINTAINER:	DIAGNOSTICS
AUTHOR:	FAY BASHAW

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.

DZDME tests the DMC-11 Line Unit (M8201 or M8202). It performs write/read tests on the DMC Line Unit registers. it checks for proper transmitter, receiver, and BCC operation in DDCMP mode. The modem signals are also checked. DZDME requires a DMC Micro-Processor (M8200 or M8204) to run. For best diagnosis a turn-around connector should be installed, however the diagnostic will run without it (some tests are skipped).

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] Jump and Crom 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 STORAGE

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
M8202 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 test
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 REGISTER 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 08 Goto beginning of the test(on error).
5. SW 10 Goto next test(on error).

SCOPE SWITCHES

1. SW06 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. SW09 (if enabled by 'SCOP1') on an error; If an '*' is printed in front of the test no. (ex. *TEST NO. 10) SW09 is incorporated in that test and therefore SW09 is usually the best switch for the scope loop (SW14=0, SW10=0, SW09=1, SW08=0). If SW09 is not enabeled; and there is a HARD error (constant); SW08 is best. (SW14=1,0, SW10=0, SW09=0, SW08=1). for intermittemt errors; SW14=1 will loop on test reguardless of error or not error. (SW14=1, SW10=0, SW09=0, SW08=1,0)
3. SW11 Inhibit interations.
4. SW14 Loop on current test.

4.2 STARTING ADDRESS

Starting address is at 000200 there are no other starting addresses for the DMC11 diagnostics. (See Section 4.0)

NOTE: If address 000042 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 4.0 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 (SW00=1) or by autosizing (SW00=0 and SW07=0). 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 SW07=1 on start up.

7.3 HARDWARE CONFIGURATION RESTRICTIONS

DMC11(M8200)- Jumper W1 must be in, and switch 7 of E76 must be in the OFF position.

KMC(M8204)- Jumper W1 must be in.

LINE UNIT(M8201)- 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 (M8202)- 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=0) 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 DZDMC CSR:  175000 VEC:  0300 PASSES:  000001
ERRORS:  000000
```

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 (1402)	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, 1500, 1502, 1504, and 1506. The second DMC status is located at 1510, 1512, 1514, and 1516. The information contained in each 4 word entry is defined as follows:

```
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 UNIT 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 RUN FREE RUNNING TESTS ON KMC11
        BIT1=0 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 626 or 16520 a DMC11 has been found, if not, the address is updated by 10 and the search continues. A -1 indicates a DMC11 with no CROM or CRAM, 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.

8.5 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
***************
  DZDME  LST
***************
```

```
6       MAINDEC-11-DZDME-B   DMC11 DDCMP LINE UNIT TESTS
        COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
        ----------------------------------------------------------------

1667    ************************** TEST 1 **************************
        OUT CONTROL REGISTER READ/ONLY TEST
        DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
        BITS ARE IN THE CORRECT STATE

1691    ************************** TEST 2 **************************
        IN CONTROL REGISTER READ/ONLY TEST
        DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
        BITS ARE IN THE CORRECT STATE

1714    ************************** TEST 3 **************************
        MODEM CONTROL REGISTER READ/ONLY TEST
        DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
        BITS ARE IN THE CORRECT STATE

1738    ************************** TEST 4 **************************
        MAINTENANCE REGISTER READ/ONLY TEST
        DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
        BITS ARE IN THE CORRECT STATE

1769    ************************** TEST 5 **************************
        LINE UNIT REGISTER WRITE/READ TEST
        SET BITS IN LU REGISTER 12, VERIFY IT IS SET
        CLEAR BITS IN LU REGISTER 12, VERIFY IT IS CLEAR

1811    ************************** TEST 6 **************************
        LINE UNIT REGISTER WRITE/READ TEST
        SET BIT1 IN LU REGISTER 17, VERIFY IT IS SET
        CLEAR BIT1 IN LU REGISTER 17, VERIFY IT IS CLEAR

1853    ************************** TEST 7 **************************
        LINE UNIT REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH LINE UNIT REGISTER 13
        FLOAT A 0 THROUGH LINE UNIT REGISTER 13

1911    ************************** TEST 10 **************************
        LINE UNIT REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH LINE UNIT REGISTER 14
        FLOAT A 0 THROUGH LINE UNIT REGISTER 14

1963    ************************** TEST 11 **************************
        SWITCH PAC TEST
        THIS TEST READS SWITCH PAC#1
        THIS SWITCH PAC CONTAINS THE DDCMP LINE #
```

```
1985    **************************** TEST 12 ****************************
        SWITCH PAC TEST
        THIS TEST READS SWITCH PAC#2
        THIS SWITCH PAC CONTAINS THE BM873 BOOT ADD

2007    **************************** TEST 13 ****************************
        LINE UNIT CLOCK TEST
        THIS TEST VERIFYS THAT THE LU INTERNAL CLOCK
        (BIT 1 IN LU-17) IS WORKING

2040    **************************** TEST 14 ****************************
        OUT DATA SILO TEST
        SET SOM AND LOAD OUT DATA SILO
        VERIFY THAT OCOR SET, INDICATING THAT THE
        CHARACTER IS AT THE BOTTOM OF THE OUT SILO

2073    **************************** TEST 15 ****************************
        DDCMP TEST OF RTS AND OUT ACTIVE
        SET SOM AND LOAD OUT DATA SILO
        SINGLE STEP 2 DATA CLOCKS, VERIFY
        THAT RTS AND ACTIVE ARE SET

2117    **************************** TEST 16 ****************************
        TEST OF OUT CLEAR
        SET SOM AND LOAD OUT DATA SILO
        SINGLE STEP DATA CLOCK, SET OUT CLEAR
        VERIFY THAT OCOR,RTS, AND ACTIVE ARE CLEARED

2174    **************************** TEST 17 ****************************
        DDCMP TRANSMITTER TEST
        SINGLE CLOCK THE CHARACTER 0
        VERIFY EACH BIT POSITION AS IT
        PASSES THE BIT WINDOW (SI BIT)
        ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE

2225    **************************** TEST 20 ****************************
        DDCMP TRANSMITTER TEST
        SINGLE CLOCK THE CHARACTER 125
        VERIFY EACH BIT POSITION AS IT
        PASSES THE BIT WINDOW (SI BIT)
        ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE

2276    **************************** TEST 21 ****************************
        DDCMP TRANSMITTER TEST
        SINGLE CLOCK THE CHARACTER 252

2279    VERIFY EACH BIT POSITION AS IT
        PASSES THE BIT WINDOW (SI BIT)
        ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
```

```
2327     *************************** TEST 22 ***************************
         DDCMP TRANSMITTER TEST
         SINGLE CLOCK THE CHARACTER 377
         VERIFY EACH BIT POSITION AS IT
         PASSES THE BIT WINDOW (SI BIT)
         ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE

2378     *************************** TEST 23 ***************************
         DDCMP TRANSMITTER TEST
         SINGLE CLOCK A BINARY COUNT PATTERN
         VERIFY EACH BIT POSITION AS IT
         PASSES THE BIT WINDOW (SI BIT)
         ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
         AND R5 CONTAINS THE CHARACTER THAT FAILED

2439     *************************** TEST 24 ***************************
         DDCMP STRIP SYNC TEST
         SET LU LOOP, SINGLE STEP 5 SYNCS,
         VERIFY THAT IN ACTIVE DOES NOT SET

2467     *************************** TEST 25 ***************************
         DDCMP IN ACTIVE TEST
         SET LU LOOP, SINGLE STEP 5 SYNCS AND A NON-SYNC (301)
         VERIFY THAT IN ACTIVE IS SET

2495     *************************** TEST 26 ***************************
         DDCMP IN ACTIVE TEST
         SET LU LOOP, SINGLE STEP 1 SYNC AND A NON-SYNC (301)
         VERIFY THAT IN ACTIVE DOES NOT SET

2523     *************************** TEST 27 ***************************
         DDCMP IN ACTIVE TEST
         SET LU LOOP, SINGLE STEP 2 SYNCS AND A NON-SYNC (301)
         VERIFY THAT IN ACTIVE IS SET

2551     *************************** TEST 30 ***************************
         IN CLEAR TEST
         SYNC UP RECEIVER AND TRANSMIT A CHARACTER
         WAIT FOR IN RDY, THEN SET IN CLEAR
         VERIFY THAT IN ACTIVE AND IN RDY ARE CLEARED

2600     *************************** TEST 31 ***************************
         DDCMP BASIC RECEICER TEST
         SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 0
         VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED

2637     *************************** TEST 32 ***************************
         DDCMP BASIC RECEICER TEST
         SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 125
         VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED
```

```
2674    *************************** TEST 33 ***************************
        DDCMP BASIC RECEICER TEST
        SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 252
        VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED

2711    *************************** TEST 34 ***************************
        DDCMP BASIC RECEICER TEST
        SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 377
        VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED

2748    *************************** TEST 35 ***************************
        DDCMP DATA TEST
        THIS TEST SINGLE STEPS A BINARY COUNT PATTERN
        CHECKING EACH CHARACTER AS IT IS RECEIVED

2787    *************************** TEST 36 ***************************
        DDCMP DATA TEST
        THIS TEST SINGLE STEPS A BINARY COUNT PATTERN
        CHECKING EACH CHARACTER AS IT IS RECEIVED
        THIS TEST IS EXACTLY THE SAME AS THE LAST TEST,
        EXCEPT LINE UNIT LOOP IS SET IN LU REGISTER 12

2831    *************************** TEST 37 ***************************
        TRANSMITTER MARK TEST
        SINGLE CLOCK 3 SYNCS AND A 301 AND 20 EXTRA
        CLOCK TICKS, VERIFY THAT A 301, A 377 AND A 377
        WERE RECEIVED INDICATING THAT THE TRANSMITTER WENT
        TO A MARK STATE FOR 16 BITS WHEN OUT SILO WAS EMPTY

2876    *************************** TEST 40 ***************************
        CABLE TURNAROUND TEST
        CLEAR LINE UNIT LOOP, SET DTR
        VERIFY THAT MODEM READY IS SET
        CLEAR DTR, VERIFY THAT MRDY IS CLEARED

2924    *************************** TEST 41 ***************************
        CABLE TURNAROUND TEST
        CLEAR LINE UNIT LOOP, LOAD OUT DATA SILO
        VERIFY THAT ALL MODEM SIGNALS ARE SET

2967    *************************** TEST 42 ***************************
        TEST OF CRC OPERATION
        USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
        0, VERIFY THE LSB OF THE BCC ON EACH SHIFT
        TEST TRANSMITTER FIRST THEN THE RECEIVER BCC

3042    *************************** TEST 43 ***************************
        TEST OF CRC OPERATION
        USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
        377, VERIFY THE LSB OF THE BCC ON EACH SHIFT
        TEST TRANSMITTER FIRST THEN THE RECEIVER BCC
```

```
3117   **************************** TEST 44 ****************************
       TEST OF CRC OPERATION

3119   USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
       125, VERIFY THE LSB OF THE BCC ON EACH SHIFT
       TEST TRANSMITTER FIRST THEN THE RECEIVER BCC

3192   **************************** TEST 45 ****************************
       TEST OF CRC OPERATION
       USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
       252, VERIFY THE LSB OF THE BCC ON EACH SHIFT
       TEST TRANSMITTER FIRST THEN THE RECEIVER BCC

3267   **************************** TEST 46 ****************************
       TRANSMITTER CRC TEST
       USING THE CRC16 POLYNOMINAL, SINGLE CLOCK A BINARY
       COUNT PATTERN, VERIFY THE LSB OF THE TRANSMITTER BCC ON EACH SHIFT

3333   **************************** TEST 47 ****************************
       RECEIVER CRC TEST
       USING THE CRC16 POLYNOMINAL, SINGLE CLOCK A BINARY
       COUNT PATTERN, VERIFY THE LSB OF THE RECEIVER BCC ON EACH SHIFT

3399   **************************** TEST 50 ****************************
       TRANSMITTER DDCMP CRC TEST
       THIS TEST TRANSMITS A FOUR CHARACTER MESSAGE WITH CRC
       BOTH DATA AND THE BCC ARE VERIFIED IN THE BIT
       WINDOW. THE FOUR CHARACTERS ARE 0,125,252,377
       THE TRANSMITTER IS CHECKED FOR GOING TO A MARK STATE AFTER THE BCC

3500   **************************** TEST 51 ****************************
       RECEIVER DDCMP CRC TEST
       THIS TEST CLOCKS A FOUR CHARACTER MESSAGE WITH BCC
       AND VERIFYS CORRECT DATA RECEPTION AND BCC MATCH
       THE FOUR CHARACTER MESSAGE IS 0,125,252,377

3557   **************************** TEST 52 ****************************
       DDCMP EOM FUNCTION TEST
       THIS TEST LOADS OUT SILO WITH: 2 SYNCS,4 CHAR MESSAGE,EOM
       4 CHARACTER MESS,EOM. THE DATA STREAM IS CHECKED TO BE
       4 CHAR,BCC,4 CHAR,BCC,MARKS. THIS TEST VERIFYS THAT
       THE CHARCTERS LOADED WITH EOM SET ARE LOST
       ALL DATA AND BCC'S ARE CHECKED IN THE BIT WINDOW
       THE FOUR CHARACTER MESSAGE IS 0,125,252,377
       RECEIVED DATA IS VERIFIED, AND IN BCC MATCH IS CHECKED

3803   **************************** TEST 53 ****************************
       DDCMP EOM FUNCTION TEST
       THIS TEST LOADS OUT SILO WITH: 2 SYNCS,4 CHAR MESSAGE,EOM
       SOM,4 CHAR MESS,EOM. THE DATA STREAM IS CHECKED TO BE
       4 CHAR,BCC,4 CHAR,BCC,MARKS. THIS TEST VERIFYS THAT
       THE CHARCTERS LOADED WITH EOM SET ARE LOST
       ALSO THAT THE CHAR LOADED WITH SOM IS NOT IN THE BCC
       ALL DATA AND BCC'S ARE CHECKED IN THE BIT WINDOW
```

```
          THE FOUR CHARACTER MESSAGE IS 0,125,252,377
          RECEIVED DATA IS VERIFIED, AND IN BCC MATCH IS CHECKED

4081      *************************** TEST 54 ***************************
          EMPTY SILO TEST
          LOAD SILO WITH 2 SYNCS, 4 CHAR MESSAGE, SINGLE CLOCK
          UNTIL THE SILO IS EMPTY, LOAD 4 MORE CHARACTERS IN THE
          SILO. GIVE MORE TICKS, AND VERIFY THAT ONLY THE FIRST
          4 CHARACTER MESSAGE WAS RECEIVED AND THAT RTS IS CLEAR

4146      *************************** TEST 55 ***************************
          HALF DUPLEX TEST
          SET LINE UNIT LOOP AND HALF DUPLEX, SEND SYNCS AND A
          MESSAGE. VERIFY THAT IN-ACTIVE AND IN-READY ARE CLEAR

4181      *************************** TEST 56 ***************************
          DDCMP CABLE DATA TEST

4183      THIS TEST LOADS OUT SILO WITH THE FOLLOWING:
          4 SYNCS,16 CHAR,EOM,16 CHAR,EOM,16 CHAR,EOM
          THE 16 CHARACTERS INCLUDE A FLOATING ONE AND ZERO
          THE DATA IS TRANSMITTED OVER THE CABLE USING THE INTERNAL CLOCK
          RECEIVED DATA IS VERIFIED AS IS IN BCC MATCH
          LOOP-BACK CONNECTOR MUST BE ON TO RUN THIS TEST

4276      *************************** TEST 57 ***************************
          DDCMP CABLE DATA TEST
          THIS TEST LOADS OUT SILO WITH THE FOLLOWING:
          4 SYNCS,59 DATA CHARACTERS,EOM WITH GARBAGE CHARACTER
          THE DATA IS TRANSMITTED OVER THE CABLE USING THE INTERNAL CLOCK
          RECEIVED DATA IS VERIFIED AS IS IN BCC MATCH
          LOOP-BACK CONNECTOR MUST BE ON TO RUN THIS TEST
```

```
                                  ;*MAINDEC-11-DZDME-B   DMC11 DDCMP LINE UNIT 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-DZDME-B   DMC11 DDCMP LINE UNIT 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 ITERATIONS
        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


```

```


                                        ;:******************************************************************
                                        ;-------------------------------------------------------------------
                                                ;TRAPCATCAER 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  005377  040515  047111          MTITLE: .ASCII  <377><12>/MAINDEC-11-DZDME-B/<377>
001025     104  041515  030461                  .ASCIZ  /DMC11 DDCMP LINE UNIT 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                  DMP04:  0               ;POINTER TO DMC11 PORT REGISTER(SEL 4)
001414  000000                  DMP06:  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

```

```
413

                                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     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 UNIT 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  005020                            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                                  XSTATQ                          ;
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
```

```
002760  104402                          3$:     TYPE
002762  007330                                  KMCM
002764  022626                          4$:     CMP     (SP)+,(SP)+         ;ADJUST STACK
002766  000727                                  BR      OK                  ;BR TO GET OUT
002770  000001                          ERRPC:  1
002772     006     002                          .BYTE   6,2
002774  001276                                  SAVPC
002776  000002                          CONTAB: 2
003000     006     004                          .BYTE   6,4
003002  003022                                  XLOC
003004     006     002                          .BYTE   6,2
003006  001404                                  DMCSR
003010     007                          DEVTAB: .BYTE   7                   ;DJ
003011     017                                  .BYTE   17                  ;DH
003012     007                                  .BYTE   7                   ;DQ
003013     007                                  .BYTE   7                   ;DU
003014     007                                  .BYTE   7                   ;DUP
003015     007                                  .BYTE   7                   ;LK
003016     007                                  .BYTE   7                   ;DMC
003017     007                                  .BYTE   7                   ;DZ
003020     007                                  .BYTE   7                   ;KMC
        003022                          .EVEN
003022  000000                          XLOC:   0
003024  005705                          AUDONE: TST     R5                  ;DMC?
003026  001005                                  BNE     1$                  ;BR IF KMC AND ALL DONE
003030  012705  177777                          MOV     #-1,R5              ;SET R5 TO -1 (KMC)
003034  012700  001500                          MOV     #DM.MAP,R0          ;RESET R0 TO START OF TABLE
003040  000602                                  BR      AUSTRT              ;GO DO KMC'S
003042  012637  000006                  1$:     MOV     (SP)+,@#6           ;RESTORE LOC 6
003046  012637  000004                          MOV     (SP)+,@#4           ;RESTORE LOC 4
003052  032737  000010  001236                  BIT     #SW03,STRTSW        ;SELECT SPECIFIC DEVICES??
003060  001422                                  BEQ     3$                  ;BR IF NO.
003062  104402  006144                          TYPE    ,MNEW               ;TYPE THE MESSAGE.
003066  005000                                  CLR     R0                  ;ZERO DATA LIGHTS
003070  000000                                  HALT                        ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
003072  027737  176104  001312                  CMP     @SWR,SAVACT         ;IS THE NUMBER VALID?
003100  101404                                  BLOS    2$                  ;BR IF NUMBER IS OK.
003102  104402  006005                          TYPE    ,MERR3              ;TELL USER OF INVALID NUMBER.
003106  000000                                  HALT                        ;STOP EVERY THING.
003110  000776                                  BR      .-2                 ;RESTART THE PROGRAM AGAIN.
003112  017737  176064  001306          2$:     MOV     @SWR,DMACTV         ;GET NEW DEVICE PATTERN
003120  013700  001306                          MOV     DMACTV,R0           ;SHOW THE USER WHAT HE SELECTED.
003124  000000                                  HALT                        ;CONTINUE DYNAMIC SWITCHES.
003126  012700  000300                  3$:     MOV     #300,R0             ;PREPARE TO CLEAR THE FLOATING
003132  012701  000302                          MOV     #302,R1             ;VECTOR AREA. 300-776
003136  010120                          4$:     MOV     R1,(R0)+            ;START PUTTING "PC+2 - HALT"
003140  005021                                  CLR     (R1)+               ;IN VECTOR AREA.
003142  022021                                  CMP     (R0)+,(R1)+         ;POP POINTERS
003144  022700  001000                          CMP     #1000,R0            ;ALL DONE??
003150  001372                                  BNE     4$                  ;BR IF NO.

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

003152  012706  001200                  .BEGIN: MOV     #STACK,SP           ;SET UP STACK
003156  013746  000006                          MOV     @#6,-(SP)           ;SAVE LOC 6
```

```
003162  013746  000004                          MOV     @#4,-(SP)         ;SAVE LOC 4
003166  005000                                  CLR     R0                ;START AT 0
003170  012737  003234  000004                  MOV     #2$,@#4           ;SET UP FOR TIME OUT
003176  005037  000006                          CLR     @#6               ;TO AUTOSIZE MEMORY
003202  005720                          6$:     TST     (R0)+             ;CHECK ADDRESS IN R0
003204  022700  157776                          CMP     #157776,R0        ;IS IT AT LEAST 28K
003210  001374                                  BNE     6$                ;BR IF NO
003212  162700  007776                          SUB     #7776,R0          ;SAVE 2K FOR MONITORS
003216  010037  001304                  7$:     MOV     R0,MEMLIM         ;STORE MEMORY LIMIT
003222  012637  000004                          MOV     (SP)+,@#4         ;RESTORE LOC 4
003226  012637  000006                          MOV     (SP)+,@#6         ;RESTORE LOC 6
003232  000413                                  BR      10$               ;CONTINUE
003234  022626                          2$:     CMP     (SP)+,(SP)+       ;ADJUST STACK
003236  162700  000004                          SUB     #4,R0             ;GET LAST GOOD ADDRESS
003242  162700  007776                          SUB     #7776,R0          ;SAVE 2K FOR MONITORS
003246  022700  030000                          CMP     #30000,R0         ;IS IT 8K?
003252  001361                                  BNE     7$                ;BR IF NO
003254  012700  037400                          MOV     #37400,R0         ;IF 8K DON'T SAVE 2K
003260  000756                                  BR      7$                ;
003262  012737  000340  177776          10$:    MOV     #340,PS           ;LOCK OUT INTERRUPTS
003270  032737  000004  001236                  BIT     #BIT2,STRTSW      ;CHECK FOR LOCK ON TEST
003276  001411                                  BEQ     1$                ;BR IF NO LOCK DESIRED.
003300  104402  006043                          TYPE    ,MLOCK            ;TYPE LOCK SELECTED.
003304  012737  000240  003612                  MOV     #NOP,TTST         ;ADJUST SCOPE ROUTINE.
003312  012737  000240  003614                  MOV     #NOP,TTST+2       ;SET UP TO LOCK
003320  000406                                  BR      3$                ;CONTINUE ALONG.
003322  013737  003730  003612          1$:     MOV     BRW,TTST          ;PREPARE NORMAL SCOPE ROUTINE
003330  013737  003732  003614                  MOV     BRX,TTST+2        ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
003336  012737  010060  001214          3$:     MOV     #CYCLE,RETURN     ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
003344  032737  000002  001236          4$:     BIT     #SW01,STRTSW      ;IS TEST NO. SELECTED?
003352  001002                                  BNE     5$                ;BR IF YES
003354  104402  005755                          TYPE    ,MR               ;TYPE R
003360  000177  175630                  5$:     JMP     @RETURN           ;START TESTING
```

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

                                           .EOP:    RESET                       ;MAKE THE  WORLD CLEAN AGAIN.
003364  000005
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  005672                          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
```

```
004464  000002                           RTI                               ;LEAVE

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

004466  104402  005672           .CONVR: TYPE    ,MCRLF
004472  010046                   .CNVRT: MOV     R0,-(SP)
004474  010146                           MOV     R1,-(SP)
004476  010346                           MOV     R3,-(SP)
004500  010446                           MOV     R4,-(SP)
004502  010546                           MOV     R5,-(SP)
004504  017601  000012                   MOV     @12(SP),R1
004510  062766  000002  000012           ADD     #2,12(SP)
004516  012137  004710                   MOV     (R1)+,WRDCNT
004522  112137  004712           1$:     MOVB    (R1)+,CHRCNT
004526  112137  004713                   MOVB    (R1)+,SPACNT
004532  013137  004714                   MOV     @(R1)+,BINWRD
004536  122737  000003  004712           CMPB    #3,CHRCNT
004544  001003                           BNE     2$
004546  042737  177400  004714           BIC     #177400,BINWRD
004554  013704  004714           2$:     MOV     BINWRD,R4
004560  113705  004712                   MOVB    CHRCNT,R5
004564  012700  001416                   MOV     #TEMP,R0
004570  010403                   3$:     MOV     R4,R3
004572  042703  177770                   BIC     #177770,R3
004576  062703  000060                   ADD     #060,R3
004602  110320                           MOVB    R3,(R0)+
004604  000241                           CLC
004606  006004                           ROR     R4
004610  000241                           CLC
004612  006004                           ROR     R4
004614  000241                           CLC
004616  006004                           ROR     R4
004620  005305                           DEC     R5
004622  001362                           BNE     3$
004624  012703  007544                   MOV     #MDATA,R3
004630  114023                   4$:     MOVB    -(R0),(R3)+
004632  105337  004712                   DECB    CHRCNT
004636  001374                           BNE     4$
004640  105737  004713                   TSTB    SPACNT
004644  001405                           BEQ     6$
004646  112723  000040           5$:     MOVB    #040,(R3)+
004652  105337  004713                   DECB    SPACNT
004656  001373                           BNE     5$
004660  105013                   6$:     CLRB    (R3)
004662  104402  007544                   TYPE    ,MDATA
004666  005337  004710                   DEC     WRDCNT
004672  001313                           BNE     1$
004674  012605                           MOV     (SP)+,R5
004676  012604                           MOV     (SP)+,R4
004700  012603                           MOV     (SP)+,R3
004702  012601                           MOV     (SP)+,R1
004704  012600                           MOV     (SP)+,R0
004706  000002                           RTI
004710  000000                   WRDCNT: 0
004712  000000                   CHRCNT: 0
```

```
        004713                          SPACNT=CHRCNT+1
004714  000000                          BINWRD: 0

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

004716  011646                          .TRPSR: MOV     (SP),-(SP)      ;GET PC OF RETURN
004720  162716  000002                          SUB     #2,(SP)         ;=PC OF TRAP
004724  017616  000000                          MOV     @(SP),(SP)      ;GET TRP
004730  006316                          TRPOK:  ASL     (SP)            ;MULTIPLY TRAP ARG BY 2
004732  042716  177001                          BIC     #177001,(SP)    ;CLEAR UNWANTED BITS
004736  062716  001330                          ADD     #.TRPTAB,(SP)   ;POINTER TO SUBROUTINE ADDRESS
004742  017616  000000                          MOV     @(SP),(SP)      ;SUBROUTINE ADDRESS
004746  000136                                  JMP     @(SP)+          ;GO TO SUBROUTINE

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

004750  004737  007606                  .HLT:   JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
004754  032777  010000  174220                  BIT     #SW12,@SWR      ;BELL ON ERROR?
004762  001406                                  BEQ     XBX             ;BR IF NO BELL
004764  105777  174220                          TSTB    @TPCSR          ;TTY READY.
004770  100003                                  BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
004772  112777  000207  174212                  MOVB    #207,@TPDBR     ;PUSH A BELL AT THE TTY.
005000  032777  020000  174174          XBX:    BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
005006  001105                                  BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
005010  021637  001234                          CMP     (SP),LSTERR     ;WAS THIS ERROR FOUND LAST TIME?
005014  001404                                  BEQ     1$              ;BR IF YES
005016  011637  001234                          MOV     (SP),LSTERR     ;RECORD BEING HERE
005022  105037  001325                          CLRB    ERRFLG          ;PREPARE HEADER
005026  104406                          1$:     SAVOS                   ;SAVE ALL PROC REGISTERS
005030  011605                                  MOV     (SP),R5         ;GET THE PC OF ERROR
005032  162705  000002                          SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
005036  011504                                  MOV     (R5),R4         ;GET HLT INSTRUCTION
005040  006304                                  ASL     R4              ;MULT BY TWO
005042  061504                                  ADD     (R5),R4         ;DOUBLE IT
005044  006304                                  ASL     R4              ;MULT AGAIN
005046  042704  177001                          BIC     #177001,R4      ;CLEAR JUNK
005052  062704  031660                          ADD     #.ERRTAB,R4     ;GET POINTER
005056  012437  005172                          MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
005062  012437  005204                          MOV     (R4)+,DATAHD    ;GET DATA HEADRER
005066  011437  005216                          MOV     (R4),DATABP     ;GET DATA TABLE
005072  105737  001325                          TSTB    ERRFLG          ;TYPE HEADREER
005076  001403                                  BEQ     TYPMSG          ;BR IF YES
005100  005737  005216                          TST     DATABP          ;DOES DATA TABLE EXIST?
005104  001040                                  BNE     TYPDAT          ;BR IF YES.
005106  104402  005672                  TYPMSG: TYPE    ,MCRLF
005112  104402  005672                          TYPE    ,MCRLF
005116  005737  001220                          TST     LOCK
005122  001402                                  BEQ     1$
005124  104402  006142                          TYPE    ,MASTEK
005130  104402  006130                  1$:     TYPE    ,MTSTN
005134  104411  005330                          CNVRT   ,XTSTN          ;SHOW IT
```

```
 005140  104402  006217                    TYPE    ,MERRPC         ;TYPE PC.
 005144  104411  005322                    CNVRT   ,ERTABO         ;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:  RESOS                   ;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$:      PUSHRO                  ;SAVE RO
 005242  016600  000002                    MOV     2(SP),RO        ;SHOW ERROR PC IN DATA LIGHTS
 005246  000000                            HALT                    ;HALT
 005250  012600                            POPRO                   ;GET RO
 005252  005237  001232           EXITER:  INC     ERRCNT          ;UPDATE ERROR COUNT
 005256  032777  000400  173716            BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
 005264  001007                            BNE     1$              ;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                   ERTABO:  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$:
     005626  104414                                    ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
     005630  021364                                    021364                      ;PORT4+IBUS* REG11
     005632  032777  000002  173552                    BIT     #2,@DMP04           ;IS PGM CLOCK BIT CLEAR?
     005640  001772                                    BEQ     1$                  ;BR IF YES
     005642                                   2$:
     005642  104414                                    ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
     005644  021364                                    021364                      ;PORT4+IBUS* REG11
     005646  032777  000002  173536                    BIT     #2,@DMP04           ;IS PGM CLOCK BIT SET?
     005654  001372                                    BNE     2$                  ;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 DZDME  /
 (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:   .ASCII  <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) 006450  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                           XSTATQ:  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  000752                              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,DMP04
010270  060037  001412                          ADD     R0,DMP04
010274  013737  001412  001414                  MOV     DMP04,DMP06
010302  060037  001414                          ADD     R0,DMP06

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  006130                                    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$:     TST     (R0)+               ;POP R0
010460  020027  026010                            CMP     R0,#TLAST+10        ;AT END YET?
010464  001351                                    BNE     5$                  ;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$:    TYPE    ,MCRLF
010616  104410                                  CONVRT                   ;TYPE WHICH DMC IS BEING DONE
010620  012002                                  WHICH                    ;TEMP5 IS WHICH DMC
010622  005237  001256                          INC     TEMP5
010626  104403                                  INSTR
010630  006510                                  CSR
010632  104405                                  PARAM
010634  160000                                  160000
010636  164000                                  164000
010640  001254                                  TEMP4
010642     000                                  .BYTE   0
010643     001                                  .BYTE   1
010644  013722  001254                          MOV     TEMP4,(R2)+      ;STORE CSR IN MAP
010650  104403                                  INSTR
010652  006526                                  VEC
010654  104405                                  PARAM
010656  000000                                  0
010660  000776                                  776
010662  001254                                  TEMP4
010664     000                                  .BYTE   0
010665     001                                  .BYTE   1
010666  013712  001254                          MOV     TEMP4,(R2)       ;STORE VECTOR IN MAP
010672  104402                          10$:    TYPE
010674  006547                                  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 REMOTE
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  007054                                  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     #BIT8,(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$:    TST     (R2)+             ;POP POINTER
011602  012761  021324  000006                 MOV     #021324,6(R1)     ;PUT INSTRUCTION IN PORT6
011610  012711  001400                         MOV     #BIT9!BIT8,(R1)   ;PORT4<LU 15
011614  156122  000004                         BISB    4(R1),(R2)+       ;STORE DDCMP LINE # IN TABLE
011620  012761  021344  000006                 MOV     #021344,6(R1)     ;PORT6<INSTRUCTION
011626  012711  001400                         MOV     #BIT8!BIT9,(R1)   ;CLOCK INSTR.
011632  156122  000004                         BISB    4(R1),(R2)+       ;STORE BM873 ADD IN TABLE
011636  005722                                 TST     (R2)+             ;POP OVER STAT3
011640  005011                                 CLR     (R1)              ;CLEAR ROMI
011642  005237  001310                         INC     DMNUM             ;UPDATE DEVICE COUNTER
011646  022737  000020  001310                 CMP     #20,DMNUM         ;ARE MAX. NO. OF DEV FOUND?
011654  001412                                 BEQ     13$               ;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
```

```
                                  02100

                                                      ;*************************** TEST 1 ***************************
                                                      ;*OUT CONTROL REGISTER READ/ONLY TEST
                                                      ;*DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
                                                      ;*BITS ARE IN THE CORRECT STATE
                                                      ;:*************************************************************

                                                      ;  TEST 1
                                                      ;---------------
012320  012737  000001  001226            TST1:   MOV     #1,TSTNO
012326  012737  012374  001216                    MOV     #TST2,NEXT
                                                                              ;R1 CONTAINS BASE DMC11 ADDRESS
012334  005077  167044                            CLR     @DMCSR              ;CLEAR SEL0
012340  012702  000011                            MOV     #11,R2              ;SAVE R2 FOR TYPEOUT
012344  104414                                    ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
012346  021224                                    021004!<20*11>              ;PORT4<LINE UNIT REG 11
012350  016104  000004                            MOV     4(R1),R4            ;PUT "FOUND" IN R4
012354  042704  000054                            BIC     #54,R4              ;CLEAR UNKNOWN BITS
012360  012705  000020                            MOV     #20,R5              ;PUT "EXPECTED" IN R5
012364  120504                                    CMPB    R5,R4               ;IS OUT READY SET?
012366  001401                                    BEQ     1$                  ;BR IF YES
012370  104002                                    HLT     2                   ;ERROR IN LU 11
012372  104400                            1$:     SCOPE                       ;SCOPE THIS TEST

                                                      ;*************************** TEST 2 ***************************
                                                      ;*IN CONTROL REGISTER READ/ONLY TEST
                                                      ;*DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
                                                      ;*BITS ARE IN THE CORRECT STATE
                                                      ;:*************************************************************

                                                      ;  TEST 2
                                                      ;---------------
012374  012737  000002  001226            TST2:   MOV     #2,TSTNO
012402  012737  012442  001216                    MOV     #TST3,NEXT
                                                                              ;R1 CONTAINS BASE DMC11 ADDRESS
012410  012702  000012                            MOV     #12,R2              ;SAVE R2 FOR TYPEOUT
012414  104414                                    ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
012416  021244                                    021004!<20*12>              ;PORT4<LINE UNIT REG 12
012420  016104  000004                            MOV     4(R1),R4            ;PUT "FOUND" IN R4
012424  042704  000017                            BIC     #17,R4              ;CLEAR UNKNOWN BITS
012430  005005                                    CLR     R5                  ;PUT "EXPECTED" IN R5
012432  120504                                    CMPB    R5,R4               ;ARE ALL BITS CLEARED?
012434  001401                                    BEQ     1$                  ;BR IF YES
012436  104002                                    HLT     2                   ;ERROR IN LU 12
012440  104400                            1$:     SCOPE                       ;SCOPE THIS TEST

                                                      ;*************************** TEST 3 ***************************
                                                      ;*MODEM CONTROL REGISTER READ/ONLY TEST
                                                      ;*DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
                                                      ;*BITS ARE IN THE CORRECT STATE
                                                      ;:*************************************************************
```

```

                                                       ;  TEST 3
                                                       ;---------------
012442  012737  000003  001226          TST3:   MOV     #3,TSTNO
012450  012737  012514  001216                  MOV     #TST4,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
012456  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
012460  012702  000013                          MOV     #13,R2          ;SAVE R2 FOR TYPEOUT
012464  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
012466  021264                                  021004!<20*13>          ;PORT4←LINE UNIT REG 13
012470  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
012474  042704  000213                          BIC     #213,R4         ;CLEAR UNKNOWN BITS
012500  012705  000100                          MOV     #100,R5         ;PUT "EXPECTED" IN R5
012504  120504                                  CMPB    R5,R4           ;ARE RING, DTR,  AND MODEM READY SET?
012506  001401                                  BEQ     1$              ;BR IF YES
012510  104002                                  HLT     2               ;ERROR IN LU 13
012512  104400                          1$:     SCOPE                   ;SCOPE THIS TEST


                                                ;*************************** TEST 4 ***************************
                                                ;*MAINTENANCE REGISTER READ/ONLY TEST
                                                ;*DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
                                                ;*BITS ARE IN THE CORRECT STATE
                                                ;:*************************************************************

                                                ;  TEST 4
                                                ;---------------
012514  012737  000004  001226          TST4:   MOV     #4,TSTNO
012522  012737  012616  001216                  MOV     #TST5,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
012530  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
012532  012702  000017                          MOV     #17,R2          ;SAVE R2 FOR TYPEOUT
012536  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
012540  021364                                  021004!<20*17>          ;PORT4←LINE UNIT REG 17
012542  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
012546  042704  000206                          BIC     #206,R4         ;CLEAR UNKNOWN BITS
012552  012705  000051                          MOV     #51,R5          ;PUT "EXPECTED" IN R5
012556  032737  020000  001366                  BIT     #BIT13,STAT1    ;IS LU AN M8202 OR M8201?
012564  001004                                  BNE     2$              ;BR IF M8202
012566  032737  040000  001366                  BIT     #BIT14,STAT1    ;CONNECTOR???
012574  001004                                  BNE     3$              ;BR IF M8201 WITH CONNECTOR
012576  042704  000040                  2$:     BIC     #40,R4          ;MASK OFF SI BIT IF M8202 OR M8201, NO CONNECTOR
012602  042705  000040                          BIC     #BIT5,R5        ;SI BIT IS UNKNOWN
012606                                  3$:
012606  120504                                  CMPB    R5,R4           ;ARE SI AND ICIR SET?
012610  001401                                  BEQ     1$              ;BR IF YES
012612  104002                                  HLT     2               ;ERROR IN LU 17
012614  104400                          1$:     SCOPE                   ;SCOPE THIS TEST


                                                ;*************************** TEST 5 ***************************
                                                ;*LINE UNIT REGISTER WRITE/READ TEST
                                                ;*SET BITS IN LU REGISTER 12, VERIFY IT IS SET
                                                ;*CLEAR BITS IN LU REGISTER 12, VERIFY IT IS CLEAR
                                                ;:*************************************************************

```

```
                                                        ;  TEST 5
                                                        ;---------------
012616  012737  000005  001226          TST5:   MOV     #5,TSTNO
012624  012737  012756  001216                  MOV     #TST6,NEXT
012632  012737  012646  001220                  MOV     #1$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
012640  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
012642  012702  000012                          MOV     #12,R2          ;SAVE REGISTER ADDRESS FOR TYPEOUT
012646  012761  000040  000004          1$:     MOV     #40,4(R1)       ;LOAD PORT4
012654  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
012656  122112                                  122112                  ;SET BITS IN LU-12
012660  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
012662  021245                                  021245                  ;READ LU-12
012664  012705  000040                          MOV     #40,R5          ;PUT "EXPECTED" IN R5
012670  116104  000005                          MOVB    5(R1),R4        ;PUT "FOUND" IN R4
012674  042704  000337                          BIC     #337,R4         ;CLEAR UNWANTED BITS
012700  120504                                  CMPB    R5,R4           ;IS BITS SET?
012702  001401                                  BEQ     2$              ;BR IF YES
012704  104003                                  HLT     3               ;ERROR, BIT 5 IS NOT SET
012706  104401                          2$:     SCOP1                   ;SCOPE SUBTEST (SW09=1)
012710  012737  012716  001220                  MOV     #3$,LOCK        ;NEW SCOP1
012716  005061  000004                  3$:     CLR     4(R1)           ;LOAD PORT4
012722  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
012724  122112                                  122112                  ;CLEAR BIT 5 IN LU-12
012726  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
012730  021245                                  021245                  ;READ LU-12
012732  005005                                  CLR     R5              ;PUT "EXPECTED" IN R5
012734  116104  000005                          MOVB    5(R1),R4        ;PUT "FOUND" IN R4
012740  042704  000337                          BIC     #337,R4         ;CLEAR UNWANTED BITS
012744  120504                                  CMPB    R5,R4           ;IS BITS CLEAR?
012746  001401                                  BEQ     4$              ;BR IF YES
012750  104003                                  HLT     3               ;ERROR, BITS IS NOT CLEAR
012752  104401                          4$:     SCOP1                   ;SCOPE SUBTEST (SW09=1)
012754  104400                                  SCOPE                   ;SCOPE THIS TEST



                                                ;*************************** TEST 6 ***************************
                                                ;*LINE UNIT REGISTER WRITE/READ TEST
                                                ;*SET BIT1 IN LU REGISTER 17, VERIFY IT IS SET
                                                ;*CLEAR BIT1 IN LU REGISTER 17, VERIFY IT IS CLEAR
                                                ;:*************************************************************


                                                ;  TEST 6
                                                ;---------------
012756  012737  000006  001226          TST6:   MOV     #6,TSTNO
012764  012737  013116  001216                  MOV     #TST7,NEXT
012772  012737  013006  001220                  MOV     #1$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
013000  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
013002  012702  000017                          MOV     #17,R2          ;SAVE REGISTER ADDRESS FOR TYPEOUT
013006  012761  000001  000004          1$:     MOV     #1,4(R1)        ;LOAD PORT4
013014  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013016  122117                                  122117                  ;SET BIT1 IN LU-17
013020  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013022  021365                                  021365                  ;READ LU-17
013024  012705  000001                          MOV     #1,R5           ;PUT "EXPECTED" IN R5
```

```
013030  116104  000005                         MOVB    5(R1),R4        ;PUT "FOUND" IN R4
013034  042704  000376                         BIC     #376,R4         ;CLEAR UNWANTED BITS
013040  120504                                 CMPB    R5,R4           ;IS BIT1 SET?
013042  001401                                 BEQ     2$              ;BR IF YES
013044  104003                                 HLT     3               ;ERROR, BIT 1 IS NOT SET
013046  104401                         2$:     SCOP1                   ;SCOPE SUBTEST (SW09=1)
013050  012737  013056  001220                 MOV     #3$,LOCK        ;NEW SCOP1
013056  005061  000004                 3$:     CLR     4(R1)           ;LOAD PORT4
013062  104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013064  122117                                 122117                  ;CLEAR BIT 1 IN LU-17
013066  104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013070  021365                                 021365                  ;READ LU-17
013072  005005                                 CLR     R5              ;PUT "EXPECTED" IN R5
013074  116104  000005                         MOVB    5(R1),R4        ;PUT "FOUND" IN R4
013100  042704  000376                         BIC     #376,R4         ;CLEAR UNWANTED BITS
013104  120504                                 CMPB    R5,R4           ;IS BIT1 CLEAR?
013106  001401                                 BEQ     4$              ;BR IF YES
013110  104003                                 HLT     3               ;ERROR, BIT1 IS NOT CLEAR
013112  104401                         4$:     SCOP1                   ;SCOPE SUBTEST (SW09=1)
013114  104400                                 SCOPE                   ;SCOPE THIS TEST


                                               ;*************************** TEST 7 ***************************
                                               ;*LINE UNIT REGISTER WRITE/READ TEST
                                               ;*FLOAT A 1 THROUGH LINE UNIT REGISTER 13
                                               ;*FLOAT A 0 THROUGH LINE UNIT REGISTER 13
                                               ;:*************************************************************

                                               ;  TEST 7
                                               ;--------------
013116  012737  000007  001226         TST7:   MOV     #7,TSTNO
013124  012737  013326  001216                 MOV     #TST10,NEXT
013132  012737  013152  001220                 MOV     #64$,LOCK
                                                                       ;R1 CONTAINS BASE DMC11 ADDRESS
013140  104412                                 MSTCLR                  ;MASTER CLEAR DMC11
013142  012702  000013                         MOV     #13,R2          ;SAVE REGISTER ADDRESS FOR TYPEOUT
013146  012700  000001                         MOV     #1,R0           ;START WITH BIT 0
013152                                 64$:
013152  010061  000004                         MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
013156  042761  000257  000004                 BIC     #257,4(R1)      ;CLEAR UNWANTED BITS
013164  104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013166  122113                                 122100!13               ;MOV DATA TO IBUS REGISTER 13
013170  104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013172  021265                                 21005!<13*20>           ;READ FROM IBUS REGISTER 13
013174  010005                                 MOV     R0,R5           ;PUT EXPECTED IN R5
013176  042705  000257                         BIC     #257,R5         ;CLEAR UNWANTED BITS
013202  116104  000005                         MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
013206  042704  000257                         BIC     #257,R4         ;CLEAR UNWANTED BITS
013212  120504                                 CMPB    R5,R4           ;DATA CORRECT?
013214  001401                                 BEQ     65$             ;BR IF YES
013216  104003                                 HLT     3               ;ERROR
013220  104401                         65$:    SCOP1                   ;SW09=1?
013222  000241                                 CLC                     ;CLEAR CARRY
013224  106100                                 ROLB    R0              ;SHIFT BIT IN R0
013226  001351                                 BNE     64$             ;IF R0=0 THEN DONE
013230  012737  013244  001220                 MOV     #67$,LOCK       ;NEW SCOP1
```

```
013236  012700  000001                              MOV     #1,R0           ;START WITH BIT 0
013242  005100                              69$:    COM     R0              ;CHANGE TO FLOATING ZERO
013244                                      67$:
013244  010061  000004                              MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
013250  042761  000257  000004                      BIC     #257,4(R1)      ;CLEAR UNWANTED BITS
013256  104414                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013260  122113                                      122100!13               ;MOV DATA TO IBUS REGISTER 13
013262  104414                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013264  021265                                      21005!<13*20>           ;READ FROM IBUS REGISTER 13
013266  010005                                      MOV     R0,R5           ;PUT EXPECTED IN R5
013270  042705  000257                              BIC     #257,R5         ;CLEAR UNWANTED BITS
013274  116104  000005                              MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
013300  042704  000257                              BIC     #257,R4         ;CLEAR UNWANTED BITS
013304  120504                                      CMPB    R5,R4           ;DATA CORRECT?
013306  001401                                      BEQ     68$             ;BR IF YES
013310  104003                                      HLT     3               ;ERROR
013312  104401                              68$:    SCOP1                   ;SW09=1?
013314  005100                                      COM     R0              ;CHANGE TO FLOATING 1
013316  000241                                      CLC                     ;CLEAR CARRY
013320  106100                                      ROLB    R0              ;SHIFT BIT IN R0
013322  001347                                      BNE     69$             ;IF R0=0 THEN DONE
013324  104400                                      SCOPE                   ;SCOPE THIS TEST



                                                    ;*************************** TEST 10 **************************
                                                    ;*LINE UNIT REGISTER WRITE/READ TEST
                                                    ;*FLOAT A 1 THROUGH LINE UNIT REGISTER 14
                                                    ;*FLOAT A 0 THROUGH LINE UNIT REGISTER 14
                                                    ;:*************************************************************


                                                    ;  TEST 10
                                                    ;--------------
013326  012737  000010  001226              TST10:  MOV     #10,TSTNO
013334  012737  013502  001216                      MOV     #TST11,NEXT
013342  012737  013362  001220                      MOV     #64$,LOCK
                                                                            ;R1 CONTAINS BASE DMC11 ADDRESS
013350  104412                                      MSTCLR                  ;MASTER CLEAR DMC11
013352  012702  000014                              MOV     #14,R2          ;SAVE REGISTER ADDRESS FOR TYPEOUT
013356  012700  000001                              MOV     #1,R0           ;START WITH BIT 0
013362                                      64$:
013362  010061  000004                              MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
013366  104414                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013370  122114                                      122100!14               ;MOV DATA TO IBUS REGISTER 14
013372  104414                                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013374  021305                                      21005!<14*20>           ;READ FROM IBUS REGISTER 14
013376  010005                                      MOV     R0,R5           ;PUT EXPECTED IN R5
013400  116104  000005                              MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
013404  120504                                      CMPB    R5,R4           ;DATA CORRECT?
013406  001401                                      BEQ     65$             ;BR IF YES
013410  104003                                      HLT     3               ;ERROR
013412  104401                              65$:    SCOP1                   ;SW09=1?
013414  000241                                      CLC                     ;CLEAR CARRY
013416  106100                                      ROLB    R0              ;SHIFT BIT IN R0
013420  001360                                      BNE     64$             ;IF R0=0 THEN DONE
013422  012737  013436  001220                      MOV     #67$,LOCK       ;NEW SCOP1
013430  012700  000001                              MOV     #1,R0           ;START WITH BIT 0
```

```
013434  005100                          69$:    COM     R0                      ;CHANGE TO FLOATING ZERO
013436                                  67$:
013436  010061  000004                          MOV     R0,4(R1)                ;PUT PATTERN INTO PORT4
013442  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013444  122114                                  122100!14                       ;MOV DATA TO IBUS REGISTER 14
013446  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013450  021305                                  21005!<14*20>                   ;READ FROM IBUS REGISTER 14
013452  010005                                  MOV     R0,R5                   ;PUT EXPECTED IN R5
013454  116104  000005                          MOVB    5(R1),R4                ;PUT "FOUND" INTO R4
013460  120504                                  CMPB    R5,R4                   ;DATA CORRECT?
013462  001401                                  BEQ     68$                     ;BR IF YES
013464  104003                                  HLT     3                       ;ERROR
013466  104401                          68$:    SCOP1                           ;SW09=1?
013470  005100                                  COM     R0                      ;CHANGE TO FLOATING 1
013472  000241                                  CLC                             ;CLEAR CARRY
013474  106100                                  ROLB    R0                      ;SHIFT BIT IN R0
013476  001356                                  BNE     69$                     ;IF R0=0 THEN DONE
013500  104400                                  SCOPE                           ;SCOPE THIS TEST


                                                ;************************** TEST 11 ***************************
                                                ;*SWITCH PAC TEST
                                                ;*THIS TEST READS SWITCH PAC#1
                                                ;*THIS SWITCH PAC CONTAINS THE DDCMP LINE #
                                                ;:*************************************************************

                                                ;   TEST 11
                                                ;--------------
013502  012737  000011  001226          TST11:  MOV     #11,TSTNO
013510  012737  013544  001216                  MOV     #TST12,NEXT
                                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
013516  104412                                  MSTCLR                          ;MASTER CLEAR DMC11
013520  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013522  021324                                  021324                          ;PORT4←LU15
013524  016104  000004                          MOV     4(R1),R4                ;PUT "FOUND" IN R4
013530  113705  001370                          MOVB    STAT2,R5                ;PUT "EXPECTED" IN R5
013534  120504                                  CMPB    R5,R4                   ;SW OK?
013536  001401                                  BEQ     1$                      ;BR IF YES
013540  104031                                  HLT     31                      ;ERROR, SWITCH PAC READ ERROR
013542  104400                          1$:     SCOPE                           ;SCOPE THIS TEST


                                                ;************************** TEST 12 ***************************
                                                ;*SWITCH PAC TEST
                                                ;*THIS TEST READS SWITCH PAC#2
                                                ;*THIS SWITCH PAC CONTAINS THE BM873 BOOT ADD
                                                ;:*************************************************************

                                                ;   TEST 12
                                                ;--------------
013544  012737  000012  001226          TST12:  MOV     #12,TSTNO
013552  012737  013606  001216                  MOV     #TST13,NEXT
                                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
013560  104412                                  MSTCLR                          ;MASTER CLEAR DMC11
013562  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013564  021344                                  021344                          ;PORT4←LU16
```

```
013566  016104  000004                           MOV     4(R1),R4          ;PUT "FOUND" IN R4
013572  113705  001371                           MOVB    STAT2+1,R5        ;PUT "EXPECTED" IN R5
013576  120504                                   CMPB    R5,R4             ;SW OK?
013600  001401                                   BEQ     1$                ;BR IF YES
013602  104031                                   HLT     31                ;ERROR, SWITCH PAC READ ERROR
013604  104400                           1$:     SCOPE                     ;SCOPE THIS TEST



                                                 ;*************************** TEST 13 ***************************
                                                 ;*LINE UNIT CLOCK TEST
                                                 ;*THIS TEST VERIFYS THAT THE LU INTERNAL CLOCK
                                                 ;*(BIT 1 IN LU-17) IS WORKING
                                                 ;:***************************************************************

                                                 ;  TEST 13
                                                 ;--------------
013606  012737  000013  001226           TST13:  MOV     #13,TSTNO
013614  012737  013706  001216                   MOV     #TST14,NEXT
                                                                           ;R1 CONTAINS BASE DMC11 ADDRESS
013622  104412                                   MSTCLR                    ;MASTER CLEAR DMC11
013624  005037  001416                           CLR     TEMP              ;PREPARE FOR DELAY
013630                                   1$:
013630  104414                                   ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013632  021364                                   021364                    ;PORT4+LU-17
013634  032761  000002  000004                   BIT     #2,4(R1)          ;IS CLOCK BIT SET?
013642  001004                                   BNE     2$                ;BR IF YES
013644  005237  001416                           INC     TEMP              ;DELAY
013650  001367                                   BNE     1$      ;DELAY FINISHED?
013652  104004                                   HLT     4                 ;ERROR BIT IS STUCK CLEAR
013654  005037  001416                   2$:     CLR     TEMP              ;PREPARE FOR DELAY
013660                                   3$:
013660  104414                                   ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013662  021364                                   021364                    ;PORT4+LU-17
013664  032761  000002  000004                   BIT     #2,4(R1)          ;IS CLOCK BIT CLEAR?
013672  001404                                   BEQ     4$                ;BR IF YES
013674  005237  001416                           INC     TEMP              ;DELAY
013700  001367                                   BNE     3$                ;BR IF DELAY NOT DONE
013702  104004                                   HLT     4                 ;ERROR BIT IS STUCK SET
013704  104400                           4$:     SCOPE



                                                 ;*************************** TEST 14 ***************************
                                                 ;*OUT DATA SILO TEST
                                                 ;*SET SOM AND LOAD OUT DATA SILO
                                                 ;*VERIFY THAT OCOR SET, INDICATING THAT THE
                                                 ;*CHARACTER IS AT THE BOTTOM OF THE OUT SILO
                                                 ;:***************************************************************

                                                 ;  TEST 14
                                                 ;--------------
013706  012737  000014  001226           TST14:  MOV     #14,TSTNO
013714  012737  014006  001216                   MOV     #TST15,NEXT
                                                                           ;R1 CONTAINS BASE DMC11 ADDRESS
013722  104412                                   MSTCLR                    ;MASTER CLEAR DMC11
013724  012711  004000                           MOV     #BIT11,(R1)       ;SET LINE UNIT LOOP
013730  012761  000001  000004                   MOV     #1,4(R1)          ;LOAD PORT4 WITH BIT0
```

```
013736  104414                                 ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013740  122111                                 122111                          ;SET SOM
013742  104414                                 ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013744  122110                                 122110                          ;LOAD OUT DATA SILO
013746  104416  000002                         TIMER,  2                       ;WAIT FOR OCOR
013752  012702  000017                         MOV     #17,R2                  ;SAVE ADDRESS FOR TYPEOUT
013756  104414                                 ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013760  021364                                 021364                          ;PORT4←LU 17
013762  016104  000004                         MOV     4(R1),R4                ;PUT "FOUND" IN R4
013766  042704  000357                         BIC     #357,R4                 ;CLEAR UNWANTED BITS
013772  012705  000020                         MOV     #20,R5                  ;PUT "EXPECTED" IN R5
013776  120504                                 CMPB    R5,R4                   ;IS OCOR SET?
014000  001401                                 BEQ     1$                      ;BR IF YES
014002  104005                                 HLT     5
014004                                 1$:
014004  104400                                 SCOPE                           ;SCOPE THIS TEST



                                               ;*************************** TEST 15 ***************************
                                               ;*DDCMP TEST OF RTS AND OUT ACTIVE
                                               ;*SET SOM AND LOAD OUT DATA SILO
                                               ;*SINGLE STEP 2 DATA CLOCKS, VERIFY
                                               ;*THAT RTS AND ACTIVE ARE SET
                                               ;:**************************************************************


                                               ;  TEST 15
                                               ;--------------
014006  012737  000015  001226         TST15:  MOV     #15,TSTNO
014014  012737  014144  001216                 MOV     #TST16,NEXT
                                                                               ;R1 CONTAINS BASE DMC11 ADDRESS
014022  104412                                 MSTCLR                          ;MASTER CLEAR DMC11
014024  012711  004000                         MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
014030  012761  000001  000004                 MOV     #1,4(R1)                ;LOAD PORT4 WITH BIT0
014036  104414                                 ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014040  122111                                 122111                          ;SET SOM
014042  104414                                 ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014044  122110                                 122110                          ;LOAD OUT DATA SILO
014046  004737  026350                         JSR     PC,OCOR                 ;WAIT FOR OCOR
014052  104415  000002                         DATACLK,        2               ;CLOCK DATA FOUR TIMES
014056  012702  000011                         MOV     #11,R2                  ;SAVE ADDRESS FOR TYPEOUT
014062  104414                                 ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014064  021224                                 021224                          ;PORT4←LU 11
014066  016104  000004                         MOV     4(R1),R4                ;PUT "FOUND" IN R4
014072  042704  000257                         BIC     #257,R4                 ;CLEAR UNWANTED BITS
014076  012705  000120                         MOV     #120,R5                 ;PUT "EXPECTED" IN R5
014102  120504                                 CMPB    R5,R4                   ;IS ACTIVE SET?
014104  001401                                 BEQ     1$                      ;BR IF YES
014106  104005                                 HLT     5
014110                                 1$:
014110  012702  000013                         MOV     #13,R2                  ;SAVE ADDRESS FOR TYPEOUT
014114  104414                                 ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014116  021264                                 021264                          ;PORT4←LU 13
014120  016104  000004                         MOV     4(R1),R4                ;PUT EXPECTED IN R4
014124  042704  000337                         BIC     #337,R4                 ;CLEAR UNWANTED BITS
014130  012705  000040                         MOV     #BIT5,R5                ;PUT "EXPECTED" IN R5, RTS SHOULD BE SET
014134  120504                                 CMPB    R5,R4                   ;IS RTS OK?
```

```
014136  001401                            BEQ     2$              ;BR IF YES
014140  104005                            HLT     5               ;RTS ERROR
014142                            2$:
014142  104400                            SCOPE                   ;SCOPE THIS TEST


                                          ;************************** TEST 16 **************************
                                          ;*TEST OF OUT CLEAR
                                          ;*SET SOM AND LOAD OUT DATA SILO
                                          ;*SINGLE STEP DATA CLOCK, SET OUT CLEAR
                                          ;*VERIFY THAT OCOR,RTS, AND ACTIVE ARE CLEARED
                                          ;*************************************************************

                                          ;  TEST 16
                                          ;--------------
014144  012737  000016  001226    TST16:  MOV     #16,TSTNO
014152  012737  014342  001216            MOV     #TST17,NEXT
                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
014160  104412                            MSTCLR                  ;MASTER CLEAR DMC11
014162  012711  004000                    MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
014166  012761  000001  000004            MOV     #1,4(R1)        ;LOAD PORT4 WITH BIT0
014174  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014176  122111                            122111                  ;SET SOM
014200  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014202  122110                            122110                  ;LOAD OUT DATA SILO
014204  004737  026350                    JSR     PC,OCOR         ;WAIT FOR OCOR
014210  104415  000002                    DATACLK,       2        ;CLOCK DATA FOUR TIMES
014214  012761  000200  000004            MOV     #BIT7,4(R1)     ;SET BIT7 IN PORT4
014222  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014224  122111                            122111                  ;SET OUT CLEAR
014226  104415  000001                    DATACLK,       1        ;GIVE A TICK TO CLEAR RTS
014232  012702  000017                    MOV     #17,R2          ;SAVE ADDRESS FOR TYPEOUT
014236  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014240  021364                            021364                  ;PORT4←LU 17
014242  016104  000004                    MOV     4(R1),R4        ;PUT "FOUND" IN R4
014246  042704  000357                    BIC     #357,R4         ;CLEAR UNWANTED BITS
014252  005005                            CLR     R5              ;PUT "EXPECTED" IN R5
014254  120504                            CMPB    R5,R4           ;IS OCOR CLEARED?
014256  001401                            BEQ     1$              ;BR IF YES
014260  104005                            HLT     5
014262                            1$:
014262  012702  000013                    MOV     #13,R2          ;SAVE ADDRESS FOR TYPEOUT
014266  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014270  021264                            021264                  ;PORT4←LU 13
014272  016104  000004                    MOV     4(R1),R4        ;PUT EXPECTED IN R4
014276  042704  000337                    BIC     #337,R4         ;CLEAR UNWANTED BITS
014302  005005                            CLR     R5              ;PUT "EXPECTED" IN R5, RTS SHOULD BE CLEARED
014304  120504                            CMPB    R5,R4           ;IS RTS OK?
014306  001401                            BEQ     2$              ;BR IF YES
014310  104005                            HLT     5               ;RTS ERROR
014312                            2$:
014312  012702  000011                    MOV     #11,R2          ;SAVE ADDRESS FOR TYPEOUT
014316  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014320  021224                            021224                  ;PORT4←LU11
014322  016104  000004                    MOV     4(R1),R4        ;PUT "FOUND" IN R4
014326  012705  000020                    MOV     #BIT4,R5        ;ONLY OUT READY SHOULD BE SET
```

```
014332  120504                                  CMPB    R5,R4                   ;IS ACTIVE CLEAR?
014334  001401                                  BEQ     3$                      ;BR IF YES
014336  104005                                  HLT     5                       ;ERROR ACTIVE NOT CLEARED
014340                                  3$:
014340  104400                                  SCOPE                           ;SCOPE THIS TEST


                                                ;************************* TEST 17 *************************
                                                ;*DDCMP TRANSMITTER TEST
                                                ;*SINGLE CLOCK THE CHARACTER O
                                                ;*VERIFY EACH BIT POSITION AS IT
                                                ;*PASSES THE BIT WINDOW (SI BIT)
                                                ;*ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
                                                ;;*********************************************************

                                                ;  TEST 17
                                                ;--------------
014342  012737  000017  001226          TST17:  MOV     #17,TSTNO
014350  012737  014524  001216                  MOV     #TST20,NEXT
                                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
014356  104412                                  MSTCLR                          ;MASTER CLEAR DMC11
014360  012711  004000                          MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
014364  004737  026502                          JSR     PC,OUTRDY               ;WAIT FOR OUT-READY
014370  012761  000001  000004                  MOV     #1,4(R1)                ;SET BITO IN PORT4
014376  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014400  122111                                  122111                          ;SET SOM!
014402  012705  000000                          MOV     #0,R5   ;LOAD CHARACTER IN R5 FOR TYPEOUT
014406  004737  026502                          JSR     PC,OUTRDY               ;WAIT FOR OUT-READY
014412  010561  000004                          MOV     R5,4(R1)                ;LOAD PORT4 WITH CHARACTER
014416  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014420  122110                                  122110                          ;LOAD OUT DATA
014422  004737  026350                          JSR     PC,OCOR                 ;WAIT FOR OCOR TO SET
014426  005003                                  CLR     R3                      ;CLEAR BIT COUNTER
014430  010502                                  MOV     R5,R2                   ;LOAD CHARACTER IN R2
014432  104415  000002                          DATACLK,        2               ;2 TICKS TO SET UP TRANSMITTER
014436  104415  000001                  1$:     DATACLK,        1               ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
014442  106002                                  RORB    R2                      ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
014444  103005                                  BCC     2$                      ;BR IF CARRY CLEAR
014446  004737  026316                          JSR     PC,GETSI                ;GET THE WINDOW
014452  103406                                  BCS     3$                      ;BR IF BIT IS A MARK
014454  104006                                  HLT     6                       ;ERROR BIT WAS A SPACE
014456  000404                                  BR      3$                      ;CONTINE WITH TEST
014460  004737  026316                  2$:     JSR     PC,GETSI                ;GET THE WINDOW
014464  103001                                  BCC     3$                      ;BR IF BIT IS A SPACE
014466  104006                                  HLT     6                       ;ERROR BIT WAS A MARK
014470                                  3$:
014470  005203                                  INC     R3                      ;NEXT BIT
014472  022703  000010                          CMP     #10,R3                  ;DONE YET?
014476  001357                                  BNE     1$                      ;BR IF NO
014500  104415  000014                          DATACLK,        14              ;CLOCK TRANSMITTER 14 MORE TICKS
014504  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014506  021264                                  021264                          ;PORT4←LU-13
014510  032761  000040  000004                  BIT     #BIT5,4(R1)             ;RTS SHOULD BE CLEAR NOW
014516  001401                                  BEQ     4$                      ;BR IF YES
014520  104034                                  HLT     34                      ;ERROR, RTS NOT CLEAR
014522  104400                          4$:     SCOPE                           ;SCOPE THIS TEST
```

```


                                                        ;*************************** TEST 20 ***************************
                                                        ;*DDCMP TRANSMITTER TEST
                                                        ;*SINGLE CLOCK THE CHARACTER 125
                                                        ;*VERIFY EACH BIT POSITION AS IT
                                                        ;*PASSES THE BIT WINDOW (SI BIT)
                                                        ;*ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
                                                        ;;*************************************************************


                                                        ;  TEST 20
                                                        ;--------------
 014524  012737  000020  001226             TST20:  MOV     #20,TSTNO
 014532  012737  014706  001216                     MOV     #TST21,NEXT
                                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
 014540  104412                                     MSTCLR                      ;MASTER CLEAR DMC11
 014542  012711  004000                             MOV     #BIT11,(R1)         ;SET LINE UNIT LOOP
 014546  004737  026502                             JSR     PC,OUTRDY           ;WAIT FOR OUT-READY
 014552  012761  000001  000004                     MOV     #1,4(R1)            ;SET BIT0 IN PORT4
 014560  104414                                     ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 014562  122111                                     122111                      ;SET SOM!
 014564  012705  000125                             MOV     #125,R5 ;LOAD CHARACTER IN R5 FOR TYPEOUT
 014570  004737  026502                             JSR     PC,OUTRDY           ;WAIT FOR OUT-READY
 014574  010561  000004                             MOV     R5,4(R1)            ;LOAD PORT4 WITH CHARACTER
 014600  104414                                     ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 014602  122110                                     122110                      ;LOAD OUT DATA
 014604  004737  026350                             JSR     PC,OCOR             ;WAIT FOR OCOR TO SET
 014610  005003                                     CLR     R3                  ;CLEAR BIT COUNTER
 014612  010502                                     MOV     R5,R2               ;LOAD CHARACTER IN R2
 014614  104415  000002                             DATACLK,        2           ;2 TICKS TO SET UP TRANSMITTER
 014620  104415  000001                     1$:     DATACLK,        1           ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
 014624  106002                                     RORB    R2                  ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
 014626  103005                                     BCC     2$                  ;BR IF CARRY CLEAR
 014630  004737  026316                             JSR     PC,GETSI            ;GET THE WINDOW
 014634  103406                                     BCS     3$                  ;BR IF BIT IS A MARK
 014636  104006                                     HLT     6                   ;ERROR BIT WAS A SPACE
 014640  000404                                     BR      3$                  ;CONTINE WITH TEST
 014642  004737  026316                     2$:     JSR     PC,GETSI            ;GET THE WINDOW
 014646  103001                                     BCC     3$                  ;BR IF BIT IS A SPACE
 014650  104006                                     HLT     6                   ;ERROR BIT WAS A MARK
 014652                                     3$:
 014652  005203                                     INC     R3                  ;NEXT BIT
 014654  022703  000010                             CMP     #10,R3              ;DONE YET?
 014660  001357                                     BNE     1$                  ;BR IF NO
 014662  104415  000014                             DATACLK,        14          ;CLOCK TRANSMITTER 14 MORE TICKS
 014666  104414                                     ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 014670  021264                                     021264                      ;PORT4+LU-13
 014672  032761  000040  000004                     BIT     #BIT5,4(R1)         ;RTS SHOULD BE CLEAR NOW
 014700  001401                                     BEQ     4$                  ;BR IF YES
 014702  104034                                     HLT     34                  ;ERROR, RTS NOT CLEAR
 014704  104400                             4$:     SCOPE                       ;SCOPE THIS TEST


                                                        ;*************************** TEST 21 ***************************
                                                        ;*DDCMP TRANSMITTER TEST
                                                        ;*SINGLE CLOCK THE CHARACTER 252
                                                        ;*VERIFY EACH BIT POSITION AS IT
                                                        ;*PASSES THE BIT WINDOW (SI BIT)
                                                        ;*ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
                                                        ;:***********************************************************************

                                                        ;  TEST 21
                                                        ;---------------
014706  012737  000021  001226           TST21:   MOV     #21,TSTNO
014714  012737  015070  001216                    MOV     #TST22,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
014722  104412                                    MSTCLR                ;MASTER CLEAR DMC11
014724  012711  004000                            MOV     #BIT11,(R1)   ;SET LINE UNIT LOOP
014730  004737  026502                            JSR     PC,OUTRDY     ;WAIT FOR OUT-READY
014734  012761  000001  000004                    MOV     #1,4(R1)      ;SET BIT0 IN PORT4
014742  104414                                    ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014744  122111                                    122111                ;SET SOM!
014746  012705  000252                            MOV     #252,R5 ;LOAD CHARACTER IN R5 FOR TYPEOUT
014752  004737  026502                            JSR     PC,OUTRDY     ;WAIT FOR OUT-READY
014756  010561  000004                            MOV     R5,4(R1)      ;LOAD PORT4 WITH CHARACTER
014762  104414                                    ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014764  122110                                    122110                ;LOAD OUT DATA
014766  004737  026350                            JSR     PC,OCOR       ;WAIT FOR OCOR TO SET
014772  005003                                    CLR     R3            ;CLEAR BIT COUNTER
014774  010502                                    MOV     R5,R2         ;LOAD CHARACTER IN R2
014776  104415  000002                            DATACLK,        2     ;2 TICKS TO SET UP TRANSMITTER
015002  104415  000001                   1$:      DATACLK,        1     ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
015006  106002                                    RORB    R2            ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
015010  103005                                    BCC     2$            ;BR IF CARRY CLEAR
015012  004737  026316                            JSR     PC,GETSI      ;GET THE WINDOW
015016  103406                                    BCS     3$            ;BR IF BIT IS A MARK
015020  104006                                    HLT     6             ;ERROR BIT WAS A SPACE
015022  000404                                    BR      3$            ;CONTINE WITH TEST
015024  004737  026316                   2$:      JSR     PC,GETSI      ;GET THE WINDOW
015030  103001                                    BCC     3$            ;BR IF BIT IS A SPACE
015032  104006                                    HLT     6             ;ERROR BIT WAS A MARK
015034                                   3$:
015034  005203                                    INC     R3            ;NEXT BIT
015036  022703  000010                            CMP     #10,R3        ;DONE YET?
015042  001357                                    BNE     1$            ;BR IF NO
015044  104415  000014                            DATACLK,        14    ;CLOCK TRANSMITTER 14 MORE TICKS
015050  104414                                    ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015052  021264                                    021264                ;PORT4←LU-13
015054  032761  000040  000004                    BIT     #BIT5,4(R1)   ;RTS SHOULD BE CLEAR NOW
015062  001401                                    BEQ     4$            ;BR IF YES
015064  104034                                    HLT     34            ;ERROR, RTS NOT CLEAR
015066  104400                           4$:      SCOPE                 ;SCOPE THIS TEST


                                                        ;*************************** TEST 22 ***************************
                                                        ;*DDCMP TRANSMITTER TEST
                                                        ;*SINGLE CLOCK THE CHARACTER 377
                                                        ;*VERIFY EACH BIT POSITION AS IT
                                                        ;*PASSES THE BIT WINDOW (SI BIT)
                                                        ;*ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
                                                        ;:***********************************************************************
```

```
                                                ;  TEST 22
                                                ;--------------
015070  012737  000022  001226          TST22:  MOV     #22,TSTNO
015076  012737  015252  001216                  MOV     #TST23,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
015104  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
015106  012711  004000                          MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
015112  004737  026502                          JSR     PC,OUTRDY       ;WAIT FOR OUT-READY
015116  012761  000001  000004                  MOV     #1,4(R1)        ;SET BIT0 IN PORT4
015124  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015126  122111                                  122111                  ;SET SOM!
015130  012705  000377                          MOV     #377,R5 ;LOAD CHARACTER IN R5 FOR TYPEOUT
015134  004737  026502                          JSR     PC,OUTRDY       ;WAIT FOR OUT-READY
015140  010561  000004                          MOV     R5,4(R1)        ;LOAD PORT4 WITH CHARACTER
015144  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015146  122110                                  122110                  ;LOAD OUT DATA
015150  004737  026350                          JSR     PC,OCOR         ;WAIT FOR OCOR TO SET
015154  005003                                  CLR     R3              ;CLEAR BIT COUNTER
015156  010502                                  MOV     R5,R2           ;LOAD CHARACTER IN R2
015160  104415  000002                          DATACLK,        2       ;2 TICKS TO SET UP TRANSMITTER
015164  104415  000001                  1$:     DATACLK,        1       ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
015170  106002                                  RORB    R2              ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
015172  103005                                  BCC     2$              ;BR IF CARRY CLEAR
015174  004737  026316                          JSR     PC,GETSI        ;GET THE WINDOW
015200  103406                                  BCS     3$              ;BR IF BIT IS A MARK
015202  104006                                  HLT     6               ;ERROR BIT WAS A SPACE
015204  000404                                  BR      3$              ;CONTINE WITH TEST
015206  004737  026316                  2$:     JSR     PC,GETSI        ;GET THE WINDOW
015212  103001                                  BCC     3$              ;BR IF BIT IS A SPACE
015214  104006                                  HLT     6               ;ERROR BIT WAS A MARK
015216                                  3$:
015216  005203                                  INC     R3              ;NEXT BIT
015220  022703  000010                          CMP     #10,R3          ;DONE YET?
015224  001357                                  BNE     1$              ;BR IF NO
015226  104415  000014                          DATACLK,        14      ;CLOCK TRANSMITTER 14 MORE TICKS
015232  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015234  021264                                  021264                  ;PORT4←LU-13
015236  032761  000040  000004                  BIT     #BIT5,4(R1)     ;RTS SHOULD BE CLEAR NOW
015244  001401                                  BEQ     4$              ;BR IF YES
015246  104034                                  HLT     34              ;ERROR, RTS NOT CLEAR
015250  104400                          4$:     SCOPE                   ;SCOPE THIS TEST


                                                ;*************************** TEST 23 ***************************
                                                ;*DDCMP TRANSMITTER TEST
                                                ;*SINGLE CLOCK A BINARY COUNT PATTERN
                                                ;*VERIFY EACH BIT POSITION AS IT
                                                ;*PASSES THE BIT WINDOW (SI BIT)
                                                ;*ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
                                                ;*AND R5 CONTAINS THE CHARACTER THAT FAILED
                                                ;;*************************************************************

                                                ;  TEST 23
                                                ;--------------
015252  012737  000023  001226          TST23:  MOV     #23,TSTNO
015260  012737  015460  001216                  MOV     #TST24,NEXT
```

```
                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
015266  104412                                  MSTCLR                    ;MASTER CLEAR DMC11
015270  012711  004000                          MOV       #BIT11,(R1)     ;SET LINE UNIT LOOP
015274  005003                                  CLR       R3              ;R3 CONTAINS BIT COUNT
015276  005004                                  CLR       R4              ;R4 CONTAINS CHAR TO BE LOADED IN SILO
015300  005005                                  CLR       R5              ;R5 CONTAINS CHARACTER CURRENTLY BEING SHIFTED O
015302  004737  026502                          JSR       PC,OUTRDY       ;WAIT FOR OUT-READY
015306  012761  000001  000004                  MOV       #1,4(R1)        ;SET BITO IN PORT4
015314  104414                                  ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015316  122111                                  122111                    ;SET SOM!
015320  004737  026502                          JSR       PC,OUTRDY       ;WAIT FOR OUT-READY
015324  010461  000004                          MOV       R4,4(R1)        ;LOAD PORT4 WITH CHARACTER
015330  104414                                  ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015332  122110                                  122110                    ;LOAD OUT DATA
015334  005204                                  INC       R4              ;INCREMENT TO NEXT CHARACTER
015336  004737  026502                          JSR       PC,OUTRDY       ;WAIT FOR OUT-READY
015342  010461  000004                          MOV       R4,4(R1)        ;LOAD PORT4 WITH CHARACTER
015346  104414                                  ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015350  122110                                  122110                    ;LOAD OUT DATA
015352  004737  026350                          JSR       PC,OCOR         ;WAIT FOR OCOR TO SET
015356  104415  000002                          DATACLK,          2       ;2 TICKS TO SET UP TRANSMITTER
015362  005003                          4$:     CLR       R3              ;CLEAR BIT COUNTER
015364  010502                                  MOV       R5,R2           ;LOAD CHARACTER IN R2
015366  104415  000001                  1$:     DATACLK,          1       ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
015372  106002                                  RORB      R2              ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
015374  103005                                  BCC       2$              ;BR IF CARRY CLEAR
015376  004737  026316                          JSR       PC,GETSI        ;GET THE WINDOW
015402  103406                                  BCS       3$              ;BR IF BIT IS A MARK
015404  104006                                  HLT       6               ;ERROR BIT WAS A SPACE
015406  000404                                  BR        3$              ;CONTINE WITH TEST
015410  004737  026316                  2$:     JSR       PC,GETSI        ;GET THE WINDOW
015414  103001                                  BCC       3$              ;BR IF BIT IS A SPACE
015416  104006                                  HLT       6               ;ERROR BIT WAS A MARK
015420                                  3$:
015420  005203                                  INC       R3              ;NEXT BIT
015422  022703  000010                          CMP       #10,R3          ;DONE YET?
015426  001357                                  BNE       1$              ;BR IF NO
015430  005204                                  INC       R4              ;NEXT CHARACTER
015432  004737  026502                          JSR       PC,OUTRDY       ;WAIT FOR OUT-READY
015436  010461  000004                          MOV       R4,4(R1)        ;LOAD PORT4 WITH CHARACTER
015442  104414                                  ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015444  122110                                  122110                    ;LOAD OUT DATA
015446  005205                                  INC       R5              ;NEXT CHARACTER
015450  022705  000400                          CMP       #400,R5         ;DONE YET?
015454  001342                                  BNE       4$              ;BR IF NO
015456  104400                          5$:     SCOPE                     ;SCOPE THIS TEST



                                                ;************************** TEST 24 **************************
                                                ;*DDCMP STRIP SYNC TEST
                                                ;*SET LU LOOP, SINGLE STEP 5 SYNCS,
                                                ;*VERIFY THAT IN ACTIVE DOES NOT SET
                                                ;;***********************************************************

                                                ;  TEST 24
                                                ;--------------
```

```
015460  012737  000024  001226      TST24:  MOV     #24,TSTNO
015466  012737  015546  001216              MOV     #TST25,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
015474  104412                              MSTCLR                      ;MASTER CLEAR DMC11
015476  012711  004000                      MOV     #BIT11,(R1)         ;SET LU LOOP
015502  012702  000012                      MOV     #12,R2              ;SAVE LU REG FOR TYPEOUT
015506  004737  026366                      JSR     PC,SYNC             ;SINGLE CLOCK 5 SYNC CHARACTERS
015512  000005                              5
015514  104415  000054                      DATACLK,        54
015520  104414                              ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015522  021244                              021244                      ;PORT4+LU12
015524  016104  000004                      MOV     4(R1),R4            ;PUT "FOUND" IN R4
015530  042704  000277                      BIC     #277,R4             ;CLEAR UNWANTED BITS
015534  005005                              CLR     R5                  ;PUT "EXPECTED" IN R5
015536  120504                              CMPB    R5,R4               ;IS ACTIVE CLEAR?
015540  001401                              BEQ     1$                  ;BR IF YES
015542  104040                              HLT     40                  ;ERROR ACTIVE IS NOT CLEAR
015544  104400                      1$:     SCOPE                       ;SCOPE THIS TEST



                                            ;*************************** TEST 25 ***************************
                                            ;*DDCMP IN ACTIVE TEST
                                            ;*SET LU LOOP, SINGLE STEP 5 SYNCS AND A NON-SYNC (301)
                                            ;*VERIFY THAT IN ACTIVE IS SET
                                            ;:**************************************************************


                                            ;  TEST 25
                                            ;--------------
015546  012737  000025  001226      TST25:  MOV     #25,TSTNO
015554  012737  015636  001216              MOV     #TST26,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
015562  104412                              MSTCLR                      ;MASTER CLEAR DMC11
015564  012711  004000                      MOV     #BIT11,(R1)         ;SET LU LOOP
015570  012702  000012                      MOV     #12,R2              ;SAVE LU REG FOR TYPEOUT
015574  004737  026366                      JSR     PC,SYNC             ;SINGLE CLOCK 5 SYNC CHARACTERS
015600  000005                              5
015602  104415  000064                      DATACLK,        64
015606  104414                              ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015610  021244                              021244                      ;PORT4+LU12
015612  016104  000004                      MOV     4(R1),R4            ;PUT "FOUND" IN R4
015616  042704  000277                      BIC     #277,R4             ;CLEAR UNWANTED BITS
015622  012705  000100                      MOV     #BIT6,R5                    ;PUT "EXPECTED" IN R5
015626  120504                              CMPB    R5,R4               ;IS ACTIVE SET?
015630  001401                              BEQ     1$                  ;BR IF YES
015632  104040                              HLT     40                  ;ERROR ACTIVE IS NOT SET
015634  104400                      1$:     SCOPE                       ;SCOPE THIS TEST



                                            ;*************************** TEST 26 ***************************
                                            ;*DDCMP IN ACTIVE TEST
                                            ;*SET LU LOOP, SINGLE STEP 1 SYNC AND A NON-SYNC (301)
                                            ;*VERIFY THAT IN ACTIVE DOES NOT SET
                                            ;:**************************************************************


                                            ;  TEST 26
                                            ;--------------
```

```
015636  012737  000026  001226          TST26:  MOV     #26,TSTNO
015644  012737  015724  001216                  MOV     #TST27,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
015652  104412                                  MSTCLR          ;MASTER CLEAR DMC11
015654  012711  004000                          MOV     #BIT11,(R1)     ;SET LU LOOP
015660  012702  000012                          MOV     #12,R2          ;SAVE LU REG FOR TYPEOUT
015664  004737  026366                          JSR     PC,SYNC         ;SINGLE CLOCK 1 SYNC CHARACTERS
015670  000001                                  1
015672  104415  000024                          DATACLK,        24
015676  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015700  021244                                  021244                  ;PORT4←LU12
015702  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
015706  042704  000277                          BIC     #277,R4         ;CLEAR UNWANTED BITS
015712  005005                                  CLR     R5              ;PUT "EXPECTED" IN R5
015714  120504                                  CMPB    R5,R4           ;IS ACTIVE CLEAR?
015716  001401                                  BEQ     1$              ;BR IF YES
015720  104040                                  HLT     40              ;ERROR ACTIVE IS NOT CLEAR
015722  104400                          1$:     SCOPE                   ;SCOPE THIS TEST


                                                ;*************************** TEST 27 ***************************
                                                ;*DDCMP IN ACTIVE TEST
                                                ;*SET LU LOOP, SINGLE STEP 2 SYNCS AND A NON-SYNC (301)
                                                ;*VERIFY THAT IN ACTIVE IS SET
                                                ;;*************************************************************

                                                ;  TEST 27
                                                ;---------------
015724  012737  000027  001226          TST27:  MOV     #27,TSTNO
015732  012737  016014  001216                  MOV     #TST30,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
015740  104412                                  MSTCLR          ;MASTER CLEAR DMC11
015742  012711  004000                          MOV     #BIT11,(R1)     ;SET LU LOOP
015746  012702  000012                          MOV     #12,R2          ;SAVE LU REG FOR TYPEOUT
015752  004737  026366                          JSR     PC,SYNC         ;SINGLE CLOCK 2 SYNC CHARACTERS
015756  000002                                  2
015760  104415  000034                          DATACLK,        34
015764  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015766  021244                                  021244                  ;PORT4←LU12
015770  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
015774  042704  000277                          BIC     #277,R4         ;CLEAR UNWANTED BITS
016000  012705  000100                          MOV     #BIT6,R5                ;PUT "EXPECTED" IN R5
016004  120504                                  CMPB    R5,R4           ;IS ACTIVE SET?
016006  001401                                  BEQ     1$              ;BR IF YES
016010  104040                                  HLT     40              ;ERROR ACTIVE IS NOT SET
016012  104400                          1$:     SCOPE                   ;SCOPE THIS TEST


                                                ;*************************** TEST 30 ***************************
                                                ;*IN CLEAR TEST
                                                ;*SYNC UP RECEIVER AND TRANSMIT A CHARACTER
                                                ;*WAIT FOR IN RDY, THEN SET IN CLEAR
                                                ;*VERIFY THAT IN ACTIVE AND IN RDY ARE CLEARED
                                                ;;*************************************************************

                                                ;  TEST 30
                                                ;---------------
016014  012737  000030  001226          TST30:  MOV     #30,TSTNO
016022  012737  016166  001216                  MOV     #TST31,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
016030  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
016032  012702  000012                          MOV     #12,R2          ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
016036  012711  004000                          MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
016042  004737  026534                          JSR     PC,CHAR         ;LOAD SILO WITH 3 SYNCS
016046  000301                                  301                     ;AND A NON-SYNC  (301)
016050  104415  000053                          DATACLK,        53      ;SINGLE CLOCK THE DATA
016054  104416  000002                          TIMER,  2               ;WAIT FOR INRDY
016060  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016062  021244                                  021244                  ;PORT4←LU 12
016064  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
016070  042704  000357                          BIC     #357,R4         ;CLEAR UNWANTED BITS
016074  012705  000020                          MOV     #BIT4,R5                ;PUT "EXPECTED" IN R5
016100  120504                                  CMPB    R5,R4           ;IS INRDY SET?
016102  001401                                  BEQ     1$
016104  104040                                  HLT     40              ;ERROR, INRDY IS NOT SET
016106                                  1$:
016106  012761  000200  000004                  MOV     #BIT7,4(R1)     ;LOAD PORT4
016114  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016116  122112                                  122112                  ;SET IN CLEAR
016120  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016122  021244                                  021244                  ;PORT4←LU 12
016124  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
016130  042704  000277                          BIC     #277,R4         ;CLEAR UNWANTED BITS
016134  005005                                  CLR     R5              ;PUT "EXPECTED" IN R5
016136  120504                                  CMPB    R5,R4           ;IS IN ACTIVE CLEAR?
016140  001401                                  BEQ     2$
016142  104040                                  HLT     40              ;ERROR, IN ACTIVE IS NOT CLEAR
016144                                  2$:
016144  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
016150  042704  000357                          BIC     #357,R4         ;CLEAR UNWANTED BITS
016154  005005                                  CLR     R5              ;PUT "EXPECTED" IN R5
016156  120504                                  CMPB    R5,R4           ;IS INRDY CLEARED?
016160  001401                                  BEQ     3$
016162  104040                                  HLT     40              ;ERROR, INRDY IS NOT CLEARED
016164  104400                          3$:     SCOPE                   ;SCOPE THIS TEST


                                                ;**************************** TEST 31 ***************************
                                                ;*DDCMP BASIC RECEICER TEST
                                                ;*SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 0
                                                ;*VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED
                                                ;:***************************************************************

                                                ;  TEST 31
                                                ;---------------
016166  012737  000031  001226          TST31:  MOV     #31,TSTNO
016174  012737  016302  001216                  MOV     #TST32,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
016202  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
016204  012702  000012                          MOV     #12,R2          ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
016210  012711  004000                          MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
016214  004737  026534                          JSR     PC,CHAR         ;LOAD SILO WITH 3 SYNCS
```

```
016220  000000                            0                               ;AND THE CHARACTER 0
016222  104415  000053                    DATACLK,        53              ;SINGLE CLOCK THE DATA
016226  104416  000002                    TIMER,  2                       ;WAIT FOR INRDY
016232  104414                            ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016234  021244                            021244                          ;PORT4+LU 12
016236  016104  000004                    MOV     4(R1),R4                ;PUT "FOUND" IN R4
016242  042704  000357                    BIC     #357,R4                 ;CLEAR UNWANTED BITS
016246  012705  000020                    MOV     #BIT4,R5                        ;PUT "EXPECTED" IN R5
016252  120504                            CMPB    R5,R4                   ;IS INRDY SET?
016254  001401                            BEQ     1$
016256  104040                            HLT     40                      ;ERROR, INRDY IS NOT SET
016260                            1$:
016260  104414                            ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016262  021204                            021204                          ;PORT4+IN DATA
016264  016104  000004                    MOV     4(R1),R4                ;PUT "FOUND" IN R4
016270  005005                            CLR     R5                      ;PUT "EXPECTED" IN R5
016272  120504                            CMPB    R5,R4                   ;WAS A 0 RECEIVED?
016274  001401                            BEQ     2$
016276  104010                            HLT     10                      ;ERROR, RECEIVED DATA IS WRONG
016300  104400                    2$:     SCOPE                           ;SCOPE THIS TEST


                                          ;**************************** TEST 32 ****************************
                                          ;*DDCMP BASIC RECEICER TEST
                                          ;*SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 125
                                          ;*VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED
                                          ;:****************************************************************

                                          ;   TEST 32
                                          ;--------------
016302  012737  000032  001226    TST32:  MOV     #32,TSTNO
016310  012737  016420  001216            MOV     #TST33,NEXT
                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
016316  104412                            MSTCLR                          ;MASTER CLEAR DMC11
016320  012702  000012                    MOV     #12,R2                  ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
016324  012711  004000                    MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
016330  004737  026534                    JSR     PC,CHAR                 ;LOAD SILO WITH 3 SYNCS
016334  000125                            125                             ;AND THE CHARACTER 125
016336  104415  000053                    DATACLK,        53              ;SINGLE CLOCK THE DATA
016342  104416  000002                    TIMER,  2                       ;WAIT FOR INRDY
016346  104414                            ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016350  021244                            021244                          ;PORT4+LU 12
016352  016104  000004                    MOV     4(R1),R4                ;PUT "FOUND" IN R4
016356  042704  000357                    BIC     #357,R4                 ;CLEAR UNWANTED BITS
016362  012705  000020                    MOV     #BIT4,R5                        ;PUT "EXPECTED" IN R5
016366  120504                            CMPB    R5,R4                   ;IS INRDY SET?
016370  001401                            BEQ     1$
016372  104040                            HLT     40                      ;ERROR, INRDY IS NOT SET
016374                            1$:
016374  104414                            ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016376  021204                            021204                          ;PORT4+IN DATA
016400  016104  000004                    MOV     4(R1),R4                ;PUT "FOUND" IN R4
016404  012705  000125                    MOV     #125,R5                 ;PUT "EXPECTED" IN R5
016410  120504                            CMPB    R5,R4                   ;WAS A 125 RECEIVED?
016412  001401                            BEQ     2$
016414  104010                            HLT     10                      ;ERROR, RECEIVED DATA IS WRONG
```

```
016416  104400                          2$:     SCOPE                           ;SCOPE THIS TEST



                                                ;**************************** TEST 33 ****************************
                                                ;*DDCMP BASIC RECEICER TEST
                                                ;*SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 252
                                                ;*VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED
                                                ;:***************************************************************

                                                ;  TEST 33
                                                ;---------------
016420  012737  000033  001226          TST33:  MOV     #33,TSTNO
016426  012737  016536  001216                  MOV     #TST34,NEXT
                                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
016434  104412                                  MSTCLR                          ;MASTER CLEAR DMC11
016436  012702  000012                          MOV     #12,R2                  ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
016442  012711  004000                          MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
016446  004737  026534                          JSR     PC,CHAR                 ;LOAD SILO WITH 3 SYNCS
016452  000252                                  252                             ;AND THE CHARACTER 252
016454  104415  000053                          DATACLK,         53             ;SINGLE CLOCK THE DATA
016460  104416  000002                          TIMER,  2                       ;WAIT FOR INRDY
016464  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016466  021244                                  021244                          ;PORT4←LU 12
016470  016104  000004                          MOV     4(R1),R4                ;PUT "FOUND" IN R4
016474  042704  000357                          BIC     #357,R4                 ;CLEAR UNWANTED BITS
016500  012705  000020                          MOV     #BIT4,R5                        ;PUT "EXPECTED" IN R5
016504  120504                                  CMPB    R5,R4                   ;IS INRDY SET?
016506  001401                                  BEQ     1$
016510  104040                                  HLT     40                      ;ERROR, INRDY IS NOT SET
016512                                  1$:
016512  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016514  021204                                  021204                          ;PORT4←IN DATA
016516  016104  000004                          MOV     4(R1),R4                ;PUT "FOUND" IN R4
016522  012705  000252                          MOV     #252,R5                 ;PUT "EXPECTED" IN R5
016526  120504                                  CMPB    R5,R4                   ;WAS A 252 RECEIVED?
016530  001401                                  BEQ     2$
016532  104010                                  HLT     10                      ;ERROR, RECEIVED DATA IS WRONG
016534  104400                          2$:     SCOPE                           ;SCOPE THIS TEST



                                                ;**************************** TEST 34 ****************************
                                                ;*DDCMP BASIC RECEICER TEST
                                                ;*SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 377
                                                ;*VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED
                                                ;:***************************************************************

                                                ;  TEST 34
                                                ;---------------
016536  012737  000034  001226          TST34:  MOV     #34,TSTNO
016544  012737  016654  001216                  MOV     #TST35,NEXT
                                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
016552  104412                                  MSTCLR                          ;MASTER CLEAR DMC11
016554  012702  000012                          MOV     #12,R2                  ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
016560  012711  004000                          MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
016564  004737  026534                          JSR     PC,CHAR                 ;LOAD SILO WITH 3 SYNCS
016570  000377                                  377                             ;AND THE CHARACTER 377
```

```
016572  104415  000053                         DATACLK,        53       ;SINGLE CLOCK THE DATA
016576  104416  000002                         TIMER,  2                ;WAIT FOR INRDY
016602  104414                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016604  021244                                 021244                   ;PORT4←LU 12
016606  016104  000004                         MOV     4(R1),R4         ;PUT "FOUND" IN R4
016612  042704  000357                         BIC     #357,R4          ;CLEAR UNWANTED BITS
016616  012705  000020                         MOV     #BIT4,R5                 ;PUT "EXPECTED" IN R5
016622  120504                                 CMPB    R5,R4            ;IS INRDY SET?
016624  001401                                 BEQ     1$
016626  104040                                 HLT     40               ;ERROR, INRDY IS NOT SET
016630                                 1$:
016630  104414                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016632  021204                                 021204                   ;PORT4←IN DATA
016634  016104  000004                         MOV     4(R1),R4         ;PUT "FOUND" IN R4
016640  012705  000377                         MOV     #377,R5          ;PUT "EXPECTED" IN R5
016644  120504                                 CMPB    R5,R4            ;WAS A 377 RECEIVED?
016646  001401                                 BEQ     2$
016650  104010                                 HLT     10               ;ERROR, RECEIVED DATA IS WRONG
016652  104400                         2$:     SCOPE                    ;SCOPE THIS TEST


                                               ;************************** TEST 35 **************************
                                               ;*DDCMP DATA TEST
                                               ;*THIS TEST SINGLE STEPS A BINARY COUNT PATTERN
                                               ;*CHECKING EACH CHARACTER AS IT IS RECEIVED
                                               ;:************************************************************

                                               ;  TEST 35
                                               ;---------------
016654  012737  000035  001226         TST35:  MOV     #35,TSTNO
016662  012737  017004  001216                 MOV     #TST36,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
016670  104412                                 MSTCLR                   ;MASTER CLEAR DMC11
016672  005037  027152                         CLR     SCHAR            ;START BINARY COUNT AT ZERO
016676  005037  027154                         CLR     STUFLG           ;CLEAR BITSTUFF FLAG
016702  005002                                 CLR     R2               ;R2 IS "EXPECTED" DATA
016704  012703  000073                         MOV     #73,R3           ;R3 IS CHARACTER COUNT
016710  012711  004000                         MOV     #BIT11,(R1)      ;SET LINE UNIT LOOP
016714  004737  026712                         JSR     PC,SILOLD        ;LOAD SILO WITH COUNT PATTERN
016720  104415  000043                         DATACLK,        43       ;SYNC RECEIVER AND GET IT ACTIVE
016724  104415  000730                 1$:     DATACLK,        730      ;CLOCK IN 73 CHARACTERS
016730  004737  027156                 4$:     JSR     PC,INRDY         ;WAIT FOR INRDY
016734  104414                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016736  021204                                 021204                   ;PORT4←IN DATA
016740  016104  000004                         MOV     4(R1),R4         ;PUT "FOUND" IN R4
016744  010205                                 MOV     R2,R5            ;PUT "EXPECTED" IN R5
016746  120504                                 CMPB    R5,R4            ;IS DATA CORRECT?
016750  001401                                 BEQ     2$               ;BR IF YES
016752  104010                                 HLT     10               ;DATA ERROR
016754  005202                         2$:     INC     R2               ;NEXT CHARACTER
016756  022702  000400                         CMP     #400,R2          ;ALL DONE?
016762  001407                                 BEQ     3$               ;BR IF YES
016764  005303                                 DEC     R3               ;DECREMENT CHARACTER COUNT
016766  001360                                 BNE     4$               ;BR IF SILO NOT EMPTY
016770  004737  026712                         JSR     PC,SILOLD        ;LOAD SILO WITH MORE OF COUNT PATTERN
016774  012703  000073                         MOV     #73,R3           ;RELOAD CHARACTER COUNT
```

```
017000  000751                              BR      1$                  ;CONTINUE
017002  104400                      3$:     SCOPE                       ;SCOPE THIS TEST


                                            ;************************** TEST 36 **************************
                                            ;*DDCMP DATA TEST
                                            ;*THIS TEST SINGLE STEPS A BINARY COUNT PATTERN
                                            ;*CHECKING EACH CHARACTER AS IT IS RECEIVED
                                            ;*THIS TEST IS EXACTLY THE SAME AS THE LAST TEST,
                                            ;*EXCEPT LINE UNIT LOOP IS SET IN LU REGISTER 12
                                            ;;************************************************************

                                            ;  TEST 36
                                            ;--------------
017004  012737  000036  001226      TST36:  MOV     #36,TSTNO
017012  012737  017144  001216              MOV     #TST37,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
017020  104412                              MSTCLR                      ;MASTER CLEAR DMC11
017022  005037  027152                      CLR     SCHAR               ;START BINARY COUNT AT ZERO
017026  005037  027154                      CLR     STUFLG              ;CLEAR BITSTUFF FLAG
017032  005002                              CLR     R2                  ;R2 IS "EXPECTED" DATA
017034  012703  000073                      MOV     #73,R3              ;R3 IS CHARACTER COUNT
017040  005011                              CLR     (R1)                ;CLEAR LU LOOP IN MAINT REG
017042  012761  000040  000004              MOV     #BIT5,4(R1)         ;LOAD PORT4
017050  104414                              ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017052  122112                              122112                      ;SET LU LOOP IN LU REG 12
017054  004737  026712                      JSR     PC,SILOLD           ;LOAD SILO WITH COUNT PATTERN
017060  104415  000043                      DATACLK,        43          ;SYNC RECEIVER AND GET IT ACTIVE
017064  104415  000730              1$:     DATACLK,        730         ;CLOCK IN 73 CHARACTERS
017070  004737  027156              4$:     JSR     PC,INRDY            ;WAIT FOR INRDY
017074  104414                              ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017076  021204                              021204                      ;PORT4←IN DATA
017100  016104  000004                      MOV     4(R1),R4            ;PUT "FOUND" IN R4
017104  010205                              MOV     R2,R5               ;PUT "EXPECTED" IN R5
017106  120504                              CMPB    R5,R4               ;IS DATA CORRECT?
017110  001401                              BEQ     2$                  ;BR IF YES
017112  104010                              HLT     10                  ;DATA ERROR
017114  005202                      2$:     INC     R2                  ;NEXT CHARACTER
017116  022702  000400                      CMP     #400,R2             ;ALL DONE?
017122  001407                              BEQ     3$                  ;BR IF YES
017124  005303                              DEC     R3                  ;DECREMENT CHARACTER COUNT
017126  001360                              BNE     4$                  ;BR IF SILO NOT EMPTY
017130  004737  026712                      JSR     PC,SILOLD           ;LOAD SILO WITH MORE OF COUNT PATTERN
017134  012703  000073                      MOV     #73,R3              ;RELOAD CHARACTER COUNT
017140  000751                              BR      1$                  ;CONTINUE
017142  104400                      3$:     SCOPE                       ;SCOPE THIS TEST


                                            ;************************** TEST 37 **************************
                                            ;*TRANSMITTER MARK TEST
                                            ;*SINGLE CLOCK 3 SYNCS AND A 301 AND 20 EXTRA
                                            ;*CLOCK TICKS, VERIFY THAT A 301, A 377 AND A 377
                                            ;*WERE RECEIVED INDICATING THAT THE TRANSMITTER WENT
                                            ;*TO A MARK STATE FOR 16 BITS WHEN OUT SILO WAS EMPTY
                                            ;;************************************************************

```

```
                                        ;  TEST 37
                                        ;-------------
017144  012737  000037  001226  TST37:  MOV     #37,TSTNO
017152  012737  017304  001216          MOV     #TST40,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
017160  104412                          MSTCLR                  ;MASTER CLEAR DMC11
017162  012711  004000                  MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
017166  004737  026534                  JSR     PC,CHAR         ;LOAD SILO WITH 3 SYNCS
017172  000301                          301                     ;AND A 301
017174  104415  000073                  DATACLK,        73      ;CLOCK THE 301 IN AND 20 EXTRA TICKS
017200  004737  027156                  JSR     PC,INRDY        ;WAIT FOR INRDY
017204  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017206  021204                          021204                  ;PORT4←IN DATA
017210  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
017214  012705  000301                  MOV     #301,R5         ;PUT "EXPECTED" IN R5
017220  120504                          CMPB    R5,R4           ;WAS A 301 RECEIVED?
017222  001401                          BEQ     1$
017224  104010                          HLT     10              ;ERROR FIRST CHARACTER INCORRECT
017226  004737  027156          1$:     JSR     PC,INRDY        ;WAIT FOR INRDY
017232  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017234  021204                          021204                  ;PORT4←IN DATA
017236  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
017242  012705  000377                  MOV     #377,R5         ;PUT "EXPECTED" IN R5
017246  120504                          CMPB    R5,R4           ;WAS A 377 RECEIVED?
017250  001401                          BEQ     2$
017252  104010                          HLT     10              ;ERROR, 377 WAS NOT RECEIVED
017254  004737  027156          2$:     JSR     PC,INRDY        ;WAIT FOR INRDY
017260  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017262  021204                          021204                  ;PORT4←IN DATA
017264  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
017270  012705  000377                  MOV     #377,R5         ;PUT "EXPECTED" IN R5
017274  120504                          CMPB    R5,R4           ;WAS A 377 RECEIVED?
017276  001401                          BEQ     3$
017300  104010                          HLT     10              ;ERROR, 377 WAS NOT RECEIVED
017302  104400                  3$:     SCOPE                   ;SCOPE THIS TEST



                                        ;************************** TEST 40 ***************************
                                        ;*CABLE TURNAROUND TEST
                                        ;*CLEAR LINE UNIT LOOP, SET DTR
                                        ;*VERIFY THAT MODEM READY IS SET
                                        ;*CLEAR DTR, VERIFY THAT MRDY IS CLEARED
                                        ;;*************************************************************

                                        ;  TEST 40
                                        ;-------------
017304  012737  000040  001226  TST40:  MOV     #40,TSTNO
017312  012737  017466  001216          MOV     #TST41,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
017320  104412                          MSTCLR                  ;MASTER CLEAR DMC11
017322  032737  020000  001366          BIT     #BIT13,STAT1    ;IS LINE UNIT M8202?
017330  001004                          BNE     .+12            ;BR IF YES (DO TEST EVEN IF NO LOOP-BACK CONN)
017332  032737  040000  001366          BIT     #BIT14,STAT1    ;IS TURNAROUND CONNECTOR ON?
017340  001451                          BEQ     2$              ;SKIP TEST IF NO
017342  005011                          CLR     (R1)            ;CLEAR LINE UNIT LOOP
017344  012761  000100  000004          MOV     #100,4(R1)      ;LOAD PORT4
```

```
017352  104414                              ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017354  122113                              122113                          ;SET DTR
017356  104416  000002                      TIMER,   2                      ;WAIT
017362  104414                              ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017364  021264                              021264                          ;PORT4←LU13
017366  016104  000004                      MOV      4(R1),R4               ;PUT "FOUND" IN R4
017372  042704  000223                      BIC      #223,R4                ;CLEAR UNWANTED BITS
017376  012705  000110                      MOV      #110,R5                ;PUT "EXPECTED" IN R5
017402  120504                              CMPB     R5,R4                  ;IS MRDY SET?
017404  001401                              BEQ      1$
017406  104011                              HLT      11                     ;ERROR, MRDY NOT SET
017410  005061  000004              1$:     CLR      4(R1)                  ;CLEAR PORT4
017414  104414                              ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017416  122113                              122113                          ;CLEAR DTR
017420  104416  000002                      TIMER,   2
017424  104414                              ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017426  021264                              021264                          ;PORT4←LU13
017430  016104  000004                      MOV      4(R1),R4               ;PUT "FOUND" IN R4
017434  042704  000223                      BIC      #223,R4                ;CLEAR UNWANTED BITS
017440  005005                              CLR      R5                     ;PUT "EXPECTED" IN R5
017442  032737  020000  001366              BIT      #BIT13,STAT1           ;IS LINE UNIT M8202?
017450  001402                              BEQ      .+6                    ;BR IF NO
017452  052705  000010                      BIS      #BIT3,R5               ;MRDY SET ON M8202
017456  120504                              CMPB     R5,R4                  ;IS MRDY CLEAR?
017460  001401                              BEQ      2$
017462  104011                              HLT      11                     ;ERROR, MRDY NOT CLEAR
017464  104400                      2$:     SCOPE                           ;SCOPE THIS TEST



                                            ;************************** TEST 41 **************************
                                            ;*CABLE TURNAROUND TEST
                                            ;*CLEAR LINE UNIT LOOP, LOAD OUT DATA SILO
                                            ;*VERIFY THAT ALL MODEM SIGNALS ARE SET
                                            ;:************************************************************

                                            ;  TEST 41
                                            ;--------------
017466  012737  000041  001226      TST41:  MOV      #41,TSTNO
017474  012737  017632  001216              MOV      #TST42,NEXT
                                                                            ;R1 CONTAINS BASE DMC11 ADDRESS
017502  104412                              MSTCLR                          ;MASTER CLEAR DMC11
017504  032737  020000  001366              BIT      #BIT13,STAT1           ;IS LINE UNIT M8202?
017512  001004                              BNE      .+12                   ;BR IF YES (DO TEST EVEN IF NO LOOP-BACK CONN)
017514  032737  040000  001366              BIT      #BIT14,STAT1           ;IS TURNAROUND CONNECTOR ON?
017522  001442                              BEQ      1$                     ;SKIP TEST IF NO
017524  012711  004000                      MOV      #BIT11,(R1)            ;SET LINE UNIT LOOP
017530  012761  000100  000004              MOV      #100,    4(R1)                  ;LOAD PORT4
017536  104414                              ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017540  122113                              122113                          ;CLEAR ALL MODEM SIGNALS,EXCEPT DTR
017542  104416  000002                      TIMER,   2                      ;WAIT
017546  012761  000001  000004              MOV      #1,4(R1)               ;LOAD PORT4
017554  104414                              ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017556  122111                              122111                          ;SET SOM
017560  004537  027620                      JSR      R5,MESLD               ;FILL OUT DATA SILO
017564  030102                              MESDAT                          ;WITH 64 CHARACTERS
017566  000100                              64.
```

```
017570  012700  000050                   MOV     #50,R0          ;PREPARE FOR DELAY
017574  005011                           CLR     (R1)            ;CLEAR LINE UNIT LOOP
017576                           2$:
017576  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017600  021264                           021264                  ;PORT4←LU13
017602  016104  000004                   MOV     4(R1),R4        ;PUT "FOUND" IN R4
017606  042704  000223                   BIC     #223,R4         ;CLEAR UNWANTED BITS
017612  012705  000154                   MOV     #154,R5         ;PUT "EXPECTED" IN R5
017616  120504                           CMPB    R5,R4           ;COMPARE EXPECTED AND FOUND
017620  001403                           BEQ     1$              ;BR IF OK
017622  005300                           DEC     R0              ;DEC DELAY COUNT
017624  001364                           BNE     2$              ;BR IF NOT ZERO
017626  104011                           HLT     11              ;ERROR, ALL SIGNALS ARE NOT SET
017630  104400                   1$:     SCOPE                   ;SCOPE THIS TEST



                                         ;*************************** TEST 42 **************************
                                         ;*TEST OF CRC OPERATION
                                         ;*USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
                                         ;*0, VERIFY THE LSB OF THE BCC ON EACH SHIFT
                                         ;*TEST TRANSMITTER FIRST THEN THE RECEIVER BCC
                                         ;;*************************************************************


                                         ;  TEST 42
                                         ;--------------
017632  012737  000042  001226   TST42:  MOV     #42,TSTNO
017640  012737  020146  001216           MOV     #TST43,NEXT
017646  012737  017662  001220           MOV     #64$,LOCK
                                                                 ;R1 CONTAINS BASE DMC11 ADDRESS
017654  104412                           MSTCLR                  ;MASTER CLEAR DMC11
017656  012711  004000                   MOV     #BIT11,(R1)     ;SET LU LOOP
017662  004737  027662           64$:    JSR     PC,CLRIO        ;CLEAR BCC REGISTERS
017666  005000                           CLR     R0              ;START SHIFT COUNTER AT ZERO
017670  012737  120001  027316           MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL FOR SOFTWARE BCC
017676  012737  000000  017736           MOV     #0,66$          ;LOAD CHAR FOR SOFTWARE BCC
017704  005037  017740                   CLR     67$             ;CLEAR OLD SOFTWARE BCC
017710  004737  027322                   JSR     PC,BCCLD        ;LOAD OUT SILO WITH 2 SYNCS
017714  000000                           0                       ;AND THE CHARACTER 0
017716  104415  000021                   DATACLK,        21      ;GET TRANSMITTER ACTIVE
017722  104415  000001           65$:    DATACLK,        1       ;SHIFT BCC ONCE
017726  005200                           INC     R0              ;BUMP SHIFT COUNT
017730  004537  027212                   JSR     R5,SIMBCC       ;CALCULATE SOFTWARE BCC LSB
017734  000001                           1                       ;ONE SHIFT
017736  000000                   66$:    0                       ;DATA CHARACTER
017740  000000                   67$:    0                       ;OLD BCC
017742  103405                           BCS     68$             ;BR IF SOFT BCC LSB IS SET
017744  004737  027434                   JSR     PC,GETQ0        ;GET HARDWARE TRANSMITTER BCC LSB
017750  103006                           BCC     69$             ;BR IF HARD BCC LSB IS CLEAR
017752  104012                           HLT     12              ;ERROR, BCC LSB IS SET
017754  000404                           BR      69$             ;CONTINUE
017756  004737  027434           68$:    JSR     PC,GETQ0        ;GET HARDWARE TRANSMITTER BCC LSB
017762  103401                           BCS     69$             ;BR IF HARD BCC LSB IS SET
017764  104016                           HLT     16              ;ERROR, HARD BCC LSB IS CLEAR
017766                           69$:
017766  006037  017736                   ROR     66$             ;SHIFT SOFT DATA
017772  013737  027320  017740           MOV     CALBCC,67$      ;LOAD OLD SOFT BCC
```

```
020000  022700  000010                      CMP     #10,R0          ;DONE YET?
020004  001346                              BNE     65$             ;BR IF NOT DONE
020006  104401                              SCOP1                   ;SCOPE SUBTEST (SW09=1)
020010  012737  020016  001220              MOV     #71$,LOCK       ;NEW SCOPE1
020016  004737  027662              71$:    JSR     PC,CLRIO        ;CLEAR BCC REGISTERS
020022  005000                              CLR     R0              ;START SHIFT COUNTER AT ZERO
020024  012737  120001  027316              MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL FOR SOFTWARE BCC
020032  012737  000000  020072              MOV     #0,73$          ;LOAD CHAR FOR SOFTWARE BCC
020040  005037  020074                      CLR     74$             ;CLEAR OLD SOFTWARE BCC
020044  004737  027322                      JSR     PC,BCCLD        ;LOAD OUT SILO WITH 2 SYNCS
020050  000000                              0                       ;AND THE CHARACTER 0
020052  104415  000032                      DATACLK,        32      ;GET RECEIVER ACTIVE
020056  104415  000001              72$:    DATACLK,        1       ;SHIFT BCC ONCE
020062  005200                              INC     R0              ;BUMP SHIFT COUNT
020064  004537  027212                      JSR     R5,SIMBCC       ;CALCULATE SOFTWARE BCC LSB
020070  000001                              1                       ;ONE SHIFT
020072  000000                      73$:    0                       ;DATA CHARACTER
020074  000000                      74$:    0                       ;OLD BCC
020076  103405                              BCS     75$             ;BR IF SOFT BCC LSB IS SET
020100  004737  027446                      JSR     PC,GETQI        ;GET HARDWARE RECEIVER BCC LSB
020104  103006                              BCC     76$             ;BR IF HARD BCC LSB IS CLEAR
020106  104013                              HLT     13              ;ERROR, BCC LSB IS SET
020110  000404                              BR      76$             ;CONTINUE
020112  004737  027446              75$:    JSR     PC,GETQI        ;GET HARDWARE RECEIVER BCC LSB
020116  103401                              BCS     76$             ;BR IF HARD BCC LSB IS SET
020120  104017                              HLT     17              ;ERROR, BCC LSB IS CLEAR
020122                              76$:
020122  006037  020072                      ROR     73$             ;SHIFT SOFT DATA
020126  013737  027320  020074              MOV     CALBCC,74$      ;LOAD OLD SOFT BCC
020134  022700  000010                      CMP     #10,R0          ;DONE YET?
020140  001346                              BNE     72$             ;BR IF NOT DONE
020142  104401                              SCOP1                   ;SCOPE SUBTEST (SW09=1)
020144  104400                      77$:    SCOPE                   ;SCOPE THIS TEST



                                            ;*************************** TEST 43 ***************************
                                            ;*TEST OF CRC OPERATION
                                            ;*USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
                                            ;*377, VERIFY THE LSB OF THE BCC ON EACH SHIFT
                                            ;*TEST TRANSMITTER FIRST THEN THE RECEIVER BCC
                                            ;:**************************************************************


                                            ;  TEST 43
                                            ;--------------
020146  012737  000043  001226      TST43:  MOV     #43,TSTNO
020154  012737  020462  001216              MOV     #TST44,NEXT
020162  012737  020176  001220              MOV     #64$,LOCK
                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
020170  104412                              MSTCLR                  ;MASTER CLEAR DMC11
020172  012711  004000                      MOV     #BIT11,(R1)     ;SET LU LOOP
020176  004737  027662              64$:    JSR     PC,CLRIO        ;CLEAR BCC REGISTERS
020202  005000                              CLR     R0              ;START SHIFT COUNTER AT ZERO
020204  012737  120001  027316              MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL FOR SOFTWARE BCC
020212  012737  000377  020252              MOV     #377,66$;       ;LOAD CHAR FOR SOFTWARE BCC
020220  005037  020254                      CLR     67$             ;CLEAR OLD SOFTWARE BCC
020224  004737  027322                      JSR     PC,BCCLD        ;LOAD OUT SILO WITH 2 SYNCS
```

```
020230  000377                                   377                         ;AND THE CHARACTER 377
020232  104415  000021                           DATACLK,        21          ;GET TRANSMITTER ACTIVE
020236  104415  000001                   65$:    DATACLK,        1           ;SHIFT BCC ONCE
020242  005200                                   INC     R0                  ;BUMP SHIFT COUNT
020244  004537  027212                           JSR     R5,SIMBCC           ;CALCULATE SOFTWARE BCC LSB
020250  000001                                   1                           ;ONE SHIFT
020252  000000                           66$:    0                           ;DATA CHARACTER
020254  000000                           67$:    0                           ;OLD BCC
020256  103405                                   BCS     68$                 ;BR IF SOFT BCC LSB IS SET
020260  004737  027434                           JSR     PC,GETQ0            ;GET HARDWARE TRANSMITTER BCC LSB
020264  103006                                   BCC     69$                 ;BR IF HARD BCC LSB IS CLEAR
020266  104012                                   HLT     12                  ;ERROR, BCC LSB IS SET
020270  000404                                   BR      69$                 ;CONTINUE
020272  004737  027434                   68$:    JSR     PC,GETQ0            ;GET HARDWARE TRANSMITTER BCC LSB
020276  103401                                   BCS     69$                 ;BR IF HARD BCC LSB IS SET
020300  104016                                   HLT     16                  ;ERROR, HARD BCC LSB IS CLEAR
020302                                   69$:
020302  006037  020252                           ROR     66$                 ;SHIFT SOFT DATA
020306  013737  027320  020254                   MOV     CALBCC,67$          ;LOAD OLD SOFT BCC
020314  022700  000010                           CMP     #10,R0              ;DONE YET?
020320  001346                                   BNE     65$                 ;BR IF NOT DONE
020322  104401                                   SCOP1                       ;SCOPE SUBTEST (SW09=1)
020324  012737  020332  001220                   MOV     #71$,LOCK           ;NEW SCOPE1
020332  004737  027662                   71$:    JSR     PC,CLRIO            ;CLEAR BCC REGISTERS
020336  005000                                   CLR     R0                  ;START SHIFT COUNTER AT ZERO
020340  012737  120001  027316                   MOV     #CRC16,XPOLY        ;LOAD POLYNOMIAL FOR SOFTWARE BCC
020346  012737  000377  020406                   MOV     #377,73$;           ;LOAD CHAR FOR SOFTWARE BCC
020354  005037  020410                           CLR     74$                 ;CLEAR OLD SOFTWARE BCC
020360  004737  027322                           JSR     PC,BCCLD            ;LOAD OUT SILO WITH 2 SYNCS
020364  000377                                   377                         ;AND THE CHARACTER 377
020366  104415  000032                           DATACLK,        32          ;GET RECEIVER ACTIVE
020372  104415  000001                   72$:    DATACLK,        1           ;SHIFT BCC ONCE
020376  005200                                   INC     R0                  ;BUMP SHIFT COUNT
020400  004537  027212                           JSR     R5,SIMBCC           ;CALCULATE SOFTWARE BCC LSB
020404  000001                                   1                           ;ONE SHIFT
020406  000000                           73$:    0                           ;DATA CHARACTER
020410  000000                           74$:    0                           ;OLD BCC
020412  103405                                   BCS     75$                 ;BR IF SOFT BCC LSB IS SET
020414  004737  027446                           JSR     PC,GETQI            ;GET HARDWARE RECEIVER BCC LSB
020420  103006                                   BCC     76$                 ;BR IF HARD BCC LSB IS CLEAR
020422  104013                                   HLT     13                  ;ERROR, BCC LSB IS SET
020424  000404                                   BR      76$                 ;CONTINUE
020426  004737  027446                   75$:    JSR     PC,GETQI            ;GET HARDWARE RECEIVER BCC LSB
020432  103401                                   BCS     76$                 ;BR IF HARD BCC LSB IS SET
020434  104017                                   HLT     17                  ;ERROR, BCC LSB IS CLEAR
020436                                   76$:
020436  006037  020406                           ROR     73$                 ;SHIFT SOFT DATA
020442  013737  027320  020410                   MOV     CALBCC,74$          ;LOAD OLD SOFT BCC
020450  022700  000010                           CMP     #10,R0              ;DONE YET?
020454  001346                                   BNE     72$                 ;BR IF NOT DONE
020456  104401                                   SCOP1                       ;SCOPE SUBTEST (SW09=1)
020460  104400                           77$:    SCOPE                       ;SCOPE THIS TEST


                                                 ;*************************** TEST 44 ***************************
                                                 ;*TEST OF CRC OPERATION
```

```
                                                        ;*USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
                                                        ;*125, VERIFY THE LSB OF THE BCC ON EACH SHIFT
                                                        ;*TEST TRANSMITTER FIRST THEN THE RECEIVER BCC
                                                        ;;*************************************************************

                                                        ;  TEST 44
                                                        ;--------------
 020462  012737  000044  001226                TST44:   MOV     #44,TSTNO
 020470  012737  020776  001216                         MOV     #TST45,NEXT
 020476  012737  020512  001220                         MOV     #64$,LOCK
                                                                                   ;R1 CONTAINS BASE DMC11 ADDRESS
 020504  104412                                         MSTCLR                     ;MASTER CLEAR DMC11
 020506  012711  004000                                 MOV     #BIT11,(R1)        ;SET LU LOOP
 020512  004737  027662                        64$:     JSR     PC,CLRIO           ;CLEAR BCC REGISTERS
 020516  005000                                         CLR     R0                 ;START SHIFT COUNTER AT ZERO
 020520  012737  120001  027316                         MOV     #CRC16,XPOLY       ;LOAD POLYNOMIAL FOR SOFTWARE BCC
 020526  012737  000125  020566                         MOV     #125,66$;          ;LOAD CHAR FOR SOFTWARE BCC
 020534  005037  020570                                 CLR     67$                ;CLEAR OLD SOFTWARE BCC
 020540  004737  027322                                 JSR     PC,BCCLD           ;LOAD OUT SILO WITH 2 SYNCS
 020544  000125                                         125                        ;AND THE CHARACTER 125
 020546  104415  000021                                 DATACLK,        21         ;GET TRANSMITTER ACTIVE
 020552  104415  000001                        65$:     DATACLK,        1          ;SHIFT BCC ONCE
 020556  005200                                         INC     R0                 ;BUMP SHIFT COUNT
 020560  004537  027212                                 JSR     R5,SIMBCC          ;CALCULATE SOFTWARE BCC LSB
 020564  000001                                         1                          ;ONE SHIFT
 020566  000000                                66$:     0                          ;DATA CHARACTER
 020570  000000                                67$:     0                          ;OLD BCC
 020572  103405                                         BCS     68$                ;BR IF SOFT BCC LSB IS SET
 020574  004737  027434                                 JSR     PC,GETQO           ;GET HARDWARE TRANSMITTER BCC LSB
 020600  103006                                         BCC     69$                ;BR IF HARD BCC LSB IS CLEAR
 020602  104012                                         HLT     12                 ;ERROR, BCC LSB IS SET
 020604  000404                                         BR      69$                ;CONTINUE
 020606  004737  027434                        68$:     JSR     PC,GETQO           ;GET HARDWARE TRANSMITTER BCC LSB
 020612  103401                                         BCS     69$                ;BR IF HARD BCC LSB IS SET
 020614  104016                                         HLT     16                 ;ERROR, HARD BCC LSB IS CLEAR
 020616                                        69$:
 020616  006037  020566                                 ROR     66$                ;SHIFT SOFT DATA
 020622  013737  027320  020570                         MOV     CALBCC,67$         ;LOAD OLD SOFT BCC
 020630  022700  000010                                 CMP     #10,R0             ;DONE YET?
 020634  001346                                         BNE     65$                ;BR IF NOT DONE
 020636  104401                                         SCOP1                      ;SCOPE SUBTEST (SW09=1)
 020640  012737  020646  001220                         MOV     #71$,LOCK          ;NEW SCOPE1
 020646  004737  027662                        71$:     JSR     PC,CLRIO           ;CLEAR BCC REGISTERS
 020652  005000                                         CLR     R0                 ;START SHIFT COUNTER AT ZERO
 020654  012737  120001  027316                         MOV     #CRC16,XPOLY       ;LOAD POLYNOMIAL FOR SOFTWARE BCC
 020662  012737  000125  020722                         MOV     #125,73$;          ;LOAD CHAR FOR SOFTWARE BCC
 020670  005037  020724                                 CLR     74$                ;CLEAR OLD SOFTWARE BCC
 020674  004737  027322                                 JSR     PC,BCCLD           ;LOAD OUT SILO WITH 2 SYNCS
 020700  000125                                         125                        ;AND THE CHARACTER 125
 020702  104415  000032                                 DATACLK,        32         ;GET RECEIVER ACTIVE
 020706  104415  000001                        72$:     DATACLK,        1          ;SHIFT BCC ONCE
 020712  005200                                         INC     R0                 ;BUMP SHIFT COUNT
 020714  004537  027212                                 JSR     R5,SIMBCC          ;CALCULATE SOFTWARE BCC LSB
 020720  000001                                         1                          ;ONE SHIFT
 020722  000000                                73$:     0                          ;DATA CHARACTER
 020724  000000                                74$:     0                          ;OLD BCC
```

```
020726  103405                            BCS     75$                     ;BR IF SOFT BCC LSB IS SET
020730  004737  027446                    JSR     PC,GETQI                ;GET HARDWARE RECEIVER BCC LSB
020734  103006                            BCC     76$                     ;BR IF HARD BCC LSB IS CLEAR
020736  104013                            HLT     13                      ;ERROR, BCC LSB IS SET
020740  000404                            BR      76$                     ;CONTINUE
020742  004737  027446            75$:    JSR     PC,GETQI                ;GET HARDWARE RECEIVER BCC LSB
020746  103401                            BCS     76$                     ;BR IF HARD BCC LSB IS SET
020750  104017                            HLT     17                      ;ERROR, BCC LSB IS CLEAR
020752                            76$:
020752  006037  020722                    ROR     73$                     ;SHIFT SOFT DATA
020756  013737  027320  020724            MOV     CALBCC,74$              ;LOAD OLD SOFT BCC
020764  022700  000010                    CMP     #10,R0                  ;DONE YET?
020770  001346                            BNE     72$                     ;BR IF NOT DONE
020772  104401                            SCOP1                           ;SCOPE SUBTEST (SW09=1)
020774  104400                    77$:    SCOPE                           ;SCOPE THIS TEST



                                          ;*************************** TEST 45 ***************************
                                          ;*TEST OF CRC OPERATION
                                          ;*USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
                                          ;*252, VERIFY THE LSB OF THE BCC ON EACH SHIFT
                                          ;*TEST TRANSMITTER FIRST THEN THE RECEIVER BCC
                                          ;;*************************************************************

                                          ;  TEST 45
                                          ;--------------
020776  012737  000045  001226    TST45:  MOV     #45,TSTNO
021004  012737  021312  001216            MOV     #TST46,NEXT
021012  012737  021026  001220            MOV     #64$,LOCK
                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
021020  104412                            MSTCLR                          ;MASTER CLEAR DMC11
021022  012711  004000                    MOV     #BIT11,(R1)             ;SET LU LOOP
021026  004737  027662            64$:    JSR     PC,CLRIO                ;CLEAR BCC REGISTERS
021032  005000                            CLR     R0                      ;START SHIFT COUNTER AT ZERO
021034  012737  120001  027316            MOV     #CRC16,XPOLY            ;LOAD POLYNOMIAL FOR SOFTWARE BCC
021042  012737  000252  021102            MOV     #252,66$;               ;LOAD CHAR FOR SOFTWARE BCC
021050  005037  021104                    CLR     67$                     ;CLEAR OLD SOFTWARE BCC
021054  004737  027322                    JSR     PC,BCCLD                ;LOAD OUT SILO WITH 2 SYNCS
021060  000252                            252                             ;AND THE CHARACTER 252
021062  104415  000021                    DATACLK,        21              ;GET TRANSMITTER ACTIVE
021066  104415  000001            65$:    DATACLK,        1               ;SHIFT BCC ONCE
021072  005200                            INC     R0                      ;BUMP SHIFT COUNT
021074  004537  027212                    JSR     R5,SIMBCC               ;CALCULATE SOFTWARE BCC LSB
021100  000001                            1                               ;ONE SHIFT
021102  000000                    66$:    0                               ;DATA CHARACTER
021104  000000                    67$:    0                               ;OLD BCC
021106  103405                            BCS     68$                     ;BR IF SOFT BCC LSB IS SET
021110  004737  027434                    JSR     PC,GETQO                ;GET HARDWARE TRANSMITTER BCC LSB
021114  103006                            BCC     69$                     ;BR IF HARD BCC LSB IS CLEAR
021116  104012                            HLT     12                      ;ERROR, BCC LSB IS SET
021120  000404                            BR      69$                     ;CONTINUE
021122  004737  027434            68$:    JSR     PC,GETQO                ;GET HARDWARE TRANSMITTER BCC LSB
021126  103401                            BCS     69$                     ;BR IF HARD BCC LSB IS SET
021130  104016                            HLT     16                      ;ERROR, HARD BCC LSB IS CLEAR
021132                            69$:
021132  006037  021102                    ROR     66$                     ;SHIFT SOFT DATA
```

```
021136  013737  027320  021104                MOV     CALBCC,67$          ;LOAD OLD SOFT BCC
021144  022700  000010                        CMP     #10,R0              ;DONE YET?
021150  001346                                BNE     65$                 ;BR IF NOT DONE
021152  104401                                SCOP1                       ;SCOPE SUBTEST (SW09=1)
021154  012737  021162  001220                MOV     #71$,LOCK           ;NEW SCOPE1
021162  004737  027662                71$:    JSR     PC,CLRIO            ;CLEAR BCC REGISTERS
021166  005000                                CLR     R0                  ;START SHIFT COUNTER AT ZERO
021170  012737  120001  027316                MOV     #CRC16,XPOLY        ;LOAD POLYNOMIAL FOR SOFTWARE BCC
021176  012737  000252  021236                MOV     #252,73$;           ;LOAD CHAR FOR SOFTWARE BCC
021204  005037  021240                        CLR     74$                 ;CLEAR OLD SOFTWARE BCC
021210  004737  027322                        JSR     PC,BCCLD            ;LOAD OUT SILO WITH 2 SYNCS
021214  000252                                252                         ;AND THE CHARACTER 252
021216  104415  000032                        DATACLK,        32          ;GET RECEIVER ACTIVE
021222  104415  000001                72$:    DATACLK,        1           ;SHIFT BCC ONCE
021226  005200                                INC     R0                  ;BUMP SHIFT COUNT
021230  004537  027212                        JSR     R5,SIMBCC           ;CALCULATE SOFTWARE BCC LSB
021234  000001                                1                           ;ONE SHIFT
021236  000000                        73$:    0                           ;DATA CHARACTER
021240  000000                        74$:    0                           ;OLD BCC
021242  103405                                BCS     75$                 ;BR IF SOFT BCC LSB IS SET
021244  004737  027446                        JSR     PC,GETQI            ;GET HARDWARE RECEIVER BCC LSB
021250  103006                                BCC     76$                 ;BR IF HARD BCC LSB IS CLEAR
021252  104013                                HLT     13                  ;ERROR, BCC LSB IS SET
021254  000404                                BR      76$                 ;CONTINUE
021256  004737  027446                75$:    JSR     PC,GETQI            ;GET HARDWARE RECEIVER BCC LSB
021262  103401                                BCS     76$                 ;BR IF HARD BCC LSB IS SET
021264  104017                                HLT     17                  ;ERROR, BCC LSB IS CLEAR
021266                                76$:
021266  006037  021236                        ROR     73$                 ;SHIFT SOFT DATA
021272  013737  027320  021240                MOV     CALBCC,74$          ;LOAD OLD SOFT BCC
021300  022700  000010                        CMP     #10,R0              ;DONE YET?
021304  001346                                BNE     72$                 ;BR IF NOT DONE
021306  104401                                SCOP1                       ;SCOPE SUBTEST (SW09=1)
021310  104400                        77$:    SCOPE                       ;SCOPE THIS TEST



                                              ;************************* TEST 46 *************************
                                              ;*TRANSMITTER CRC TEST
                                              ;*USING THE CRC16 POLYNOMINAL, SINGLE CLOCK A BINARY
                                              ;*COUNT PATTERN, VERIFY THE LSB OF THE TRANSMITTER BCC ON EACH SHIFT
                                              ;:*********************************************************


                                              ;  TEST 46
                                              ;--------------
021312  012737  000046  001226        TST46:  MOV     #46,TSTNO
021320  012737  021550  001216                MOV     #TST47,NEXT
                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
021326  104412                                MSTCLR                      ;MASTER CLEAR DMC11
021330  012711  004000                        MOV     #BIT11,(R1)         ;SET LINE UNIT LOOP
021334  005003                                CLR     R3                  ;ZERO BIT COUNT
021336  005004                                CLR     R4                  ;R4 CONTAINS CHAR TO BE LOADED IN SILO
021340  005005                                CLR     R5                  ;R5 CONTAINS CHAR CURRENTLY BEING SHIFTED OUT
021342  005037  021444                        CLR     4$                  ;CLEAR SOFT BCC
021346  012737  120001  027316                MOV     #CRC16,XPOLY        ;LOAD POLYNOMINAL
021354  004737  027464                        JSR     PC,SYNLD            ;LOAD SILO WITH 2 SYNCS, SOM SET
021360  010461  000004                        MOV     R4,4(R1)            ;PORT4←CHAR
```

```
021364  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021366  122110                        122110                  ;LOAD OUT DATA
021370  005204                        INC     R4              ;INCREMENT TO NEXT CHARACTER
021372  010461  000004                MOV     R4,4(R1)        ;PORT4←CHAR
021376  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021400  122110                        122110                  ;LOAD OUT DATA
021402  005204                        INC     R4              ;INCREMENT TO NEXT CHARACTER
021404  010461  000004                MOV     R4,4(R1)        ;PORT4←CHAR
021410  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021412  122110                        122110                  ;LOAD OUT DATA
021414  004737  026350                JSR     PC,OCOR         ;WAIT FOR OCOR
021420  104415  000021                DATACLK,21              ;CLOCK DATA
021424  010537  021442        1$:     MOV     R5,3$           ;LOAD CHAR FOR SOFT CRC
021430  104415  000001        2$:     DATACLK,1               ;SHIFT BCC ONCE
021434  004537  027212                JSR     R5,SIMBCC       ;CALCULATE SOFT BCC
021440  000001                        1                       ;SOFT SHIFT COUNT
021442  000000                3$:     0                       ;SOFT CHARACTER
021444  000000                4$:     0                       ;OLD SOFT BCC
021446  103405                        BCS     5$              ;BR IF SOFT BCC LSB IS SET
021450  004737  027434                JSR     PC,GETQ0                ;GET HARDWARE TRANSMITTER BCC LSB
021454  103006                        BCC     6$              ;BR IF OK (CLEARED)
021456  104020                        HLT     20              ;ERROR, BCC LSB WAS SET
021460  000404                        BR      6$              ;CONTINUE WITH TEST
021462  004737  027434        5$:     JSR     PC,GETQ0                ;GET HARDWARE TRANSMITTER BCC LSB
021466  103401                        BCS     6$              ;BR IF OK (SET)
021470  104021                        HLT     21              ;ERROR, BCC LSB WAS CLEAR

021472                        6$:
021472  006037  021442                ROR     3$              ;SHIFT SOFT DATA
021476  013737  027320  021444        MOV     CALBCC,4$       ;LOAD OLD SOFT BCC
021504  005203                        INC     R3              ;INCREMENT BIT COUNTER
021506  022703  000010                CMP     #10,R3          ;DONE A FULL CHARACTER YET?
021512  001346                        BNE     2$              ;BR IF NO
021514  005003                        CLR     R3              ;RESTART BIT COUNTER
021516  005204                        INC     R4              ;INCREMENT DATA FOR SILO
021520  022704  000400                CMP     #400,R4         ;DONE BINARY COUNT YET?
021524  003404                        BLE     9$              ;BR IF YES
021526  010461  000004                MOV     R4,4(R1)        ;PORT4←DATA
021532  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021534  122110                        122110                  ;LOAD OUT DATA
021536  005205                9$:     INC     R5              ;INCREMENT DATA
021540  022705  000400                CMP     #400,R5         ;DONE BINARY PATTERN YET?
021544  001327                        BNE     1$              ;BR IF NO
021546  104400                7$:     SCOPE                   ;SCOPE THIS TEST


                                      ;**************************** TEST 47 ****************************
                                      ;*RECEIVER CRC TEST
                                      ;*USING THE CRC16 POLYNOMINAL, SINGLE CLOCK A BINARY
                                      ;*COUNT PATTERN, VERIFY THE LSB OF THE RECEIVER BCC ON EACH SHIFT
                                      ;:*****************************************************************

                                      ;  TEST 47
                                      ;  --------------
021550  012737  000047  001226  TST47: MOV    #47,TSTNO
021556  012737  022006  001216         MOV    #TST50,NEXT
```

```
                                                                     ;R1 CONTAINS BASE DMC11 ADDRESS
021564  104412                                    MSTCLR             ;MASTER CLEAR DMC11
021566  012711  004000                            MOV     #BIT11,(R1) ;SET LINE UNIT LOOP
021572  005003                                    CLR     R3         ;ZERO BIT COUNT
021574  005004                                    CLR     R4         ;R4 CONTAINS CHAR TO BE LOADED IN SILO
021576  005005                                    CLR     R5         ;R5 CONTAINS CHAR CURRENTLY BEING SHIFTED OUT
021600  005037  021702                            CLR     4$         ;CLEAR SOFT BCC
021604  012737  120001  027316                    MOV     #CRC16,XPOLY ;LOAD POLYNOMINAL
021612  004737  027464                            JSR     PC,SYNLD   ;LOAD SILO WITH 2 SYNCS, SOM SET
021616  010461  000004                            MOV     R4,4(R1)   ;PORT4+CHAR
021622  104414                                    ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021624  122110                                    122110             ;LOAD OUT DATA
021626  005204                                    INC     R4         ;INCREMENT TO NEXT CHARACTER
021630  010461  000004                            MOV     R4,4(R1)   ;PORT4+CHAR
021634  104414                                    ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021636  122110                                    122110             ;LOAD OUT DATA
021640  005204                                    INC     R4         ;INCREMENT TO NEXT CHARACTER
021642  010461  000004                            MOV     R4,4(R1)   ;PORT4+CHAR
021646  104414                                    ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021650  122110                                    122110             ;LOAD OUT DATA
021652  004737  026350                            JSR     PC,OCOR    ;WAIT FOR OCOR
021656  104415  000032                            DATACLK,32         ;CLOCK DATA
021662  010537  021700                    1$:     MOV     R5,3$      ;LOAD CHAR FOR SOFT CRC
021666  104415  000001                    2$:     DATACLK,1          ;SHIFT BCC ONCE
021672  004537  027212                            JSR     R5,SIMBCC  ;CALCULATE SOFT BCC
021676  000001                                    1                  ;SOFT SHIFT COUNT
021700  000000                            3$:     0                  ;SOFT CHARACTER
021702  000000                            4$:     0                  ;OLD SOFT BCC
021704  103405                                    BCS     5$         ;BR IF SOFT BCC LSB IS SET
021706  004737  027446                            JSR     PC,GETQI   ;GET HARDWARE RECEIVER BCC LSB
021712  103006                                    BCC     6$         ;BR IF OK (CLEARED)
021714  104022                                    HLT     22         ;ERROR, BCC LSB WAS SET
021716  000404                                    BR      6$         ;CONTINUE WITH TEST
021720  004737  027446                    5$:     JSR     PC,GETQI   ;GET HARDWARE RECEIVER BCC LSB
021724  103401                                    BCS     6$         ;BR IF OK (SET)
021726  104023                                    HLT     23         ;ERROR, BCC LSB WAS CLEAR

021730                                    6$:
021730  006037  021700                            ROR     3$         ;SHIFT SOFT DATA
021734  013737  027320  021702                    MOV     CALBCC,4$  ;LOAD OLD SOFT BCC
021742  005203                                    INC     R3         ;INCREMENT BIT COUNTER
021744  022703  000010                            CMP     #10,R3     ;DONE A FULL CHARACTER YET?
021750  001346                                    BNE     2$         ;BR IF NO
021752  005003                                    CLR     R3         ;RESTART BIT COUNTER
021754  005204                                    INC     R4         ;INCREMENT DATA FOR SILO
021756  022704  000400                            CMP     #400,R4    ;DONE BINARY COUNT YET?
021762  003404                                    BLE     9$         ;BR IF YES
021764  010461  000004                            MOV     R4,4(R1)   ;PORT4+DATA
021770  104414                                    ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021772  122110                                    122110             ;LOAD OUT DATA
021774  005205                            9$:     INC     R5         ;INCREMENT DATA
021776  022705  000400                            CMP     #400,R5    ;DONE BINARY PATTERN YET?
022002  001327                                    BNE     1$         ;BR IF NO
022004  104400                            7$:     SCOPE              ;SCOPE THIS TEST
```

```
                                                      ;************************** TEST 50 ***************************
                                                      ;*TRANSMITTER DDCMP CRC TEST
                                                      ;*THIS TEST TRANSMITS A FOUR CHARACTER MESSAGE WITH CRC
                                                      ;*BOTH DATA AND THE BCC ARE VERIFIED IN THE BIT
                                                      ;*WINDOW. THE FOUR CHARACTERS ARE 0,125,252,377
                                                      ;*THE TRANSMITTER IS CHECKED FOR GOING TO A MARK STATE AFTER THE BCC
                                                      ;;************************************************************

                                                      ;  TEST 50
                                                      ;--------------
022006  012737  000050  001226              TST50:  MOV     #50,TSTNO
022014  012737  022340  001216                      MOV     #TST51,NEXT
                                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
022022  104412                                      MSTCLR                      ;MASTER CLEAR DMC11

                                                      ;LOAD OUT DATA SILO

022024  012711  004000                              MOV     #BIT11,(R1)         ;SET LINE UNIT LOOP
022030  012704  030102                              MOV     #MESDAT,R4          ;LOAD POINTER TO DATA
022034  005037  022130                              CLR     10$                 ;CLEAR SOFT BCC
022040  012700  000004                              MOV     #4,R0               ;LOAD CHARACTER COUNT
022044  004737  027464                              JSR     PC,SYNLD            ;LOAD 2 SYNCS IN OUT SILO
022050  004737  026502                              JSR     PC,OUTRDY           ;WAIT FOR OUTRDY
022054  004537  027620                              JSR     R5,MESLD            ;LOAD SILO WITH 4 CHAR MESS
022060  030102                                      MESDAT                      ;ADDRESS OF MESSAGE
022062  000004                                      4                           ;NUMBER OF CHARACTERS
022064  004737  027574                              JSR     PC,EOM              ;LOAD GARBAGE CHARACTER, WITH EOM SET
022070  004737  026350                              JSR     PC,OCOR             ;WAIT FOR OCOR
022074  005003                                      CLR     R3                  ;CLEAR BIT COUNTER
022076  104415  000022                              DATACLK,22                  ;CLOCK DATA
022102  112405                              12$:    MOVB    (R4)+,R5            ;LOAD R5 WITH CHAR
022104  010502                                      MOV     R5,R2               ;LOAD R2 WITH CHAR

                                                      ;CHECK FIRST FOUR CHARACTER MESSAGE
                                                      ;IN THE BIT WINDOW (0,125,252,377)

022106  012737  120001  027316                      MOV     #CRC16,XPOLY        ;LOAD POLYNOMIAL
022114  010537  022126                              MOV     R5,67$              ;LOAD SOFT CHAR FOR BCC
022120  004537  027212                              JSR     R5,SIMBCC           ;CALCULATE SOFT BCC
022124  000010                                      10                          ;SHIFT COUNT
022126  000000                              67$:    0                           ;CHARACTER
022130  000000                              10$:    0                           ;OLD BCC
022132  013737  027320  022130                      MOV     CALBCC,10$          ;LOAD SOFT BCC FOR NEXT SHIFT
022140  104415  000001                      64$:    DATACLK,        1           ;SHIFT DATA IN TO BIT WINDOW
022144  106002                                      RORB    R2                  ;SHIFT SOFT DATA
022146  103005                                      BCC     65$                 ;BR IF A SPACE
022150  004737  026316                              JSR     PC,GETSI            ;LOOK AT BIT WINDOW
022154  103406                                      BCS     66$                 ;BR IF OK (MARK)
022156  104006                                      HLT     6                   ;ERROR, BIT WINDOW WAS A SPACE
022160  000404                                      BR      66$                 ;CONTINUE
022162  004737  026316                      65$:    JSR     PC,GETSI            ;LOOK AT BIT WINDOW
022166  103001                                      BCC     66$                 ;BR IF OK (SPACE)
022170  104006                                      HLT     6                   ;ERROR, BIT WINDOW WAS A MARK
022172                                      66$:
022172  005203                                      INC     R3                  ;BUMP BIT COUNTER
022174  022703  000010                              CMP     #10,R3              ;DONE FULL 8 BITS YET
```

```
022200  001357                            BNE     64$             ;BR IF NO
022202  005003                            CLR     R3              ;CLEAR BIT COUNTER
022204  005300                            DEC     R0              ;DEC CHARACTER COUNT
022206  001335                            BNE     12$             ;BR IF NOT DONE YET

                                          ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

022210  013700  027320                    MOV     CALBCC,R0       ;PUT BCC IN R0
022214  104415  000001            68$:    DATACLK,1               ;SHIFT HARDWARE BCC
022220  006000                            ROR     R0              ;SHIFT SOFT BCC
022222  103005                            BCC     69$             ;BR IF CARRY CLEAR
022224  004737  026316                    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
022230  103406                            BCS     70$             ;BR IF OK (MARK)
022232  104014                            HLT     14              ;ERROR, CRC WRONG (SPACE)
022234  000404                            BR      70$             ;CONTINUE
022236  004737  026316            69$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
022242  103001                            BCC     70$             ;BR IF OK (SPACE)
022244  104014                            HLT     14              ;ERROR, CRC WRONG (MARK)
022246                            70$:
022246  005203                            INC     R3              ;BUMP BIT COUNTER
022250  022703  000020                    CMP     #20,R3          ;FINISHED BCC YET?
022254  001357                            BNE     68$             ;BR IF NO
022256  005003                            CLR     R3              ;CLEAR BIT COUNTER

                                          ;CHECK TO SEE IF TRANSMITTER IS MARKING

022260  104415  000001            2$:     DATACLK,        1       ;CLOCK TRANSMITTER
022264  004737  026316                    JSR     PC,GETSI        ;LOOK AT WINDOW
022270  103401                            BCS     3$              ;IT SHOULD BE MARKING
022272  104024                            HLT     24              ;ERROR, BIT WAS A SPACE
022274  005203                    3$:     INC     R3              ;BUMP BIT COUNTER
022276  022703  000007                    CMP     #7,R3           ;DONE YET
022302  001366                            BNE     2$              ;BR IF NO
022304  104415  000010                    DATACLK,        10      ;GIVE ENOUGH TICKS TO CLEAR OUT ACTIVE
022310  005003                            CLR     R3              ;CLEAR BIT COUNTER
022312  104415  000001            4$:     DATACLK,        1       ;SHIFT OUT NEXT BIT
022316  004737  026316                    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
022322  103401                            BCS     .+4             ;BR IF IT IS A MARK
022324  104024                            HLT     24              ;ERROR, TRANSMITTER IS NOT MARKING
022326  005203                            INC     R3              ;INC BIT COUNT
022330  022703  000020                    CMP     #20,R3          ;DONE YET?
022334  001366                            BNE     4$              ;BR IF NO
022336  104400                    5$:     SCOPE                   ;SCOPE THIS TEST


                                          ;**************************** TEST 51 ****************************
                                          ;*RECEIVER DDCMP CRC TEST
                                          ;*THIS TEST CLOCKS A FOUR CHARACTER MESSAGE WITH BCC
                                          ;*AND VERIFYS CORRECT DATA RECEPTION AND BCC MATCH
                                          ;*THE FOUR CHARACTER MESSAGE IS 0,125,252,377
                                          ;;*************************************************************

                                          ;  TEST 51
                                          ;--------------
022340  012737  000051  001226    TST51:  MOV     #51,TSTNO
022346  012737  022542  001216            MOV     #TST52,NEXT
```

```
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
022354  104412                          MSTCLR                  ;MASTER CLEAR DMC11
022356  012711  004000                  MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
022362  012702  030102                  MOV     #MESDAT,R2      ;LOAD POINTER TO DATA
022366  012700  000004                  MOV     #4,R0           ;LOAD CHARACTER COUNT
022372  004737  027464                  JSR     PC,SYNLD        ;LOAD 2 SYNCS IN OUT SILO
022376  004737  026502                  JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
022402  004537  027620                  JSR     R5,MESLD        ;LOAD SILO WITH 4 CHAR MESS
022406  030102                          MESDAT                  ;ADDRESS OF MESSAGE
022410  000004                          4                       ;NUMBER OF CHARACTERS
022412  004737  027574                  JSR     PC,EOM          ;LOAD GARBAGE CHARACTER, WITH EOM SET
022416  004737  026350                  JSR     PC,OCOR         ;WAIT FOR OCOR
022422  104415  000114                  DATACLK,114             ;CLOCK DATA
022426  004737  027156          3$:     JSR     PC,INRDY        ;WAIT FOR INRDY
022432  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022434  021204                          021204                  ;GET IN DATA
022436  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
022442  112205                          MOVB    (R2)+,R5        ;PUT "EXPECTED" IN R5
022444  120504                          CMPB    R5,R4           ;COMPARE RECEIVED DATA
022446  001401                          BEQ     1$              ;BR IF OK
022450  104010                          HLT     10              ;DATA ERROR
022452  005300                  1$:     DEC     R0              ;DEC CHARACTER COUNT
022454  001364                          BNE     3$              ;BR IF NOT DONE YET

                                ;CHECK TO SEE THAT IN BCC MATCH IS SET

022456  004737  027156                  JSR     PC,INRDY        ;WAIT FOR INRDY
022462  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022464  021204                          021204                  ;GET FIRST HALF OF CRC
022466  116137  000004  001252          MOVB    4(R1),TEMP3     ;PUT IN TEMP3
022474  042737  177400  001252          BIC     #177400,TEMP3   ;CLEAR HI BYTE
022502  004737  027156                  JSR     PC,INRDY        ;WAIT FOR INRDY
022506  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022510  021244                          021244
022512  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
022516  042704  000376                  BIC     #376,R4         ;CLEAR UNWANTED BITS
022522  012705  000001                  MOV     #1,R5           ;PUT "EXPECTED" IN R5
022526  120504                          CMPB    R5,R4           ;IS IN BCC MATCH SET?
022530  001401                          BEQ     25$
022532  104015                          HLT     15              ;IN BCC MATCH ERROR
022534                          25$:
022534  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022536  021204                          021204                  ;GET LAST HALF
022540  104400                  2$:     SCOPE                   ;SCOPE THIS TEST



                                ;*************************** TEST 52 **************************
                                ;*DDCMP EOM FUNCTION TEST
                                ;*THIS TEST LOADS OUT SILO WITH: 2 SYNCS,4 CHAR MESSAGE,EOM
                                ;*4 CHARACTER MESS,EOM. THE DATA STREAM IS CHECKED TO BE
                                ;*4 CHAR,BCC,4 CHAR,BCC,MARKS. THIS TEST VERIFYS THAT
                                ;*THE CHARCTERS LOADED WITH EOM SET ARE LOST
                                ;*ALL DATA AND BCC'S ARE CHECKED IN THE BIT WINDOW
                                ;*THE FOUR CHARACTER MESSAGE IS 0,125,252,377
                                ;*RECEIVED DATA IS VERIFIED, AND IN BCC MATCH IS CHECKED
                                ;;************************************************************
```

```

                                                      ;  TEST 52
                                                      ;---------------
 022542  012737  000052  001226        TST52:  MOV     #52,TSTNO
 022550  012737  023642  001216                MOV     #TST53,NEXT
                                                                       ;R1 CONTAINS BASE DMC11 ADDRESS
 022556  104412                                MSTCLR                  ;MASTER CLEAR DMC11

                                                      ;LOAD OUT DATA SILO

 022560  012711  004000                        MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
 022564  012704  030102                        MOV     #MESDAT,R4      ;LOAD POINTER TO DATA
 022570  005037  022700                        CLR     10$             ;CLEAR SOFT BCC
 022574  012700  000004                        MOV     #4,R0           ;LOAD CHARACTER COUNT
 022600  004737  027464                        JSR     PC,SYNLD        ;LOAD 2 SYNCS IN OUT SILO
 022604  004737  026502                        JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
 022610  004537  027620                        JSR     R5,MESLD        ;LOAD SILO WITH 4 CHAR MESS
 022614  030102                                MESDAT                  ;ADDRESS OF MESSAGE
 022616  000004                                4                       ;NUMBER OF CHARACTERS
 022620  004737  027574                        JSR     PC,EOM          ;LOAD GARBAGE CHARACTER, WITH EOM SET
 022624  004537  027620                        JSR     R5,MESLD        ;LOAD FOUR MORE CHARACTERS
 022630  030102                                MESDAT                  ;ADDRESS OF MESSAGE
 022632  000004                                4                       ;NUMBER OF CHACTERS
 022634  004737  027574                        JSR     PC,EOM          ;SET EOM
 022640  004737  026350                        JSR     PC,OCOR         ;WAIT FOR OCOR
 022644  005003                                CLR     R3              ;CLEAR BIT COUNTER
 022646  104415  000022                        DATACLK,22              ;CLOCK DATA
 022652  112405                        12$:    MOVB    (R4)+,R5        ;LOAD R5 WITH CHAR
 022654  010502                                MOV     R5,R2           ;LOAD R2 WITH CHAR

                                                      ;CHECK FIRST FOUR CHARACTER MESSAGE
                                                      ;IN THE BIT WINDOW (0,125,252,377)

 022656  012737  120001  027316                MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL
 022664  010537  022676                        MOV     R5,67$          ;LOAD SOFT CHAR FOR BCC
 022670  004537  027212                        JSR     R5,SIMBCC       ;CALCULATE SOFT BCC
 022674  000010                                10                      ;SHIFT COUNT
 022676  000000                        67$:    0                       ;CHARACTER
 022700  000000                        10$:    0                       ;OLD BCC
 022702  013737  027320  022700                MOV     CALBCC,10$      ;LOAD SOFT BCC FOR NEXT SHIFT
 022710  104415  000001                64$:    DATACLK,        1       ;SHIFT DATA IN TO BIT WINDOW
 022714  106002                                RORB    R2              ;SHIFT SOFT DATA
 022716  103005                                BCC     65$             ;BR IF A SPACE
 022720  004737  026316                        JSR     PC,GETSI        ;LOOK AT BIT WINDOW
 022724  103406                                BCS     66$             ;BR IF OK (MARK)
 022726  104006                                HLT     6               ;ERROR, BIT WINDOW WAS A SPACE
 022730  000404                                BR      66$             ;CONTINUE
 022732  004737  026316                65$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
 022736  103001                                BCC     66$             ;BR IF OK (SPACE)
 022740  104006                                HLT     6               ;ERROR, BIT WINDOW WAS A MARK
 022742                                66$:
 022742  005203                                INC     R3              ;BUMP BIT COUNTER
 022744  022703  000010                        CMP     #10,R3          ;DONE FULL 8 BITS YET
 022750  001357                                BNE     64$             ;BR IF NO
 022752  005003                                CLR     R3              ;CLEAR BIT COUNTER
 022754  005300                                DEC     R0              ;DEC CHARACTER COUNT
```

```
022756  001335                          BNE     12$             ;BR IF NOT DONE YET

                                        ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

022760  013700  027320                  MOV     CALBCC,R0       ;PUT BCC IN R0
022764  104415  000001          68$:    DATACLK,1               ;SHIFT HARDWARE BCC
022770  006000                          ROR     R0              ;SHIFT SOFT BCC
022772  103005                          BCC     69$             ;BR IF CARRY CLEAR
022774  004737  026316                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
023000  103406                          BCS     70$             ;BR IF OK (MARK)
023002  104014                          HLT     14              ;ERROR, CRC WRONG (SPACE)
023004  000404                          BR      70$             ;CONTINUE
023006  004737  026316          69$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
023012  103001                          BCC     70$             ;BR IF OK (SPACE)
023014  104014                          HLT     14              ;ERROR, CRC WRONG (MARK)
023016                          70$:
023016  005203                          INC     R3              ;BUMP BIT COUNTER
023020  022703  000020                  CMP     #20,R3          ;FINISHED BCC YET?
023024  001357                          BNE     68$             ;BR IF NO
023026  005003                          CLR     R3              ;CLEAR BIT COUNTER
023030  012700  000004                  MOV     #4,R0           ;RESET CHARACTER COUNTER
023034  012704  030102                  MOV     #MESDAT,R4      ;LOAD MESSAGE POINTER
023040  005037  023072                  CLR     11$             ;CLR SOFT BCC
023044  112405                  13$:    MOVB    (R4)+,R5        ;LOAD CHAR IN R5
023046  010502                          MOV     R5,R2           ;LOAD CHAR IN R2

                                        ;CHECK SECOND MESSAGE IN THE BIT WINDOW (0,125,252,377)

023050  012737  120001  027316          MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL
023056  010537  023070                  MOV     R5,76$          ;LOAD SOFT CHAR FOR BCC
023062  004537  027212                  JSR     R5,SIMBCC       ;CALCULATE SOFT BCC
023066  000010                          10                      ;SHIFT COUNT
023070  000000                  76$:    0                       ;CHARACTER
023072  000000                  11$:    0                       ;OLD BCC
023074  013737  027320  023072          MOV     CALBCC,11$      ;LOAD SOFT BCC FOR NEXT SHIFT
023102  104415  000001          73$:    DATACLK,        1       ;SHIFT DATA IN TO BIT WINDOW
023106  106002                          RORB    R2              ;SHIFT SOFT DATA
023110  103005                          BCC     74$             ;BR IF A SPACE
023112  004737  026316                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
023116  103406                          BCS     75$             ;BR IF OK (MARK)
023120  104006                          HLT     6               ;ERROR, BIT WINDOW WAS A SPACE
023122  000404                          BR      75$             ;CONTINUE
023124  004737  026316          74$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
023130  103001                          BCC     75$             ;BR IF OK (SPACE)
023132  104006                          HLT     6               ;ERROR, BIT WINDOW WAS A MARK
023134                          75$:
023134  005203                          INC     R3              ;BUMP BIT COUNTER
023136  022703  000010                  CMP     #10,R3          ;DONE FULL 8 BITS YET
023142  001357                          BNE     73$             ;BR IF NO
023144  005003                          CLR     R3              ;CLEAR BIT COUNTER
023146  005300                          DEC     R0              ;DEC CHARACTER COUNT
023150  001335                          BNE     13$             ;BR IF NOT DONE YET

                                        ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

023152  013700  027320                  MOV     CALBCC,R0       ;PUT BCC IN R0
```

```
023156  104415  000001     77$:   DATACLK,1                  ;SHIFT HARDWARE BCC
023162  006000                    ROR     R0                 ;SHIFT SOFT BCC
023164  103005                    BCC     78$                ;BR IF CARRY CLEAR
023166  004737  026316            JSR     PC,GETSI           ;LOOK AT BIT WINDOW
023172  103406                    BCS     79$                ;BR IF OK (MARK)
023174  104014                    HLT     14                 ;ERROR, CRC WRONG (SPACE)
023176  000404                    BR      79$                ;CONTINUE
023200  004737  026316     78$:   JSR     PC,GETSI           ;LOOK AT BIT WINDOW
023204  103001                    BCC     79$                ;BR IF OK (SPACE)
023206  104014                    HLT     14                 ;ERROR, CRC WRONG (MARK)
023210                     79$:
023210  005203                    INC     R3                 ;BUMP BIT COUNTER
023212  022703  000020            CMP     #20,R3             ;FINISHED BCC YET?
023216  001357                    BNE     77$                ;BR IF NO
023220  005003                    CLR     R3                 ;CLEAR BIT COUNTER

                                  ;CHECK TO SEE IF TRANSMITTER IS MARKING

023222  104415  000001     2$:    DATACLK,        1          ;CLOCK TRANSMITTER
023226  004737  026316            JSR     PC,GETSI           ;LOOK AT WINDOW
023232  103401                    BCS     3$                 ;IT SHOULD BE MARKING
023234  104024                    HLT     24                 ;ERROR, BIT WAS A SPACE
023236  005203             3$:    INC     R3                 ;BUMP BIT COUNTER
023240  022703  000007            CMP     #7,R3              ;DONE YET
023244  001366                    BNE     2$                 ;BR IF NO
023246  104415  000010            DATACLK,            10     ;GIVE ENOUGH TICKS TO CLEAR OUT ACTIVE
023252  005003                    CLR     R3                 ;CLEAR BIT COUNTER
023254  104415  000001     4$:    DATACLK,        1          ;SHIFT OUT NEXT BIT
023260  004737  026316            JSR     PC,GETSI           ;LOOK AT BIT WINDOW
023264  103401                    BCS     .+4                ;BR IF IT IS A MARK
023266  104024                    HLT     24                 ;ERROR, TRANSMITTER IS NOT MARKING
023270  005203                    INC     R3                 ;INC BIT COUNT
023272  022703  000020            CMP     #20,R3             ;DONE YET?
023276  001366                    BNE     4$                 ;BR IF NO

                                  ;CHECK TO SEE THAT FIRST FOUR CHARACTER MESSAGE
                                  ;WAS RECEIVED CORRECTLY (0,125,252,377)

023300  104415  000001            DATACLK,        1          ;GET LAST BIT IN RECEIVER
023304  012703  000004            MOV     #4,R3              ;R3=CHARACTER COUNT
023310  012702  030102            MOV     #MESDAT,R2         ;LOAD MESSAGE POINTER IN R2
023314  004737  027156     40$:   JSR     PC,INRDY           ;WAIT FOR INRDY
023320  104414                    ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023322  021204                    021204
023324  016104  000004            MOV     4(R1),R4           ;PUT "FOUND" IN R4
023330  112205                    MOVB    (R2)+,R5           ;PUT "EXPECTED" IN R5
023332  120504                    CMPB    R5,R4              ;IS RECEIVED DATA CORRECT?
023334  001401                    BEQ     41$                ;BR IF YES
023336  104010                    HLT     10                 ;RECEIVE DATA ERROR
023340  005303             41$:   DEC     R3                 ;DEC CHARACTER COUNT
023342  001364                    BNE     40$                ;BR IF NOT DONE YET

                                  ;CHECK TO SEE THAT IN BCC MATCH IS SET
                                  ;AND THAT THE BCC WAS RECEIVED CORRECTLY

023344  004737  027156            JSR     PC,INRDY           ;WAIT FOR INRDY
```

```
023350  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023352  021204                                  021204                          ;GET FIRST HALF OF CRC
023354  116137  000004  001252                  MOVB    4(R1),TEMP3             ;PUT IN TEMP3
023362  042737  177400  001252                  BIC     #177400,TEMP3           ;CLEAR HI BYTE
023370  004737  027156                          JSR     PC,INRDY                ;WAIT FOR INRDY
023374  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023376  021244                                  021244
023400  016104  000004                          MOV     4(R1),R4                ;PUT "FOUND" IN R4
023404  042704  000376                          BIC     #376,R4                 ;CLEAR UNWANTED BITS
023410  012705  000001                          MOV     #1,R5                   ;PUT "EXPECTED" IN R5
023414  120504                                  CMPB    R5,R4                   ;IS IN BCC MATCH SET?
023416  001401                                  BEQ     50$
023420  104015                                  HLT     15                      ;IN BCC MATCH ERROR
023422                                  50$:
023422  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023424  021204                                  021204                          ;GET LAST HALF
023426  116137  000004  001251                  MOVB    4(R1),TEMP2+1           ;PUT IN TEMP2
023434  042737  000377  001250                  BIC     #377,TEMP2              ;CLEAR LO BYTE
023442  053737  001250  001252                  BIS     TEMP2,TEMP3             ;16 BIT BCC NOW IN TEMP3
023450  023737  027320  001252                  CMP     CALBCC,TEMP3            ;IS IT CORRECT?
023456  001401                                  BEQ     42$                     ;BR IF OK
023460  104027                                  HLT     27

                                                ;CHECK TO SEE THAT SECOND FOUR CHARACTER MESSAGE
                                                ;WAS RECEIVED CORRECTLY (0,125,252,377)

023462  012703  000004                  42$:    MOV     #4,R3                   ;R3=CHARACTER COUNT
023466  012702  030102                          MOV     #MESDAT,R2              ;LOAD MESSAGE POINTER IN R2
023472  004737  027156                  43$:    JSR     PC,INRDY                ;WAIT FOR INRDY
023476  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023500  021204                                  021204
023502  016104  000004                          MOV     4(R1),R4                ;PUT "FOUND" IN R4
023506  112205                                  MOVB    (R2)+,R5                ;PUT "EXPECTED" IN R5
023510  120504                                  CMPB    R5,R4                   ;IS RECEIVED DATA CORRECT?
023512  001401                                  BEQ     44$                     ;BR IF YES
023514  104010                                  HLT     10                      ;RECEIVE DATA ERROR
023516  005303                          44$:    DEC     R3                      ;DEC CHARACTER COUNT
023520  001364                                  BNE     43$                     ;BR IF NOT DONE YET

                                                ;CHECK TO SEE THAT IN BCC MATCH IS SET
                                                ;AND THAT THE BCC WAS RECEIVED CORRECTLY

023522  004737  027156                          JSR     PC,INRDY                ;WAIT FOR INRDY
023526  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023530  021204                                  021204                          ;GET FIRST HALF OF CRC
023532  116137  000004  001252                  MOVB    4(R1),TEMP3             ;PUT IN TEMP3
023540  042737  177400  001252                  BIC     #177400,TEMP3           ;CLEAR HI BYTE
023546  004737  027156                          JSR     PC,INRDY                ;WAIT FOR INRDY
023552  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023554  021244                                  021244
023556  016104  000004                          MOV     4(R1),R4                ;PUT "FOUND" IN R4
023562  042704  000376                          BIC     #376,R4                 ;CLEAR UNWANTED BITS
023566  012705  000001                          MOV     #1,R5                   ;PUT "EXPECTED" IN R5
023572  120504                                  CMPB    R5,R4                   ;IS IN BCC MATCH SET?
023574  001401                                  BEQ     51$
023576  104015                                  HLT     15                      ;IN BCC MATCH ERROR
```

```
023600                                 51$:
023600  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023602  021204                                  021204                  ;GET LAST HALF
023604  116137  000004  001251                  MOVB    4(R1),TEMP2+1   ;PUT IN TEMP2
023612  042737  000377  001250                  BIC     #377,TEMP2      ;CLEAR LO BYTE
023620  053737  001250  001252                  BIS     TEMP2,TEMP3     ;16 BIT BCC NOW IN TEMP3
023626  023737  027320  001252                  CMP     CALBCC,TEMP3    ;IS IT CORRECT?
023634  001401                                  BEQ     5$              ;BR IF OK
023636  104027                                  HLT     27
023640  104400                          5$:     SCOPE                   ;SCOPE THIS TEST


                                                ;*************************** TEST 53 ***************************
                                                ;*DDCMP EOM FUNCTION TEST
                                                ;*THIS TEST LOADS OUT SILO WITH: 2 SYNCS,4 CHAR MESSAGE,EOM
                                                ;*SOM,4 CHAR MESS,EOM. THE DATA STREAM IS CHECKED TO BE
                                                ;*4 CHAR,BCC,4 CHAR,BCC,MARKS. THIS TEST VERIFYS THAT
                                                ;*THE CHARCTERS LOADED WITH EOM SET ARE LOST
                                                ;*ALSO THAT THE CHAR LOADED WITH SOM IS NOT IN THE BCC
                                                ;*ALL DATA AND BCC'S ARE CHECKED IN THE BIT WINDOW
                                                ;*THE FOUR CHARACTER MESSAGE IS 0,125,252,377
                                                ;*RECEIVED DATA IS VERIFIED, AND IN BCC MATCH IS CHECKED
                                                ;:**************************************************************

                                                ;  TEST 53
                                                ;--------------
023642  012737  000053  001226          TST53:  MOV     #53,TSTNO
023650  012737  025042  001216                  MOV     #TST54,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
023656  104412                                  MSTCLR                  ;MASTER CLEAR DMC11

                                                ;LOAD OUT DATA SILO

023660  012711  004000                          MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
023664  012704  030102                          MOV     #MESDAT,R4      ;LOAD POINTER TO DATA
023670  005037  024004                          CLR     10$             ;CLEAR SOFT BCC
023674  012700  000004                          MOV     #4,R0           ;LOAD CHARACTER COUNT
023700  004737  027464                          JSR     PC,SYNLD        ;LOAD 2 SYNCS IN OUT SILO
023704  004737  026502                          JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
023710  004537  027620                          JSR     R5,MESLD        ;LOAD SILO WITH 4 CHAR MESS
023714  030102                                  MESDAT                  ;ADDRESS OF MESSAGE
023716  000004                                  4                       ;NUMBER OF CHARACTERS
023720  004737  027574                          JSR     PC,EOM          ;LOAD GARBAGE CHARACTER, WITH EOM SET
023724  004737  027544                          JSR     PC,SOM          ;LOAD GARBAGE CHAR WITH SOM SET
023730  004537  027620                          JSR     R5,MESLD        ;LOAD FOUR MORE CHARACTERS
023734  030102                                  MESDAT                  ;ADDRESS OF MESSAGE
023736  000004                                  4                       ;NUMBER OF CHACTERS
023740  004737  027574                          JSR     PC,EOM          ;SET EOM
023744  004737  026350                          JSR     PC,OCOR         ;WAIT FOR OCOR
023750  005003                                  CLR     R3              ;CLEAR BIT COUNTER
023752  104415  000022                          DATACLK,22              ;CLOCK DATA
023756  112405                          12$:    MOVB    (R4)+,R5        ;LOAD R5 WITH CHAR
023760  010502                                  MOV     R5,R2           ;LOAD R2 WITH CHAR


                                                ;CHECK FIRST FOUR CHARACTER MESSAGE
                                                ;IN THE BIT WINDOW (0,125,252,377)
```

```

023762  012737  120001  027316                  MOV     #CRC16,XPOLY            ;LOAD POLYNOMIAL
023770  010537  024002                          MOV     R5,67$                  ;LOAD SOFT CHAR FOR BCC
023774  004537  027212                          JSR     R5,SIMBCC               ;CALCULATE SOFT BCC
024000  000010                                  10                              ;SHIFT COUNT
024002  000000                          67$:    0                               ;CHARACTER
024004  000000                          10$:    0                               ;OLD BCC
024006  013737  027320  024004                  MOV     CALBCC,10$              ;LOAD SOFT BCC FOR NEXT SHIFT
024014  104415  000001                  64$:    DATACLK,        1               ;SHIFT DATA IN TO BIT WINDOW
024020  106002                                  RORB    R2                      ;SHIFT SOFT DATA
024022  103005                                  BCC     65$                     ;BR IF A SPACE
024024  004737  026316                          JSR     PC,GETSI                ;LOOK AT BIT WINDOW
024030  103406                                  BCS     66$                     ;BR IF OK (MARK)
024032  104006                                  HLT     6                       ;ERROR, BIT WINDOW WAS A SPACE
024034  000404                                  BR      66$                     ;CONTINUE
024036  004737  026316                  65$:    JSR     PC,GETSI                ;LOOK AT BIT WINDOW
024042  103001                                  BCC     66$                     ;BR IF OK (SPACE)
024044  104006                                  HLT     6                       ;ERROR, BIT WINDOW WAS A MARK
024046                                  66$:
024046  005203                                  INC     R3                      ;BUMP BIT COUNTER
024050  022703  000010                          CMP     #10,R3                  ;DONE FULL 8 BITS YET
024054  001357                                  BNE     64$                     ;BR IF NO
024056  005003                                  CLR     R3                      ;CLEAR BIT COUNTER
024060  005300                                  DEC     R0                      ;DEC CHARACTER COUNT
024062  001335                                  BNE     12$                     ;BR IF NOT DONE YET

                                                ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

024064  013700  027320                          MOV     CALBCC,R0               ;PUT BCC IN R0
024070  104415  000001                  68$:    DATACLK,1                       ;SHIFT HARDWARE BCC
024074  006000                                  ROR     R0                      ;SHIFT SOFT BCC
024076  103005                                  BCC     69$                     ;BR IF CARRY CLEAR
024100  004737  026316                          JSR     PC,GETSI                ;LOOK AT BIT WINDOW
024104  103406                                  BCS     70$                     ;BR IF OK (MARK)
024106  104014                                  HLT     14                      ;ERROR, CRC WRONG (SPACE)
024110  000404                                  BR      70$                     ;CONTINUE
024112  004737  026316                  69$:    JSR     PC,GETSI                ;LOOK AT BIT WINDOW
024116  103001                                  BCC     70$                     ;BR IF OK (SPACE)
024120  104014                                  HLT     14                      ;ERROR, CRC WRONG (MARK)
024122                                  70$:
024122  005203                                  INC     R3                      ;BUMP BIT COUNTER
024124  022703  000020                          CMP     #20,R3                  ;FINISHED BCC YET?
024130  001357                                  BNE     68$                     ;BR IF NO
024132  005003                                  CLR     R3                      ;CLEAR BIT COUNTER

                                                ;CHECK CHARACTER LOADED WITH SOM (000), IN THE BIT WINDOW

024134  005005                                  CLR     R5                      ;CHARACTER LOADED WITH SOM
024136  010502                                  MOV     R5,R2                   ;LOAD R2 WITH CHAR
024140  104415  000001                  32$:    DATACLK,        1               ;CLOCK TRANSMITTER
024144  106002                                  RORB    R2                      ;SHIFT SOFT DATA
024146  103005                                  BCC     30$                     ;BR IF SPACE
024150  004737  026316                          JSR     PC,GETSI                ;LOOK AT BIT WINDOW
024154  103406                                  BCS     31$                     ;BR IF OK (MARK)
024156  104006                                  HLT     6                       ;ERROR,BIT WINDOW WAS A SPACE
024160  000404                                  BR      31$                     ;CONTINUE
```

```
024162  004737  026316          30$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024166  103001                          BCC     31$             ;BR IF OK (SPACE)
024170  104006                          HLT     6               ;ERROR,BIT WINDOW WAS A MARK
024172  005203                  31$:    INC     R3              ;BUMP BIT COUNTER
024174  022703  000010                  CMP     #10,R3          ;DONE CHARACTER YET?
024200  001357                          BNE     32$             ;BR IF NO
024202  005003                          CLR     R3              ;RESET BIT COUNTER
024204  012700  000004                  MOV     #4,R0           ;RESET CHARACTER COUNTER
024210  012704  030102                  MOV     #MESDAT,R4      ;LOAD MESSAGE POINTER
024214  005037  024246                  CLR     11$             ;CLR SOFT BCC
024220  112405                  13$:    MOVB    (R4)+,R5        ;LOAD CHAR IN R5
024222  010502                          MOV     R5,R2           ;LOAD CHAR IN R2

                                ;CHECK SECOND MESSAGE IN THE BIT WINDOW (0,125,252,377)

024224  012737  120001  027316          MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL
024232  010537  024244                  MOV     R5,76$          ;LOAD SOFT CHAR FOR BCC
024236  004537  027212                  JSR     R5,SIMBCC       ;CALCULATE SOFT BCC
024242  000010                          10                      ;SHIFT COUNT
024244  000000                  76$:    0                       ;CHARACTER
024246  000000                  11$:    0                       ;OLD BCC
024250  013737  027320  024246          MOV     CALBCC,11$      ;LOAD SOFT BCC FOR NEXT SHIFT
024256  104415  000001          73$:    DATACLK,        1       ;SHIFT DATA IN TO BIT WINDOW
024262  106002                          RORB    R2              ;SHIFT SOFT DATA
024264  103005                          BCC     74$             ;BR IF A SPACE
024266  004737  026316                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024272  103406                          BCS     75$             ;BR IF OK (MARK)
024274  104006                          HLT     6               ;ERROR, BIT WINDOW WAS A SPACE
024276  000404                          BR      75$             ;CONTINUE
024300  004737  026316          74$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024304  103001                          BCC     75$             ;BR IF OK (SPACE)
024306  104006                          HLT     6               ;ERROR, BIT WINDOW WAS A MARK
024310                          75$:
024310  005203                          INC     R3              ;BUMP BIT COUNTER
024312  022703  000010                  CMP     #10,R3          ;DONE FULL 8 BITS YET
024316  001357                          BNE     73$             ;BR IF NO
024320  005003                          CLR     R3              ;CLEAR BIT COUNTER
024322  005300                          DEC     R0              ;DEC CHARACTER COUNT
024324  001335                          BNE     13$             ;BR IF NOT DONE YET

                                ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

024326  013700  027320                  MOV     CALBCC,R0       ;PUT BCC IN R0
024332  104415  000001          77$:    DATACLK,1               ;SHIFT HARDWARE BCC
024336  006000                          ROR     R0              ;SHIFT SOFT BCC
024340  103005                          BCC     78$             ;BR IF CARRY CLEAR
024342  004737  026316                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024346  103406                          BCS     79$             ;BR IF OK (MARK)
024350  104014                          HLT     14              ;ERROR, CRC WRONG (SPACE)
024352  000404                          BR      79$             ;CONTINUE
024354  004737  026316          78$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024360  103001                          BCC     79$             ;BR IF OK (SPACE)
024362  104014                          HLT     14              ;ERROR, CRC WRONG (MARK)
024364                          79$:
024364  005203                          INC     R3              ;BUMP BIT COUNTER
024366  022703  000020                  CMP     #20,R3          ;FINISHED BCC YET?
```

```
024372  001357                                  BNE     77$             ;BR IF NO
024374  005003                                  CLR     R3              ;CLEAR BIT COUNTER

                                        ;CHECK TO SEE IF TRANSMITTER IS MARKING

024376  104415  000001          2$:     DATACLK,        1       ;CLOCK TRANSMITTER
024402  004737  026316                  JSR     PC,GETSI        ;LOOK AT WINDOW
024406  103401                          BCS     3$              ;IT SHOULD BE MARKING
024410  104024                          HLT     24              ;ERROR, BIT WAS A SPACE
024412  005203                  3$:     INC     R3              ;BUMP BIT COUNTER
024414  022703  000007                  CMP     #7,R3           ;DONE YET
024420  001366                          BNE     2$              ;BR IF NO
024422  104415  000010                  DATACLK,        10      ;GIVE ENOUGH TICKS TO CLEAR OUT ACTIVE
024426  005003                          CLR     R3              ;CLEAR BIT COUNTER
024430  104415  000001          4$:     DATACLK,        1       ;SHIFT OUT NEXT BIT
024434  004737  026316                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024440  103401                          BCS     .+4             ;BR IF IT IS A MARK
024442  104024                          HLT     24              ;ERROR, TRANSMITTER IS NOT MARKING
024444  005203                          INC     R3              ;INC BIT COUNT
024446  022703  000020                  CMP     #20,R3          ;DONE YET?
024452  001366                          BNE     4$              ;BR IF NO

                                        ;CHECK TO SEE THAT FIRST FOUR CHARACTER MESSAGE
                                        ;WAS RECEIVED CORRECTLY (0,125,252,377)

024454  104415  000001                  DATACLK,        1       ;GET LAST BIT IN RECEIVER
024460  012703  000004                  MOV     #4,R3           ;R3=CHARACTER COUNT
024464  012702  030102                  MOV     #MESDAT,R2      ;LOAD MESSAGE POINTER IN R2
024470  004737  027156          40$:    JSR     PC,INRDY        ;WAIT FOR INRDY
024474  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024476  021204                          021204
024500  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
024504  112205                          MOVB    (R2)+,R5        ;PUT "EXPECTED" IN R5
024506  120504                          CMPB    R5,R4           ;IS RECEIVED DATA CORRECT?
024510  001401                          BEQ     41$             ;BR IF YES
024512  104010                          HLT     10              ;RECEIVE DATA ERROR
024514  005303                  41$:    DEC     R3              ;DEC CHARACTER COUNT
024516  001364                          BNE     40$             ;BR IF NOT DONE YET

                                        ;CHECK TO SEE THAT IN BCC MATCH IS SET
                                        ;AND THAT THE BCC WAS RECEIVED CORRECTLY

024520  004737  027156                  JSR     PC,INRDY        ;WAIT FOR INRDY
024524  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024526  021204                          021204                  ;GET FIRST HALF OF CRC
024530  116137  000004  001252          MOVB    4(R1),TEMP3     ;PUT IN TEMP3
024536  042737  177400  001252          BIC     #177400,TEMP3   ;CLEAR HI BYTE
024544  004737  027156                  JSR     PC,INRDY        ;WAIT FOR INRDY
024550  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024552  021244                          021244
024554  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
024560  042704  000376                  BIC     #376,R4         ;CLEAR UNWANTED BITS
024564  012705  000001                  MOV     #1,R5           ;PUT "EXPECTED" IN R5
024570  120504                          CMPB    R5,R4           ;IS IN BCC MATCH SET?
024572  001401                          BEQ     50$
024574  104015                          HLT     15              ;IN BCC MATCH ERROR
```

```
024576                                      50$:
024576  104414                                      ROMCLK                        ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024600  021204                                      021204                        ;GET LAST HALF
024602  116137  000004  001251                      MOVB    4(R1),TEMP2+1         ;PUT IN TEMP2
024610  042737  000377  001250                      BIC     #377,TEMP2            ;CLEAR LO BYTE
024616  053737  001250  001252                      BIS     TEMP2,TEMP3           ;16 BIT BCC NOW IN TEMP3
024624  023737  027320  001252                      CMP     CALBCC,TEMP3          ;IS IT CORRECT?
024632  001401                                      BEQ     45$                   ;BR IF OK
024634  104027                                      HLT     27

                                                    ;CHECK THAT CHARACTER LOADED WITH SOM WAS RECEIVED (000)

024636  004737  027156                      45$:    JSR     PC,INRDY              ;WAIT FOR INRDY
024642  104414                                      ROMCLK                        ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024644  021204                                      021204                        ;GET RECEIVE DATA
024646  016104  000004                              MOV     4(R1),R4              ;PUT "FOUND" IN R4
024652  005005                                      CLR     R5                    ;PUT "EXPECTED" IN R5
024654  120504                                      CMPB    R5,R4                 ;IS RECEVIED DATA CORRECT?
024656  001401                                      BEQ     42$                   ;BR IF YES
024660  104010                                      HLT     10                    ;RECEIVE DATA ERROR

                                                    ;CHECK TO SEE THAT SECOND FOUR CHARACTER MESSAGE
                                                    ;WAS RECEIVED CORRECTLY (0,125,252,377)

024662  012703  000004                      42$:    MOV     #4,R3                 ;R3=CHARACTER COUNT
024666  012702  030102                              MOV     #MESDAT,R2            ;LOAD MESSAGE POINTER IN R2
024672  004737  027156                      43$:    JSR     PC,INRDY              ;WAIT FOR INRDY
024676  104414                                      ROMCLK                        ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024700  021204                                      021204
024702  016104  000004                              MOV     4(R1),R4              ;PUT "FOUND" IN R4
024706  112205                                      MOVB    (R2)+,R5              ;PUT "EXPECTED" IN R5
024710  120504                                      CMPB    R5,R4                 ;IS RECEIVED DATA CORRECT?
024712  001401                                      BEQ     44$                   ;BR IF YES
024714  104010                                      HLT     10                    ;RECEIVE DATA ERROR
024716  005303                              44$:    DEC     R3                    ;DEC CHARACTER COUNT
024720  001364                                      BNE     43$                   ;BR IF NOT DONE YET

                                                    ;CHECK TO SEE THAT IN BCC MATCH IS SET
                                                    ;AND THAT THE BCC WAS RECEIVED CORRECTLY

024722  004737  027156                              JSR     PC,INRDY              ;WAIT FOR INRDY
024726  104414                                      ROMCLK                        ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024730  021204                                      021204                        ;GET FIRST HALF OF CRC
024732  116137  000004  001252                      MOVB    4(R1),TEMP3           ;PUT IN TEMP3
024740  042737  177400  001252                      BIC     #177400,TEMP3         ;CLEAR HI BYTE
024746  004737  027156                              JSR     PC,INRDY              ;WAIT FOR INRDY
024752  104414                                      ROMCLK                        ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024754  021244                                      021244
024756  016104  000004                              MOV     4(R1),R4              ;PUT "FOUND" IN R4
024762  042704  000376                              BIC     #376,R4               ;CLEAR UNWANTED BITS
024766  012705  000001                              MOV     #1,R5                 ;PUT "EXPECTED" IN R5
024772  120504                                      CMPB    R5,R4                 ;IS IN BCC MATCH SET?
024774  001401                                      BEQ     51$
024776  104015                                      HLT     15                    ;IN BCC MATCH ERROR
025000                                      51$:
025000  104414                                      ROMCLK                        ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
025002  021204                                  021204                  ;GET LAST HALF
025004  116137  000004  001251                  MOVB    4(R1),TEMP2+1   ;PUT IN TEMP2
025012  042737  000377  001250                  BIC     #377,TEMP2      ;CLEAR LO BYTE
025020  053737  001250  001252                  BIS     TEMP2,TEMP3     ;16 BIT BCC NOW IN TEMP3
025026  023737  027320  001252                  CMP     CALBCC,TEMP3    ;IS IT CORRECT?
025034  001401                                  BEQ     5$              ;BR IF OK
025036  104027                                  HLT     27
025040  104400                          5$:     SCOPE                   ;SCOPE THIS TEST


                                                ;*************************** TEST 54 ****************************
                                                ;*EMPTY SILO TEST
                                                ;*LOAD SILO WITH 2 SYNCS, 4 CHAR MESSAGE, SINGLE CLOCK
                                                ;*UNTIL THE SILO IS EMPTY, LOAD 4 MORE CHARACTERS IN THE
                                                ;*SILO. GIVE MORE TICKS, AND VERIFY THAT ONLY THE FIRST
                                                ;*4 CHARACTER MESSAGE WAS RECEIVED AND THAT RTS IS CLEAR
                                                ;:**************************************************************

                                                ;  TEST 54
                                                ;--------------
025042  012737  000054  001226          TST54:  MOV     #54,TSTNO
025050  012737  025274  001216                  MOV     #TST55,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
025056  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
025060  012711  004000                          MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
025064  012702  030102                          MOV     #MESDAT,R2      ;R2 POINTS TO MESSAGE
025070  012700  000004                          MOV     #4,R0           ;R0 = CHAR COUNT
025074  004737  027464                          JSR     PC,SYNLD        ;LOAD SILO WITH TWO SYNCS
025100  004737  026502                          JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
025104  004537  027620                          JSR     R5,MESLD        ;LOAD MESSAGE IN SILO
025110  030102                                  MESDAT                  ;START OF MESSAGE
025112  000004                                  4                       ;CHARACTER COUNT
025114  004737  026350                          JSR     PC,OCOR         ;WAIT FOR OCOR
025120  104415  000065                          DATACLK,        65      ;CLOCK DATA (EMPTY SILO)
025124  004537  027620                          JSR     R5,MESLD        ;PUT MORE CHARACTERS IN SILO
025130  030102                                  MESDAT
025132  000004                                  4
025134  004737  026350                          JSR     PC,OCOR
025140  104415  000005                          DATACLK,        5       ;CLOCK UNTIL RTS IS CLEARED
025144  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025146  021264                                  021264                  ;GET RTS
025150  032761  000040  000004                  BIT     #BIT5,4(R1)     ;IS IT CLEAR?
025156  001401                                  BEQ     5$              ;BR IF YES
025160  104034                                  HLT     34              ;ERROR, RTS NOT CLEAR
025162  104415  000041                  5$:     DATACLK,        41      ;CLOCK XMITTER SOME MORE
025166  004737  027156                  1$:     JSR     PC,INRDY        ;OK LETS CHECK WHAT WAS RECEIVED
025172  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025174  021204                                  021204                  ;GET RECEIVE DATA
025176  016104  000004                          MOV     4(R1),R4        ;PUT IT IN R4
025202  112205                                  MOVB    (R2)+,R5        ;R5 = "EXPECTED"
025204  120504                                  CMPB    R5,R4           ;IS DATA CORRECT?
025206  001401                                  BEQ     2$              ;BR IF OK
025210  104010                                  HLT     10              ;DATA ERROR
025212  005300                          2$:     DEC     R0              ;DEC CHAR COUNT
025214  001364                                  BNE     1$              ;BR IF NOT DONE YET
025216  004737  027156                  3$:     JSR     PC,INRDY        ;WAIT FOR INRDY
```

```
025222  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025224  021204                                  021204                          ;GET RECEIVE DATA
025226  016104  000004                          MOV     4(R1),R4                ;PUT IT IN "FOUND"
025232  012705  000377                          MOV     #377,R5                 ;R5 = "EXPECTED"
025236  120504                                  CMPB    R5,R4                   ;SHOULD SEE 377
025240  001401                                  BEQ     4$                      ;BR IF OK
025242  104010                                  HLT     10                      ;ERROR, TRANSMITTER DID NOT ABORT
025244  004737  027156                  4$:     JSR     PC,INRDY                ;WAIT FOR INRDY
025250  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025252  021204                                  021204                          ;GET RECEIVE DATA
025254  016104  000004                          MOV     4(R1),R4                ;PUT IT IN "FOUND"
025260  012705  000377                          MOV     #377,R5                 ;R5 = "EXPECTED"
025264  120504                                  CMPB    R5,R4                   ;SHOULD SEE 377
025266  001401                                  BEQ     10$                     ;BR IF OK
025270  104010                                  HLT     10                      ;ERROR, TRANSMITTER DID NOT ABORT
025272                                  10$:
025272  104400                                  SCOPE                           ;SCOPE THIS TEST



                                                ;*************************** TEST 55 ***************************
                                                ;*HALF DUPLEX TEST
                                                ;*SET LINE UNIT LOOP AND HALF DUPLEX, SEND SYNCS AND A
                                                ;*MESSAGE. VERIFY THAT IN-ACTIVE AND IN-READY ARE CLEAR
                                                ;:*************************************************************


                                                ;  TEST 55
                                                ;--------------
025274  012737  000055  001226          TST55:  MOV     #55,TSTNO
025302  012737  025412  001216                  MOV     #TST56,NEXT
                                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
025310  104412                                  MSTCLR                          ;MASTER CLEAR DMC11
025312  012702  000012                          MOV     #12,R2                  ;SAVE R2 FOR TYPEOUT
025316  012711  004000                          MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
025322  012761  000020  000004                  MOV     #BIT4,4(R1)             ;LOAD PORT4
025330  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025332  122113                                  122113                          ;SET H/D BIT
025334  004737  027464                          JSR     PC,SYNLD                ;LOAD 2 SYNCS
025340  004737  026502                          JSR     PC,OUTRDY               ;WAIT FOR OUTRDY
025344  004537  027620                          JSR     R5,MESLD                ;LOAD 4 CHAR MESSAGE
025350  030102                                  MESDAT                          ;ADDRESS OF MESSAGE
025352  000004                                  4                               ;CHARACTER COUNT
025354  004737  026350                          JSR     PC,OCOR                 ;WAIT FOR OCOR
025360  104415  000073                          DATACLK,        73              ;SEND MESSAGE
025364  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025366  021244                                  021244                          ;READ LU-12
025370  016104  000004                          MOV     4(R1),R4                ;PUT "FOUND" IN R4
025374  042704  000257                          BIC     #257,R4                 ;CLEAR UNWANTED BITS
025400  005005                                  CLR     R5                      ;R5 = "EXPECTED"
025402  120504                                  CMPB    R5,R4                   ;IN-ACTIVE AND IN-RDY SHOULD BE CLEAR
025404  001401                                  BEQ     1$                      ;BR IF OK
025406  104035                                  HLT     35                      ;ERROR BOTH ARE NOT CLEAR
025410  104400                          1$:     SCOPE



                                                ;*************************** TEST 56 ***************************
                                                ;*DDCMP CABLE DATA TEST
```

```
                                                      ;*THIS TEST LOADS OUT SILO WITH THE FOLLOWING:
                                                      ;*4 SYNCS,16 CHAR,EOM,16 CHAR,EOM,16 CHAR,EOM
                                                      ;*THE 16 CHARACTERS INCLUDE A FLOATING ONE AND ZERO
                                                      ;*THE DATA IS TRANSMITTED OVER THE CABLE USING THE INTERNAL CLOCK
                                                      ;*RECEIVED DATA IS VERIFIED AS IS IN BCC MATCH
                                                      ;*LOOP-BACK CONNECTOR MUST BE ON TO RUN THIS TEST
                                                      ;:************************************************************

                                                      ;  TEST 56
                                                      ;---------------
025412  012737  000056  001226          TST56:  MOV     #56,TSTNO
025420  012737  026000  001216                  MOV     #TST57,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
025426  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
025430  032737  040000  001366                  BIT     #BIT14,STAT1    ;SKIP TEST IF NO
025436  001557                                  BEQ     3$              ;LOOPBACK CONNECTOR ON
025440  012711  004000                          MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
025444  004737  027464                          JSR     PC,SYNLD        ;LOAD 2 SYNCS
025450  004737  027464                          JSR     PC,SYNLD        ;LOAD 2 MORE SYNCS
025454  012737  120001  027316                  MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL FOR SOFT CRC CALC
025462  005037  025512                          CLR     6$              ;CLEAR OLD BCC
025466  012703  000020                          MOV     #16.,R3         ;CHARACTER COUNT
025472  012702  030106                          MOV     #FLTDAT,R2      ;R2= POINTER
025476  112237  025510                  7$:     MOVB    (R2)+,5$        ;LOAD CHAR FOR SOFT BCC CALC.
025502  004537  027212                          JSR     R5,SIMBCC       ;CALC SOFT BCC
025506  000010                                  10                      ;SHIFT COUNT
025510  000000                          5$:     0                       ;CHARACTER
025512  000000                          6$:     0                       ;OLD BCC
025514  013737  027320  025512                  MOV     CALBCC,6$       ;LOAD OLD BCC
025522  005303                                  DEC     R3              ;DEC COUNT
025524  001364                                  BNE     7$              ;BR IF NOT DONE YET
025526  004537  027620                          JSR     R5,MESLD        ;LOAD SILO
025532  030106                                  FLTDAT                  ;MESSAGE ADDRESS
025534  000020                                  16.                     ;CHARACTER COUNT
025536  004737  027574                          JSR     PC,EOM          ;LOAD AN EOM
025542  004537  027620                          JSR     R5,MESLD        ;LOAD SILO
025546  030106                                  FLTDAT                  ;MESSAGE ADDRESS
025550  000020                                  16.                     ;CHARACTER COUNT
025552  004737  027574                          JSR     PC,EOM          ;LOAD AN EOM
025556  004537  027620                          JSR     R5,MESLD        ;LOAD SILO
025562  030106                                  FLTDAT                  ;MESSAGE ADDRESS
025564  000020                                  16.                     ;CHARACTER COUNT
025566  004737  027574                          JSR     PC,EOM          ;LOAD AN EOM
025572  004737  026350                          JSR     PC,OCOR         ;WAIT FOR OCOR
025576  005011                                  CLR     (R1)            ;CLEAR LINE UNIT LOOP
025600  012700  000003                          MOV     #3,R0           ;R0 = MESSAGE COUNT
025604  012703  000020                          MOV     #16.,R3         ;R3= CHARACTER COUNT
025610  012702  030106                          MOV     #FLTDAT,R2      ;LOAD MESSAGE POINTER IN R2
025614  004737  027156                  1$:     JSR     PC,INRDY        ;WAIT FOR INRDY
025620  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025622  021204                                  021204                  ;GET DATA FROM IN SILO
025624  016104  000004                          MOV     4(R1),R4        ;PUT CHARACTER IN "FOUND"
025630  112205                                  MOVB    (R2)+,R5        ;PUT "EXPECTED" IN R5
025632  120504                                  CMPB    R5,R4           ;IS RECEIVED DATA CORRECT
025634  001401                                  BEQ     2$              ;BR IF OK
025636  104025                                  HLT     25              ;DATA ERROR
```

```
025640                                   2$:
025640  005303                                   DEC     R3                 ;DEC CHARACTER COUNT
025642  001364                                   BNE     1$                 ;BR IF NOT DONE THIS MESSAGE
025644  012703  000020                           MOV     #16.,R3            ;RESET CHARACTER COUNT

                                                 ;CHECK TO SEE THAT IN BCC MATCH IS SET
                                                 ;AND THAT THE BCC WAS RECEIVED CORRECTLY

025650  004737  027156                           JSR     PC,INRDY           ;WAIT FOR INRDY
025654  104414                                   ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025656  021204                                   021204                     ;GET FIRST HALF OF CRC
025660  116137  000004  001252                   MOVB    4(R1),TEMP3        ;PUT IN TEMP3
025666  042737  177400  001252                   BIC     #177400,TEMP3      ;CLEAR HI BYTE
025674  004737  027156                           JSR     PC,INRDY           ;WAIT FOR INRDY
025700  104414                                   ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025702  021244                                   021244
025704  016104  000004                           MOV     4(R1),R4           ;PUT "FOUND" IN R4
025710  042704  000376                           BIC     #376,R4            ;CLEAR UNWANTED BITS
025714  012705  000001                           MOV     #1,R5              ;PUT "EXPECTED" IN R5
025720  120504                                   CMPB    R5,R4              ;IS IN BCC MATCH SET?
025722  001401                                   BEQ     25$
025724  104015                                   HLT     15                 ;IN BCC MATCH ERROR
025726                                   25$:
025726  104414                                   ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025730  021204                                   021204                     ;GET LAST HALF
025732  116137  000004  001251                   MOVB    4(R1),TEMP2+1      ;PUT IN TEMP2
025740  042737  000377  001250                   BIC     #377,TEMP2         ;CLEAR LO BYTE
025746  053737  001250  001252                   BIS     TEMP2,TEMP3        ;16 BIT BCC NOW IN TEMP3
025754  023737  027320  001252                   CMP     CALBCC,TEMP3       ;IS IT CORRECT?
025762  001401                                   BEQ     4$                 ;BR IF OK
025764  104027                                   HLT     27
025766  012702  030106                   4$:     MOV     #FLTDAT,R2         ;RESET MESSAGE POINTER
025772  005300                                   DEC     R0                 ;DECREMENT COUNTER
025774  001307                                   BNE     1$                 ;BR IF NOT DONE
025776  104400                           3$:     SCOPE                      ;SCOPE THIS TEST

                                                 ;*************************** TEST 57 ***************************
                                                 ;*DDCMP CABLE DATA TEST
                                                 ;*THIS TEST LOADS OUT SILO WITH THE FOLLOWING:
                                                 ;*4 SYNCS,59 DATA CHARACTERS,EOM WITH GARBAGE CHARACTER
                                                 ;*THE DATA IS TRANSMITTED OVER THE CABLE USING THE INTERNAL CLOCK
                                                 ;*RECEIVED DATA IS VERIFIED AS IS IN BCC MATCH
                                                 ;*LOOP-BACK CONNECTOR MUST BE ON TO RUN THIS TEST
                                                 ;;**************************************************************

                                                 ;  TEST 57
                                                 ;--------------
026000  012737  000057  001226           TST57:  MOV     #57,TSTNO
026006  012737  003364  001216                   MOV     #.EOP,NEXT
                                                                            ;R1 CONTAINS BASE DMC11 ADDRESS
026014  104412                                   MSTCLR                     ;MASTER CLEAR DMC11
026016  032737  040000  001366                   BIT     #BIT14,STAT1       ;SKIP TEST IF NO
026024  001533                                   BEQ     3$                 ;LOOPBACK CONNECTOR ON
026026  012711  004000                           MOV     #BIT11,(R1)        ;SET LINE UNIT LOOP
026032  004737  027464                           JSR     PC,SYNLD           ;LOAD 2 SYNCS
```

```
026036  004737  027464                     JSR     PC,SYNLD          ;LOAD 2 MORE SYNCS
026042  012737  120001  027316             MOV     #CRC16,XPOLY      ;LOAD POLYNOMIAL FOR SOFT CRC CALC
026050  005037  026100                     CLR     6$                ;CLEAR OLD BCC
026054  012703  000073                     MOV     #59.,R3           ;CHARACTER COUNT
026060  012702  030102                     MOV     #MESDAT,R2        ;R2= POINTER
026064  112237  026076             7$:     MOVB    (R2)+,5$          ;LOAD CHAR FOR SOFT BCC CALC.
026070  004537  027212                     JSR     R5,SIMBCC         ;CALC SOFT BCC
026074  000010                             10                        ;SHIFT COUNT
026076  000000                     5$:     0                         ;CHARACTER
026100  000000                     6$:     0                         ;OLD BCC
026102  013737  027320  026100             MOV     CALBCC,6$         ;LOAD OLD BCC
026110  005303                             DEC     R3                ;DEC COUNT
026112  001364                             BNE     7$                ;BR IF NOT DONE YET
026114  004537  027620                     JSR     R5,MESLD          ;LOAD SILO
026120  030102                             MESDAT                    ;MESSAGE ADDRESS
026122  000073                             59.                       ;CHARACTER COUNT
026124  004737  027574                     JSR     PC,EOM            ;LOAD AN EOM
026130  004737  026350                     JSR     PC,OCOR           ;WAIT FOR OCOR
026134  005011                             CLR     (R1)              ;CLEAR LINE UNIT LOOP
026136  012700  000073                     MOV     #59.,R0           ;R0= CHARACTER COUNT
026142  012702  030102                     MOV     #MESDAT,R2        ;LOAD MESSAGE POINTER IN R2
026146  004737  027156             1$:     JSR     PC,INRDY          ;WAIT FOR INRDY
026152  104414                             ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026154  021204                             021204                    ;GET DATA FROM IN SILO
026156  016104  000004                     MOV     4(R1),R4          ;PUT CHARACTER IN "FOUND"
026162  112205                             MOVB    (R2)+,R5          ;PUT "EXPECTED" IN R5
026164  120504                             CMPB    R5,R4             ;IS RECEIVED DATA CORRECT
026166  001401                             BEQ     2$                ;BR IF OK
026170  104025                             HLT     25                ;DATA ERROR
026172                             2$:
026172  005300                             DEC     R0                ;DECREMENT COUNTER
026174  001364                             BNE     1$                ;BR IF NOT DONE

                                           ;CHECK TO SEE THAT IN BCC MATCH IS SET
                                           ;AND THAT THE BCC WAS RECEIVED CORRECTLY

026176  004737  027156                     JSR     PC,INRDY          ;WAIT FOR INRDY
026202  104414                             ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026204  021204                             021204                    ;GET FIRST HALF OF CRC
026206  116137  000004  001252             MOVB    4(R1),TEMP3       ;PUT IN TEMP3
026214  042737  177400  001252             BIC     #177400,TEMP3     ;CLEAR HI BYTE
026222  004737  027156                     JSR     PC,INRDY          ;WAIT FOR INRDY
026226  104414                             ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026230  021244                             021244
026232  016104  000004                     MOV     4(R1),R4          ;PUT "FOUND" IN R4
026236  042704  000376                     BIC     #376,R4           ;CLEAR UNWANTED BITS
026242  012705  000001                     MOV     #1,R5             ;PUT "EXPECTED" IN R5
026246  120504                             CMPB    R5,R4             ;IS IN BCC MATCH SET?
026250  001401                             BEQ     25$
026252  104015                             HLT     15                ;IN BCC MATCH ERROR
026254                             25$:
026254  104414                             ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026256  021204                             021204                    ;GET LAST HALF
026260  116137  000004  001251             MOVB    4(R1),TEMP2+1     ;PUT IN TEMP2
026266  042737  000377  001250             BIC     #377,TEMP2        ;CLEAR LO BYTE
026274  053737  001250  001252             BIS     TEMP2,TEMP3       ;16 BIT BCC NOW IN TEMP3
```

```
026302  023737  027320  001252               CMP     CALBCC,TEMP3     ;IS IT CORRECT?
026310  001401                               BEQ     3$               ;BR IF OK
026312  104027                               HLT     27
026314  104400                       3$:     SCOPE                    ;SCOPE THIS TEST


                                     ;SUBROUTINES
                                     ;-----------

026316                               GETSI:
                                             ;THIS SUBROUTINE READS LU 17, AND PUTS IT INTO NITCH.
                                             ;NITCH IS ROTATED LEFT UNTILL THE SI BIT IS IN CARRY

026316  104414                               ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026320  021364                               021364                   ;PORT4←LU 17
026322  017737  153064  026346               MOV     @DMPO4,NITCH     ;STORE LU 17
026330  106137  026346                       ROLB    NITCH
026334  106137  026346                       ROLB    NITCH
026340  106137  026346                       ROLB    NITCH            ;PUT SI IN THE CARRY BIT
026344  000207                               RTS     PC
026346  000000                       NITCH:  0


026350                               OCOR:
                                             ;THIS SUBROUTINE SPINS ON OCOR

026350  104414                               ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026352  021364                               021364                   ;PORT4←LU 17
026354  032777  000020  153030               BIT     #BIT4,@DMPO4     ;IS OCOR SET?
026362  001772                               BEQ     OCOR             ;BR IF NO
026364  000207                               RTS     PC               ;OK OCOR IS SET, GO BACK


026366                               SYNC:
                                             ;THIS SUBROUNTINE LOADS THE SILO WITH THE NUMBER OF SYNC
                                             ;CHARACTERS PASSED TO IT IN THE WORD AFTER THE JSR CALL
                                             ;AND A NON-SYNC CHARACTER (301)

026366  013637  001246                       MOV     @(SP)+,TEMP1     ;GET COUNT
026372  062746  000002                       ADD     #2,-(SP)         ;ADJUST STACK
026376  012761  000026  000004               MOV     #26,4(R1)        ;LOAD PORT4
026404  104414                               ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026406  122114                               122114                   ;LOAD SYNC REGISTER
026410  004737  026502               1$:     JSR     PC,OUTRDY        ;WAIT FOR OUTRDY
026414  012761  000001  000004               MOV     #1,4(R1)         ;LOAD PORT4
026422  104414                               ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026424  122111                               122111                   ;SET SOM
026426  012761  000026  000004               MOV     #26,4(R1)        ;LOAD PORT4
026434  104414                               ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026436  122110                               122110                   ;LOAD OUT DATA
026440  005337  001246                       DEC     TEMP1            ;ALL DONE?
026444  001361                               BNE     1$               ;BR IF NOT
026446  004737  026502                       JSR     PC,OUTRDY        ;WAIT FOR OUTRDY
026452  005061  000004                       CLR     4(R1)            ;LOAD PORT4
026456  104414                               ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026460  122111                               122111                   ;SET SOM
```

```
026462  012761  000301  000004               MOV     #301,4(R1)         ;LOAD PORT4
026470  104414                               ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026472  122110                               122110                     ;LOAD OUT DATA
026474  004737  026350                       JSR     PC,OCOR            ;WAIT FOR OCOR
026500  000207                               RTS     PC


026502                              OUTRDY:
                                             ;THIS SUBROUTINE SPINS ON OUT READY

026502  005037  001256                       CLR     TEMP5              ;CLEAR TIMER
026506                              1$:
026506  104414                               ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026510  021224                               021224                     ;PORT4←LU11
026512  032777  000020  152672               BIT     #BIT4,@DMP04       ;IS OUT RDY SET?
026520  001004                               BNE     2$                 ;BR IF YES
026522  005237  001256                       INC     TEMP5              ;INC TIMER
026526  001367                               BNE     1$                 ;KEEP CHECKING IF NOT DONE
026530  104036                               HLT     36                 ;ERROR, OUT READY NOT SET
026532  000207                      2$:      RTS     PC


026534                              CHAR:
                                             ;THIS SUBROUTINE LOADS THE SILO WITH 3 SYNCS
                                             ;AND THE CHARACTER PASSED TO IT.

026534  013637  001250                       MOV     @(SP)+,TEMP2       ;GET CHARACTER
026540  062746  000002                       ADD     #2,-(SP)           ;ADJUST STACK
026544  012737  000003  001246               MOV     #3,TEMP1           ;SET FOR 3 SYNCS
026552  012761  000026  000004               MOV     #26,4(R1)          ;LOAD PORT4
026560  104414                               ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026562  122114                               122114                     ;LOAD SYNC REGISTER
026564  004737  026502              1$:      JSR     PC,OUTRDY          ;WAIT FOR OUTRDY
026570  012761  000001  000004               MOV     #1,4(R1)           ;LOAD PORT4
026576  104414                               ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026600  122111                               122111                     ;SET SOM
026602  012761  000026  000004               MOV     #26,4(R1)          ;LOAD PORT4
026610  104414                               ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026612  122110                               122110                     ;LOAD OUT DATA
026614  005337  001246                       DEC     TEMP1              ;ALL DONE?
026620  001361                               BNE     1$                 ;BR IF NOT
026622  004737  026502                       JSR     PC,OUTRDY          ;WAIT FOR OUTRDY
026626  013761  001250  000004               MOV     TEMP2,4(R1)        ;LOAD PORT4
026634  104414                               ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026636  122110                               122110                     ;LOAD OUT DATA
026640  004737  026350                       JSR     PC,OCOR            ;WAIT FOR OCOR
026644  000207                               RTS     PC


026646                              CHARSD:
                                             ;THIS SUBROUTINE LOADS THE SILO WITH THE CHARACTER PASSED TO IT.

026646  013637  001250                       MOV     @(SP)+,TEMP2       ;GET CHARACTER
026652  062746  000002                       ADD     #2,-(SP)           ;ADJUST STACK
026656  004737  026502                       JSR     PC,OUTRDY          ;WAIT FOR OUTRDY
026662  013761  001250  000004               MOV     TEMP2,4(R1)        ;LOAD PORT4
```

```
026670 104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026672 122110                           122110                  ;LOAD OUT DATA
026674 004737 026502                    JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
026700 104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026702 122110                           122110                  ;LOAD GARBAGE CHAR
026704 004737 026350                    JSR     PC,OCOR         ;WAIT FOR OCOR
026710 000207                           RTS     PC


026712                          SILOLD:
                                        ;THIS SUBROUTINE FILLS THE OUT SILO
                                        ; WITH A BINARY COUNT PATTERN

026712 012737 000073 001250             MOV     #73,TEMP2       ;LOAD COUNT
026720 005737 027152                    TST     SCHAR           ;FIRST TIME HERE?
026724 100470                           BMI     4$              ;BR IF BITSTUFF
026726 001032                           BNE     2$              ;BR IF NO
026730 062737 000002 001250             ADD     #2,TEMP2        ;ADD 2 TO CHARACTER COUNT
026736 012737 000003 001246             MOV     #3,TEMP1        ;SET FOR 3 SYNCS
026744 012761 000026 000004             MOV     #26,4(R1)       ;LOAD PORT4
026752 104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026754 122114                           122114                  ;LOAD SYNC REGISTER
026756 004737 026502            1$:     JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
026762 012761 000001 000004             MOV     #1,4(R1)        ;LOAD PORT4
026770 104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026772 122111                           122111                  ;SET SOM
026774 012761 000026 000004             MOV     #26,4(R1)       ;LOAD PORT4
027002 104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027004 122110                           122110                  ;LOAD OUT DATA
027006 005337 001246                    DEC     TEMP1           ;ALL DONE?
027012 001361                           BNE     1$              ;BR IF NOT
027014 004737 026502            2$:     JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
027020 013761 027152 000004             MOV     SCHAR,4(R1)     ;LOAD PORT4
027026 104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027030 122110                           122110                  ;LOAD OUT DATA
027032 005737 027154                    TST     STUFLG          ;BITSTUFF???
027036 001407                           BEQ     6$              ;BR IF NO
027040 013737 027152 027052             MOV     SCHAR,5$        ;IT IS SDLD SO CHECK BITSTUFFING
027046 004537 027702                    JSR     R5,STFFCL       ;ADD ANY BIT STUFF CLOCK TICKS
027052 000000                   5$:     0                       ;CHARACTER
027054 000010                           10                      ;CHIFT COUNT
027056 005237 027152            6$:     INC     SCHAR           ;NEXT CHARACTER
027062 022737 000400 027152             CMP     #400,SCHAR      ;ALL DONE?
027070 001403                           BEQ     3$
027072 005337 001250                    DEC     TEMP2           ;DECREMENT COUNT
027076 001346                           BNE     2$              ;BR IF NOT DONE
027100 004737 026350            3$:     JSR     PC,OCOR         ;WAIT FOR OCOR
027104 000207                           RTS     PC
027106 005037 027152            4$:     CLR     SCHAR           ;START PATTERN AT ZERO
027112 012737 177777 027154             MOV     #-1,STUFLG      ;SET BITSTUFF FLAG
027120 005037 030100                    CLR     BITCON          ;CLEAR STUFF COUNT
027124 062737 000002 001250             ADD     #2,TEMP2        ;ADD 2 TO CHARACTER COUNT
027132 012761 000001 000004             MOV     #1,4(R1)        ;SET BIT0 IN PORT4
027140 104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027142 122111                           122111                  ;SET SOM!
027144 104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
027146  122110                          122110                    ;LOAD GARBAGE CHAR
027150  000721                          BR      2$                ;GO LOAD SILO
027152  000000                  SCHAR:  0
027154  000000                  STUFLG: 0


027156                          INRDY:
                                        ;THIS SUBROUTINE SPINS ON INRDY
                                        ;IF INRDY FAILS TO SET THE DELAY TIMES OUT AND AN
                                        ;ERROR IS REPORTED. FOR BETTER SCOPE LOOPS THIS
                                        ;DELAY CAN BE MADE SHORTER BY ALTERING THE NUMBER
                                        ;INITIALLY LOADED INTO TEMP1, THE SMALLER THE NUMBER
                                        ;THE SHORTER THE DELAY. 0 IS THE LONGEST DELAY.

027156  012737  000000  001246          MOV     #0,TEMP1          ;SET UP DELAY COUNTER
027164                          1$:
027164  104414                          ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027166  021244                          021244                    ;PORT4+LU12
027170  032777  000020  152214          BIT     #BIT4,@DMP04      ;IS INRDY SET?
027176  001004                          BNE     2$                ;BR IF YES
027200  005237  001246                  INC     TEMP1             ;INC DELAY
027204  001367                          BNE     1$                ;TRY AGAIN
027206  104037                          HLT     37                ;ERROR,NO INRDY
027210  000207                  2$:     RTS     PC                ;RETURN


027212                          SIMBCC:
                                        ;THIS SUBROUTINE CALCULATES THE CRC USING POLYNOMIAL GIVEN
                                        ;IN XPOLY. THE CORRECT CRC IS RETURNED IN CALBCC, AND THE
                                        ;STATE OF THE LSB OF THE BCC IS RETURNED IN THE C BIT.

027212  010046                          MOV     R0,-(SP)          ;SAVE R0 ON STACK
027214  012537  001246                  MOV     (R5)+,TEMP1       ;TEMP1 = SHIFT COUNT
027220  012537  001250                  MOV     (R5)+,TEMP2       ;TEMP2 = CHARACTER
027224  012537  027320                  MOV     (R5)+,CALBCC      ;CALBCC = OLD BCC
027230  013700  027320          1$:     MOV     CALBCC,R0         ;PUT OLD BCC IN R0
027234  000241                          CLC
027236  006037  027320                  ROR     CALBCC            ;SHIFT OLD BCC
027242  006037  001250                  ROR     TEMP2             ;SHIFT CHARACTER
027246  005500                          ADC     R0                ;ADD CHAR CARRY TO OLD BCC
027250  006000                          ROR     R0                ;PUT BIT0 TO CARRY BIT
027252  103011                          BCC     2$                ;CARRY IS FEEDBACK BIT
027254  013700  027316                  MOV     XPOLY,R0          ;IF FEEDBACK = 1
027260  043700  027320                  BIC     CALBCC,R0         ;EXCLUSIVLY OR XPOLY TO CALBCC
027264  043737  027316  027320          BIC     XPOLY,CALBCC
027272  050037  027320                  BIS     R0,CALBCC
027276  005337  001246          2$:     DEC     TEMP1             ;DEC SHIFT COUNT
027302  001352                          BNE     1$                ;BR IF NOT DONE
027304  013700  027320                  MOV     CALBCC,R0         ;PUT RESULT IN R0
027310  006000                          ROR     R0                ;SHIFT BIT0 TO CARRY
027312  012600                          MOV     (SP)+,R0          ;RESTORE R0
027314  000205                          RTS     R5                ;RETURN
027316  000000                  XPOLY:  0
027320  000000                  CALBCC: 0
        000200                  LRC8=200
        120001                  CRC16=120001
```

```
                102010                                  CRC.CCITT=102010


        027322                                  19800   BCCLD:
                                                19900           ;THIS SUBROUTINE LOADS THE OUT SILO WITH 2 SYNCS
                                                20000           ;WITH SOM SET, AND ONE CHARACTER PASSED TO IT
                                                20100           ;WITH THE SOM BIT CLEAR (ENABLE CRC)
                                                20200
        027322  013637  001250                  20300           MOV     @(SP)+,TEMP2            ;GET CHARACTER
        027326  062746  000002                  20400           ADD     #2,-(SP)                ;ADJUST STACK
        027332  012737  000002  001246          20500           MOV     #2,TEMP1                ;SET FOR 2 SYNCS
        027340  012761  000026  000004          20600           MOV     #26,4(R1)               ;LOAD PORT4
        027346  104414                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
        027350  122114                          20800           122114                          ;LOAD SYNC REGISTER
        027352  004737  026502                  20900   1$:     JSR     PC,OUTRDY               ;WAIT FOR OUTRDY
        027356  012761  000001  000004          21000           MOV     #1,4(R1)                ;LOAD PORT4
        027364  104414                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
        027366  122111                          21200           122111                          ;SET SOM
        027370  012761  000026  000004          21300           MOV     #26,4(R1)               ;LOAD PORT4
        027376  104414                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
        027400  122110                          21500           122110                          ;LOAD OUT DATA
        027402  005337  001246                  21600           DEC     TEMP1                   ;ALL DONE?
        027406  001361                          21700           BNE     1$                      ;BR IF NOT
        027410  004737  026502                  21800           JSR     PC,OUTRDY               ;WAIT FOR OUTRDY
        027414  013761  001250  000004          21900           MOV     TEMP2,4(R1)             ;LOAD PORT4
        027422  104414                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
        027424  122110                          22100           122110                          ;LOAD OUT DATA
        027426  004737  026350                  22200           JSR     PC,OCOR                 ;WAIT FOR OCOR
        027432  000207                          22300           RTS     PC
                                                22400
                                                22500
        027434                                          GETQO:
                                                                ;THIS SUBROUTINE READS THE STATE OF THE TRANSMIT
                                                                ;BCC LSB AND PUTS IT IN THE CARRY BIT

        027434  104414                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
        027436  021364                                          021364                          ;PORT4←LU-17
        027440  106177  151746                                  ROLB    @DMP04                  ;PUT QO IN CARRY
        027444  000207                                          RTS     PC                      ;RETURN


        027446                                          GETQI:
                                                                ;THIS SUBROUTINE READS THE STATE OF THE RECEIVE
                                                                ;BCC LSB AND PUTS IT IN THE CARRY BIT

        027446  104414                                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
        027450  021364                                          021364                          ;PORT4←LU-17
        027452  106177  151734                                  ROLB    @DMP04                  ;PUT QO IN CARRY
        027456  106177  151730                                  ROLB    @DMP04                  ;PUT QI IN CARRY
        027462  000207                                          RTS     PC                      ;RETURN


        027464                                  22800   SYNLD:
                                                22900           ;THIS SUBROUTINE LOADS OUT SILO WITH
                                                23000           ;2 SYNC CHARACTERS WITH SOM SET
                                                23100
```

```
4631  027464  012737  000002  001246  23200          MOV     #2,TEMP1            ;LOAD COUNTER FOR 2 SYNCS
4632  027472  012761  000026  000004  23300          MOV     #26,4(R1)           ;PORT4+26
4633  027500  104414                                ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4634  027502  122114                        23500    122114                      ;LOAD SYNC REG
4635  027504  004737  026502                23600  1$:  JSR     PC,OUTRDY        ;WAIT FOR OUTRDY
4636  027510  012761  000001  000004  23700          MOV     #1,4(R1)            ;LOAD PORT4
4637  027516  104414                                ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4638  027520  122111                        23900    122111                      ;SET SOM
4639  027522  012761  000026  000004  24000          MOV     #26,4(R1)           ;PORT+26
4640  027530  104414                                ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4641  027532  122110                        24200    122110                      ;LOAD OUT DATA WITH SYNC
4642  027534  005337  001246                24300    DEC     TEMP1               ;DECREMENT COUNTER
4643  027540  001361                        24400    BNE     1$                  ;BR IF NOT DONE
4644  027542  000207                        24500    RTS     PC                  ;RETURN
4645                                        24600
4646                                        24700
4647  027544                                24800  SOM:
4648                                        24900    ;THIS SUBROUTINE LOADS SOM AND OUT DATA WITH A
4649                                        25000    ;GARBAGE CHARACTER (0)
4650                                        25100
4651  027544  004737  026502                25200    JSR     PC,OUTRDY           ;WAIT FOR OUTRDY
4652  027550  012761  000001  000004  25300          MOV     #1,4(R1)            ;PORT4+1
4653  027556  104414                                ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4654  027560  122111                        25500    122111                      ;SET SOM
4655  027562  005061  000004                25600    CLR     4(R1)               ;CLEAR DATA CHAR
4656  027566  104414                                ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4657  027570  122110                        25800    122110                      ;LOAD GARBAGE CHARACTER
4658  027572  000207                        25900    RTS     PC                  ;RETURN
4659                                        26000
4660                                        26100
4661  027574                                26200  EOM:
4662                                        26300    ;THIS SUBROUTINE LOADS EOM AND OUT DATA WITH A
4663                                        26400    ;GARBAGE CHARACTER (2) TO ENABLE TRANSMISSION OF BCC
4664                                        26500
4665  027574  004737  026502                26600    JSR     PC,OUTRDY           ;WAIT FOR OUTRDY
4666  027600  012761  000002  000004  26700          MOV     #2,4(R1)            ;PORT4+2
4667  027606  104414                                ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4668  027610  122111                        26900    122111                      ;SET EOM
4669  027612  104414                                ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4670  027614  122110                        27100    122110                      ;LOAD GARBAGE CHARACTER
4671  027616  000207                        27200    RTS     PC                  ;RETURN
4672                                        27300
4673                                        27400
4674  027620                                27500  MESLD:
4675                                        27600    ;THIS SUBROUTINE LOADS SILO WITH MESSAGE
4676                                        27700    ;THE FIRST ARUGUMENT IS THE ADDRESS OF THE MESSAGE
4677                                        27800    ;THE SECOND ARGUMENT IS THE NUMBER OF CHARACTERS IN THE MESSAGE
4678                                        27900
4679  027620  010046                        28000    MOV     R0,-(SP)            ;SAVE R0
4680  027622  012500                        28100    MOV     (R5)+,R0            ;R0=MESSAGE POINTER
4681  027624  012537  001246                28200    MOV     (R5)+,TEMP1         ;TEMP1=CHARACTER COUNT
4682  027630  004737  026502                28300  1$:  JSR     PC,OUTRDY        ;WAIT FOR OUT RDY
4683  027634  112061  000004                28400    MOVB    (R0)+,4(R1)         ;LOAD PORT4 WITH CHARACTER
4684  027640  104414                                ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4685  027642  122110                        28600    122110                      ;LOAD OUT DATA SILO
4686  027644  005337  001246                28700    DEC     TEMP1               ;DEC CHAR COUNT
```

```
027650  001367                           BNE     1$                      ;BR IF NOT DONE
027652  004737  026350                   JSR     PC,OCOR                 ;WAIT FOR OCOR
027656  012600                           MOV     (SP)+,R0                ;RESTORE R0
027660  000205                           RTS     R5                      ;RETURN


027662                           CLRIO:
                                         ;THIS SUBROUTINE SETS IN CLR AND OUT CLR TO
                                         ;CLEAR THE TRANSMIT AND RECEIVE BCC REGISTERS

027662  012761  000200  000004           MOV     #BIT7,4(R1)             ;LOAD PORT4
027670  104414                           ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027672  122112                           122112                          ;SET IN CLR!
027674  104414                           ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027676  122111                           122111                          ;SET OUT CLR!
027700  000207                           RTS     PC                      ;RETURN


027702                           STFFCL:
                                         ;THIS SUBROUTINE ADDS ANY NECESSSARY BIT STUFF CLOCK TICKS
                                         ;FIRST ARGUMENT IS CHAR, SECOND ARGUMENT IS SHIFT COUNT.

027702  010046                           MOV     R0,-(SP)                ;SAVE R0
027704  012500                           MOV     (R5)+,R0                ;PUT CHAR IN R0
027706  012537  001252                   MOV     (R5)+,TEMP3             ;PUT SHIFT COUNT IN TEMP3
027712  106000                   1$:     RORB    R0                      ;LOOK AT NEXT BIT
027714  103403                           BCS     2$                      ;BR IF A MARK
027716  005037  030100                   CLR     BITCON                  ;IT WAS A SPACE, CLEAR 1'S COUNTER
027722  000412                           BR      3$                      ;CONTINUE
027724  005237  030100           2$:     INC     BITCON                  ;INC CONSECUTIVE 1'S COUNTER
027730  022737  000005  030100           CMP     #5,BITCON               ;IS IT 5 YET?
027736  001004                           BNE     3$                      ;BR IF NO
027740  005037  030100                   CLR     BITCON                  ;YES!  SO START AGAIN
027744  104415  000001                   DATACLK,        1               ;GIVE EXTRA TICK TO STUFF ZERO
027750  005337  001252           3$:     DEC     TEMP3                   ;DEC SHIFT COUNT
027754  001356                           BNE     1$                      ;BR IF NOT DONE
027756  012600                           MOV     (SP)+,R0                ;RESTORE R0
027760  000205                           RTS     R5                      ;RETURN


027762                           STFFCK:
                                         ;THIS SUBROUTINE CHECKS TO SEE IF TRANSMITTER
                                         ;IS STUFFING ZEROS WHEN IT SHOULD. FIRST ARGUMENT
                                         ;IS THE CHARACTER, SECOND ARGUMENT IS SHIFT COUNT.

027762  010046                           MOV     R0,-(SP)                ;SAVE R0
027764  012500                           MOV     (R5)+,R0                ;PUT CHAR IN R0
027766  012537  001252                   MOV     (R5)+,TEMP3             ;PUT SHIFT COUNT IN TEMP3
027772  106000                   1$:     RORB    R0                      ;SHIFT OUT NEXT BIT
027774  103403                           BCS     2$                      ;BR IF IT IS A MARK
027776  005037  030100                   CLR     BITCON                  ;IT WAS A SPACE, CLEAR 1'S COUNTER
030002  000416                           BR      3$                      ;CONTINUE
030004  005237  030100           2$:     INC     BITCON                  ;INC CONSECUTIVE I'S COUNTER
030010  022737  000005  030100           CMP     #5,BITCON               ;5 IN A ROW YET?
030016  001010                           BNE     3$                      ;BR IF NO
030020  005037  030100                   CLR     BITCON                  ;YES, SO START OVER
```

```
030024  104415  000001          34400           DATACLK,         1           ;EXTRA TICK TO STUFF ZERO
030030  004737  026316          34500           JSR      PC,GETSI            ;LOOK AT WINDOW
030034  103001                  34600           BCC      3$                  ;IS IT A ZERO, BR IF YES
030036  104030                  34700           HLT      30                  ;NO, ERROR ZERO WAS NOT STUFFED
030040  005337  001252          34800   3$:     DEC      TEMP3               ;DEC SHIFT COUNT
030044  001352                  34900           BNE      1$                  ;BR IF NOT DONE
030046  012600                  35000           MOV      (SP)+,R0            ;RESTORE R0
030050  000205                  35100           RTS      R5                  ;RETURN
                                35200
                                35300
030052                          35400   CTSDLY:
                                35500           ;THIS SUBROUTINE WASTES TIME UNTIL CTS SETS,
                                35600           ;BUT HOPEFULLY NOT SO LONG THAT THE SILO RUNS OUT
                                35700
030052  010046                  35800           MOV      R0,-(SP)            ;SAVE R0
030054  012700  000032          35900           MOV      #32,R0              ;LOAD R0 WITH COUNT
030060  027777  151120  151116  36000   1$:     CMP      @TKCSR,@TKCSR       ;WASTE TIME
030066  005300                  36100           DEC      R0                  ;DECREMENT COUNTER
030070  001373                  36200           BNE      1$                  ;DO IT AGAIN IF NOT = 0
030072  012600                  36300           MOV      (SP)+,R0            ;RESTORE R0
030074  000207                  36400           RTS      PC                  ;RETURN
                                36500
                                36600
030076  000176                  36700   FLAG:   ↑B<01111110>                 ;FLAG CHARACTER
030100  000000                  36800   BITCON: 0
030102     000     125     252  36900   MESDAT: .BYTE    0,125,252,377
030105     377
030106     001     002     004  37000   FLTDAT: .BYTE    1,2,4,10,20,40,100,200,376,375,373,367,357,337,277,177
030111     010     020     040
030114     100     200     376
030117     375     373     367
030122     357     337     277
030125     177
030126     100     140     160  37100   STUFDT: .BYTE    100,140,160,170,3,300,174,176,177,1
030131     170     003     300
030134     174     176     177
030137     001
030140     363     347     317  37200           .BYTE    363,347,317,200,0,377,377,377,200,37
030143     200     000     377
030146     377     377     200
030151     037
                                37300   .EVEN
030152  046377  047111  020105  00300   EM1:    .ASCIZ   <377>/LINE UNIT INITALIZATION TEST/
030210  046377  047111  020105  00400   EM2:    .ASCIZ   <377>↑LINE UNIT REGISTER READ/ONLY TEST↑
030253     377  044514  042516  00500   EM3:    .ASCIZ   <377>↑LINE UNIT REGISTER WRITE/READ TEST↑
030317     377  044514  042516  00600   EM4:    .ASCIZ   <377>/LINE UNIT INTERNAL CLOCK FAILURE/
030361     377  051124  047101  00700   EM5:    .ASCIZ   <377>/TRANSMITTER DATA ERROR/
030411     377  042522  042503  00800   EM6:    .ASCIZ   <377>/RECEIVER TEST/
030430  051377  041505  044505  00900   EM7:    .ASCIZ   <377>/RECEIVER DATA ERROR/
030455     377  047515  042504  01000   EM10:   .ASCIZ   <377>/MODEM SIGNAL ERROR/
030501     377  051124  047101  01100   EM11:   .ASCIZ   <377>/TRANSMITTER CRC ERROR/
030530  051377  041505  044505  01200   EM12:   .ASCIZ   <377>/RECEIVER CRC ERROR/
030554  044777  020116  041502  01300   EM13:   .ASCIZ   <377>/IN BCC MATCH ERROR (LU REG 12)/
030614  052377  040522  051516  01400   EM14:   .ASCIZ   <377>/TRANSMITTER FAILED TO GO TO MARK STATE/
030664  041777  041101  042514  01500   EM15:   .ASCIZ   <377>/CABLE DATA TEST/
030705     377  046106  043501  01600   EM16:   .ASCIZ   <377>/FLAG ERROR/
```

```
030721      377  051124  047101   EM17:    .ASCIZ   <377>/TRANSMITTER FAILED TO STUFF A ZERO/
030765      377  053523  052111   EM20:    .ASCIZ   <377>/SWITCH PAC TEST/
031006   040777  047502  052122   EM21:    .ASCIZ   <377>/ABORT ERROR/
031023      377  051124  047101   EM22:    .ASCIZ   <377>/TRANSMITTER ERROR/
031046   044377  046101  020106   EM23:    .ASCIZ   <377>/HALF DUPLEX TEST/
031070   047777  052125  051040   EM24:    .ASCIZ   <377>/OUT READY NOT SET/
031113      377  047111  051040   EM25:    .ASCIZ   <377>/IN READY NOT SET/

031135      377  054105  042520   DH1:     .ASCIZ   <377>/EXPECTED  FOUND/
031156   042777  050130  041505   DH2:     .ASCIZ   <377>/EXPECTED  FOUND  LU-REGISTER/
031214   041777  040510  040522   DH3:     .ASCIZ   <377>/CHARACTER    BIT THAT FAILED/
031252   041777  051117  042522   DH4:     .ASCIZ   <377>/CORRECT CRC    BIT THAT FAILED/
031312   042777  050130  041505   DH5:     .ASCIZ   <377>/EXPECTED   FOUND    SHIFT/
031344   042777  050130  041505   DH6:     .ASCIZ   <377>/EXPECTED   FOUND    CHARACTER   SHIFT/
031412   041377  047514  045503   DH7:     .ASCIZ   <377>/BLOCK END NOT SET/
031435      377  052122  020123   DH10:    .ASCIZ   <377>/RTS DID NOT CLEAR/
                                  .EVEN

031460   000002                   DT1:     2
031462      003     007                    .BYTE    3,7
031464   001272                            SAVR5
031466      003     002                    .BYTE    3,2
031470   001270                            SAVR4
031472   000003                   DT2:     3
031474      003     007                    .BYTE    3,7
031476   001272                            SAVR5
031500      003     010                    .BYTE    3,10
031502   001270                            SAVR4
031504      003     002                    .BYTE    3,2
031506   001264                            SAVR2
031510   000002                   DT3:     2
031512      003     017                    .BYTE    3,17
031514   001272                            SAVR5
031516      002     002                    .BYTE    2,2
031520   001266                            SAVR3
031522   000002                   DT4:     2
031524      006     021                    .BYTE    6,21
031526   027320                            CALBCC
031530      002     002                    .BYTE    2,2
031532   001266                            SAVR3
031534   000003                   DT5:     3
031536      001     011                    .BYTE    1,11
031540   001300                            ZERO
031542      001     011                    .BYTE    1,11
031544   001302                            ONE
031546      002     002                    .BYTE    2,2
031550   001260                            SAVR0
031552   000003                   DT6:     3
031554      001     011                    .BYTE    1,11
031556   001302                            ONE
031560      001     011                    .BYTE    1,11
031562   001300                            ZERO
031564      002     002                    .BYTE    2,2
031566   001260                            SAVR0
031570   000004                   DT7:     4
031572      001     011                    .BYTE    1,11
```

```
031574  001300                                  ZERO
031576     001     011                          .BYTE   1,11
031600  001302                                  ONE
031602     003     007                          .BYTE   3,7
031604  001272                                  SAVR5
031606     002     001                          .BYTE   2,1
031610  001266                                  SAVR3
031612  000004                          DT10:   4
031614     001     011                          .BYTE   1,11
031616  001302                                  ONE
031620     001     011                          .BYTE   1,11
031622  001300                                  ZERO
031624     003     007                          .BYTE   3,7
031626  001272                                  SAVR5
031630     002     001                          .BYTE   2,1
031632  001266                                  SAVR3
031634  000002                          DT11:   2
031636     003     007                          .BYTE   3,7
031640  030076                                  FLAG
031642     002     002                          .BYTE   2,2
031644  001266                                  SAVR3
031646  000002                          DT12:   2
031650     006     004                          .BYTE   6,4
031652  027320                                  CALBCC
031654     006     002                          .BYTE   6,2
031656  001252                                  TEMP3

031660                                  .ERRTAB:
031660  000000                                  0
031662  000000                                  0
031664  000000                                  0
031666  030152                          EM1
031670  031156                          DH2     ;HLT    1
031672  031472                          DT2
031674  030210                          EM2
031676  031156                          DH2     ;HLT    2
031700  031472                          DT2
031702  030253                          EM3
031704  031156                          DH2     ;HLT    3
031706  031472                          DT2
031710  030317                          EM4
031712  000000                          0       ;HLT    4
031714  000000                          0
031716  030361                          EM5
031720  031156                          DH2     ;HLT    5
031722  031472                          DT2
031724  030361                          EM5
031726  031214                          DH3     ;HLT    6
031730  031510                          DT3
031732  030411                          EM6
031734  031135                          DH1     ;HLT    7
031736  031460                          DT1
031740  030430                          EM7
031742  031135                          DH1     ;HLT    10
031744  031460                          DT1
031746  030455                          EM10
```

```
        031750  031135                    12900   DH1     ;HLT    11
        031752  031460                    13000   DT1
        031754  030501                    13100   EM11
        031756  031312                    13200   DH5     ;HLT    12
        031760  031534                    13300   DT5
        031762  030530                    13400   EM12
        031764  031312                    13500   DH5     ;HLT    13
        031766  031534                    13600   DT5
        031770  030501                    13700   EM11
        031772  031252                    13800   DH4     ;HLT    14
        031774  031522                    13900   DT4
        031776  030554                    14000   EM13
        032000  000000                    14100   0       ;HLT    15
        032002  000000                    14200   0
        032004  030501                    14300   EM11
        032006  031312                    14400   DH5     ;HLT    16
        032010  031552                    14500   DT6
        032012  030530                    14600   EM12
        032014  031312                    14700   DH5     ;HLT    17
        032016  031552                    14800   DT6
        032020  030501                    14900   EM11
        032022  031344                    15000   DH6     ;HLT    20
        032024  031570                    15100   DT7
        032026  030501                    15200   EM11
        032030  031344                    15300   DH6     ;HLT    21
        032032  031612                    15400   DT10
        032034  030530                    15500   EM12
        032036  031344                    15600   DH6     ;HLT    22
        032040  031570                    15700   DT7
        032042  030530                    15800   EM12
        032044  031344                    15900   DH6     ;HLT    23
        032046  031612                    16000   DT10
        032050  030614                    16100   EM14
        032052  000000                    16200   0       ;HLT    24
        032054  000000                    16300   0
        032056  030664                    16400   EM15
        032060  031135                    16500   DH1     ;HLT    25
        032062  031460                    16600   DT1
        032064  030705                    16700   EM16
        032066  031214                    16800   DH3     ;HLT    26
        032070  031634                    16900   DT11
        032072  030530                    17000   EM12
        032074  031135                    17100   DH1     ;HLT    27
        032076  031646                    17200   DT12
        032100  030721                    17300   EM17
        032102  000000                    17400   0       ;HLT    30
        032104  000000                    17500   0
        032106  030765                    17600   EM20
        032110  031135                    17700   DH1     ;HLT    31
        032112  031460                    17800   DT1
        032114  031006                    17900   EM21
        032116  031412                    18000   DH7     ;HLT    32
        032120  000000                    18100   0
        032122  031006                    18200   EM21
        032124  031214                    18300   DH3     ;HLT    33
        032126  031510                    18400   DT3
        032130  031023                 18500   EM22
        032132  031435                 18600   DH10      ;HLT      34
        032134  000000                 18700   0
        032136  031046                 18800   EM23
        032140  031156                 18900   DH2       ;HLT      35
        032142  031472                 19000   DT2
        032144  031070                 19100   EM24
        032146  000000                 19200   0         ;HLT      36
        032150  000000                 19300   0
        032152  031113                 19400   EM25
        032154  000000                 19500   0         ;HLT      37
        032156  000000                 19600   0
        032160  030411                 19700   EM6
        032162  031156                 19800   DH2       ;HLT      40
        032164  031472                 19900   DT2
        032166  030361                 20000   EM5
        032170  031312                 20100   DH5       ;HLT      41
        032172  031534                 20200   DT5
        032174  030554                 20300   EM13
        032176  031135                 20400   DH1       ;HLT      42
        032200  031460                 20500   DT1
                                       20600
                                       20700
        032202                         20800   CORMAX:
                000001                 21300   .END
```

```
DZDME   MACY11 30(1046)  11-JUL-77  11:40  PAGE 95
DZDME.P11     12-MAY-77 14:18           CROSS REFERENCE TABLE -- USER SYMBOLS

ADRCNT= 004373            879*    915*    924#
AUDONE  003024            569     608     613     659#
AUSTRT  002446            568#    663
AUTO.S  010512            526    1368#
BCCLD   027322           2987    3016    3062    3091    3137    3166    3212    3241    4578#
BINWRD  004714            965*    968*    969    1006#
BITCON  030100           4513*   4714*   4716*   4717    4719*   4737*   4739*   4740    4742*   4767#
BIT0  = 000001             95#   1155    1156
BIT1  = 000002             94#    531    1149    1155    1156    1449    1532    1545
BIT10 = 002000             85#   1513    1524
BIT11 = 004000             84#   2053    2086    2130    2188    2239    2290    2341    2393    2451    2479    2507    2535
                         2565    2613    2650    2687    2724    2764    2845    2940    2981    3056    3131    3206    3279
                         3345    3416    3513    3577    3824    4095    4159    4199    4293
BIT12 = 010000             83#   1464    1543
BIT13 = 020000             82#   1467    1516    1547    1756    2889    2915    2936
BIT14 = 040000             81#    781    1478    1480    1547    1558    1758    2891    2938    4197    4291
BIT15 = 100000             80#    485     572     575     596     599    1451    1519
BIT2  = 000004             93#    531     712    1156
BIT3  = 000010             92#   1549    1556    2917
BIT4  = 000020             91#   1139    1162    1164    2166    2574    2622    2659    2696    2733    4160    4379    4421
                         4537
BIT5  = 000040             90#   1661    1761    2109    2219    2270    2321    2372    2806    4112
BIT6  = 000100             89#   1144    1145    1551    2488    2544
BIT7  = 000200             88#   1145    1265    1661    2138    2579    4697
BIT8  = 000400             87#   1523    1538    1540    1553    1555    1561    1564    1622
BIT9  = 001000             86#   1521    1523    1535    1561    1564    1620    1622
BM      007054           1187#   1492
BRLVL   012252           1617    1627    1635    1647#
BRW     003730            718     804#
BRX     003732            719     805#
CALBCC  027320           3006    3035    3081    3110    3156    3185    3231    3260    3316    3382    3441    3462    3606
                         3627    3657    3678    3754    3797    3854    3875    3924    3945    4021    4075    4211    4267
                         4305    4351    4553*   4554    4556*   4562    4563*   4564*   4567    4572#   4785
CHAR    026534           2566    2614    2651    2688    2725    2846    4429#
CHARSD  026646           4456#
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#
CLRIO   027662           2982    3011    3057    3086    3132    3161    3207    3236    4693#
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  032202           4785#
CRAM    006606           1187#   1430
CRC.CC= 102010           4575#
CRC16 = 120001           2984    3013    3059    3088    3134    3163    3209    3238    3284    3350    3435    3600    3651
                         3848    3918    4202    4296    4574#
```

K09

```
DZDME   MACY11 30(1046)  11-JUL-77  11:40  PAGE 96                                                      PAGE:  0114
DZDME.P11    12-MAY-77 14:18            CROSS REFERENCE TABLE -- USER SYMBOLS

CREAM    001320           195#    483*   1289*   1290    1292*   1296
CSR      006510          1187#   1395
CSRMAP   010514          1370#
CTSDLY   030052          4753#
CYCLE    010060           720     761     762    1280#
DATABP   005216          1049*   1052    1074    1077#
DATACL=  104415           241#   2093    2137    2141    2201    2202    2216    2252    2253    2267    2303    2304    2318
                         2354    2355    2369    2411    2414    2455    2483    2511    2539    2568    2616    2653    2690
                         2727    2766    2767    2810    2811    2848    2989    2990    3018    3019    3064    3065    3093
                         3094    3139    3140    3168    3169    3214    3215    3243    3244    3298    3300    3364    3366
                         3428    3442    3463    3481    3488    3490    3523    3593    3607    3628    3658    3679    3697
                         3704    3706    3717    3841    3855    3876    3896    3925    3946    3964    3971    3973    3984
                         4104    4109    4115    4169    4720    4743
DATAHD   005204          1048*   1070    1073#
DELAY =  104413           237#
DEVADR   004370           877*    912     922#
DEVTAB   003010           583     648#
DH1      031135          4785#
DH10     031435          4785#
DH2      031156          4785#
DH3      031214          4785#
DH4      031252          4785#
DH5      031312          4785#
DH6      031344          4785#
DH7      031412          4785#
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    1678*
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
DMP04    001412           265#   1133*   1139    1176    1181    1311*   1312*   1313    4366    4379    4421    4537    4612*
                         4622*   4623*
DMP06    001414           266#   1150*   1313*   1314*
DMRLVL   001376           259#   1316*   1317*   1318
DMRVEC   001374           258#    768    1299*   1300*   1316
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*
DMTVEC   001400          260#   1318*   1319*   1320
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      031460         4785#
DT10     031612         4785#
DT11     031634         4785#
DT12     031646         4785#
```

M09

```
DZDME    MACY11 30(1046)  11-JUL-77  11:40  PAGE 98                                                    PAGE:  0116
DZDME.P11    12-MAY-77 14:18          CROSS REFERENCE TABLE -- USER SYMBOLS

DT2       031472          4785#
DT3       031510          4785#
DT4       031522          4785#
DT5       031534          4785#
DT6       031552          4785#
DT7       031570          4785#
EM1       030152          4785#
EM10      030455          4785#
EM11      030501          4785#
EM12      030530          4785#
EM13      030554          4785#
EM14      030614          4785#
EM15      030664          4785#
EM16      030705          4785#
EM17      030721          4785#
EM2       030210          4785#
EM20      030765          4785#
EM21      031006          4785#
EM22      031023          4785#
EM23      031046          4785#
EM24      031070          4785#
EM25      031113          4785#
EM3       030253          4785#
EM4       030317          4785#
EM5       030361          4785#
EM6       030411          4785#
EM7       030430          4785#
EOM       027574          3425     3521     3586     3590     3833     3838     4217     4221     4225     4311     4661#
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          1082     1087#
FLAG      030076          4766#    4785
FLOAT     002536           585#     591
FLTDAT    030106          4205     4215     4219     4223     4230     4270     4770#
FY        002566           594#     605      609      615
```

N09

```
DZDME    MACY11 30(1046)  11-JUL-77  11:40  PAGE 99                                                    PAGE:  0117
DZDME.P11     12-MAY-77 14:18          CROSS REFERENCE TABLE -- USER SYMBOLS

GETQI    027446          3026    3030    3101    3105    3176    3180    3251    3255    3372    3376    4616#
GETQO    027434          2997    3001    3072    3076    3147    3151    3222    3226    3306    3310    4606#
GETSI    026316          2205    2209    2256    2260    2307    2311    2358    2362    2417    2421    3445    3449    3466
                         3470    3482    3491    3610    3614    3631    3635    3661    3665    3682    3686    3698    3707
                         3858    3862    3879    3883    3899    3903    3928    3932    3949    3953    3965    3974    4360#
                         4744
HALTS    005222          1033    1079#
HILIM    004366           876*    903     921#
ICOUNT   001222           159#    793     798*
INBUF    007502           846     882    1202#
INCHAR   010020          1231    1259#
INIFLG   001324           201#    507     527     534*
INRDY    027156          2768    2812    2849    2857    2865    3524    3537    3542    3720    3734    3739    3763    3777
                         3782    3987    4001    4006    4027    4041    4055    4060    4116    4126    4134    4231    4247
                         4252    4316    4331    4336    4525#
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    1779*   1795*   1821*   1837*   1863*   1886*   1921*   1941*
                         2978*   3010*   3053*   3085*   3128*   3160*   3203*   3235*
LOKFLG   001326           203#
LOLIM    004364           875*    905     920#
LPCNT    001224           160#    792*    793     796*
LRC8  =  000200          4573#
LSTERR   001234           164#    490*    732*   1034    1036*   1124*
LUTYPE=  ****** U           1    4545
MASKX    001244           172#
MASTEK   006142          1058    1187#
MCRLF    005672           831     954    1054    1055    1063    1187#   1328    1390
MCSRX    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#
MESDAT   030102          2949    3417    3423    3514    3519    3578    3584    3588    3644    3719    3762    3825    3831
                         3836    3911    3986    4040    4096    4101    4106    4166    4299    4309    4315    4768#
MESLD    027620          2948    3422    3518    3583    3587    3830    3835    4100    4105    4165    4214    4218    4222
                         4308    4674#
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#
MQM      005666           861    1187#   1352    1427    1437    1447    1462    1476
MR       005755           723    1187#   1346
MRESET=  004000            96#
MSTCLR=  104412           235#   1126    1725    1749    1781    1823    1865    1923    1974    1996    2018    2052    2085
```

B10

```
DZDME    MACY11 30(1046)  11-JUL-77  11:40  PAGE 100                                                          PAGE:  0118
DZDME.P11     12-MAY-77 14:18              CROSS REFERENCE TABLE -- USER SYMBOLS

                          2129     2187     2238     2289     2340     2392     2450     2478     2506     2534     2563     2611     2648
                          2685     2722     2759     2800     2844     2888     2935     2980     3055     3130     3205     3278     3344
                          3412     3512     3573     3820     4094     4157     4196     4290
MTITLE   001000            136#     511
MTSTN    006130           1059     1187#    1330
MTSTPC   006031           1187#
MVECX    006100            739     1187#
NEXT     001216            157#     799     1092     1676*    1700*    1723*    1747*    1778*    1820*    1862*    1920*    1972*    1994*
                          2016*    2050*    2083*    2127*    2185*    2236*    2287*    2338*    2390*    2448*    2476*    2504*    2532*
                          2561*    2609*    2646*    2683*    2720*    2757*    2798*    2842*    2886*    2933*    2977*    3052*    3127*
                          3202*    3276*    3342*    3410*    3510*    3571*    3818*    4092*    4155*    4194*    4288*
NITCH    026346           4366*    4367*    4368*    4369*    4371#
NOACT    007154            523     1187#    1282
NODEV    002674            577      616#
NUM      006450           1187#    1382
OCOR     026350           2092     2136     2198     2249     2300     2351     2410     3297     3363     3426     3522     3591     3839
                          4103     4108     4168     4226     4312     4374#    4380     4410     4452     4468     4509     4602     4688
OK       002646            603      610#     639
ONE      001302            187#    4785
OUTRDY   026502           2189     2194     2240     2245     2291     2296     2342     2347     2397     2401     2406     2429     3421
                          3517     3582     3829     4099     4164     4394     4403     4414#    4439     4448     4461     4465     4485
                          4494     4589     4598     4635     4651     4665     4682
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#
POPRO  = 012600             72#    1086
POP1SP=  005726             70#
POP2SP=  022626             74#     801
PRIO     006547           1187#    1413
PS     = 177776             63#     476*     711*    1617*    1627*
PUSHRO=  010046             71#    1083
PUSH1S=  005746             69#
PUSH2S=  024646             73#
QV.FLG   001327            204#     482*     750*     790
RESREG   005220           1075     1078#
RESTAR   005350           1108     1114#
```

C10

```
DZDME   MACY11 30(1046)  11-JUL-77  11:40  PAGE 101                                                   PAGE: 0119
DZDME.P11    12-MAY-77 14:18          CROSS REFERENCE TACLE -- USER SYMBOLS

RESTRT  003534        749     753     761#
RESOS = 104407        229#   1078
RETURN  001214        156#    492*    720*    724     761*    799*    803    1092*   1095    1127    1345*   1355*   1357
ROMCLK= 104414        239#   1134    1137    1174    1179    1680    1703    1727    1751    1784    1786    1797    1799
                     1826    1828    1839    1841    1871    1873    1892    1894    1928    1930    1946    1948    1975
                     1997    2021    2030    2055    2057    2061    2088    2090    2095    2105    2132    2134    2139
                     2143    2153    2163    2191    2196    2217    2242    2247    2268    2293    2298    2319    2344
                     2349    2370    2399    2403    2408    2431    2456    2484    2512    2540    2570    2580    2582
                     2618    2627    2655    2664    2692    2701    2729    2738    2769    2807    2813    2850    2858
                     2866    2895    2898    2907    2910    2942    2946    2954    3287    3291    3295    3325    3353
                     3357    3361    3391    3525    3538    3543    3552    3721    3735    3740    3749    3764    3778
                     3783    3792    3988    4002    4007    4016    4028    4042    4056    4061    4070    4110    4117
                     4127    4135    4161    4170    4232    4248    4253    4262    4317    4332    4337    4346    4364
                     4377    4392    4396    4399    4405    4408    4419    4437    4441    4444    4450    4463    4466
                     4483    4487    4490    4496    4516    4518    4535    4587    4591    4594    4600    4610    4620
                     4633    4637    4640    4653    4656    4667    4669    4684    4698    4700
RUN     001316        193#    485*   1286*   1287*   1294
SAVACT  001312        191#    671    1588*
SAVNUM  001314        192#    479*    748*    751*   1581*
SAVPC   001276        185#    622*    642     929*   1099
SAVR0   001260        178#    938*    943    4785
SAVR1   001262        179#    628*    937*    944
SAVR2   001264        180#    936*    945    4785
SAVR3   001266        181#    935*    946    4785
SAVR4   001270        182#    934*    947    4785
SAVR5   001272        183#    933*    948    4785
SAVSP   001274        184#
SAVOS = 104406        227#   1038
SCHAR   027152       2760*   2801*   4477    4495    4500    4504*   4505    4511*   4521#
SCOPE = 104400        215#   1688    1711    1735    1766    1808    1850    1908    1960    1982    2004    2037    2070
                     2114    2171    2222    2273    2324    2375    2436    2464    2492    2520    2548    2597    2634
                     2671    2708    2745    2784    2828    2873    2921    2964    3039    3114    3189    3264    3330
                     3396    3497    3554    3800    4078    4143    4178    4273    4354
SCOP1 = 104401        217#   1794    1807    1836    1849    1882    1903    1937    1955    3009    3038    3084    3113
                     3159    3188    3234    3263
SILOLD  026712       2765    2781    2809    2825    4472#
SIMBCC  027212       2992    3021    3067    3096    3142    3171    3217    3246    3301    3367    3437    3602    3653
                     3850    3920    4207    4301    4545#
SKIP    002632        573     576     597     600     606#
SOFTSW  010052       1229    1268#
SOM     027544       3834    4647#
SPACNT= 004713        964*    988     991*   1005#
SPEED   007340       1187#   1440
STACK = 001200         64#    477     690    1093    1116
STAT    001240        170#
STAT1   001366        251#   1301*   1756    1758    2889    2891    2915    2936    2938    4197    4291
STAT2   001370        252#   1302*   1978    2000
STAT3   001372        253#   1303*
STFFCK  027762       4727#
STFFCL  027702       4501    4705#
STRTSW  001236        169#    513*    516*    517     519     529     531     666     712     721    1323    1347*   1377
                     1601
STUFDT  030126       4776#
STUFLG  027154       2761*   2802*   4498    4512*   4522#
SVOS    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#
SYNC      026366            2453     2481     2509     2537     4384#
SYNLD     027464            3285     3351     3420     3516     3581     3828     4098     4163     4200     4201     4294     4295     4627#
TEMP      001416             271#     971     1118*    1119*    1160*    1165*    1171*    1183*    2019*    2025*    2028*    2034*
TEMP1     001246             173#     537*    1189     1613*    1614*    4389*    4401*    4435*    4446*    4481*    4492*    4533*    4539*
                            4551*    4565*    4585*    4596*    4631*    4642*    4681*    4686*
TEMP2     001250             174#     538*    1191     3751*    3752*    3753     3794*    3795*    3796     4018*    4019*    4020     4072*
                            4073*    4074     4264*    4265*    4266     4348*    4349*    4350     4433*    4449     4459*    4462     4476*
                            4480*    4507*    4514*    4552*    4557*    4583*    4599
TEMP3     001252             175#     540*     566*     618      627*    1193     1386     1389     1501*    3540*    3541*    3737*    3738*
                            3753*    3754     3780*    3781*    3796*    3797     4004*    4005*    4020*    4021     4058*    4059*    4074*
                            4075     4250*    4251*    4266*    4267     4334*    4335*    4350*    4351     4711*    4721*    4734*    4747*
                            4785
TEMP4     001254             176#     541*    1195     1399     1402     1408     1411     1487     1490     1496     1499
TEMP5     001256             177#     542*    1197     1380*    1393*    1599     4417*    4423*
TIMER  =  104416             243#    2059     2569     2617     2654     2691     2728     2897     2909     2944
TKCSR     001204             148#     848     1214     1259     1655     4759
TKDBR     001206             149#     850      856     1217     1219     1261     1657
TLAST  =  026000            1350     4355#
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     1675*    1699*    1722*    1746*    1777*    1819*
                            1861*    1919*    1971*    1993*    2015*    2049*    2082*    2126*    2184*    2235*    2286*    2337*    2389*
                            2447*    2475*    2503*    2531*    2560*    2608*    2645*    2682*    2719*    2756*    2797*    2841*    2895*
                            2932*    2976*    3051*    3126*    3201*    3275*    3341*    3409*    3509*    3570*    3817*    4091*    4154*
                            4193*    4287*
TST1      012320            1337     1355     1675#
TST10     013326            1862     1919#
TST11     013502            1920     1971#
TST12     013544            1972     1993#
TST13     013606            1994     2015#
TST14     013706            2016     2049#
TST15     014006            2050     2082#
TST16     014144            2083     2126#
TST17     014342            2127     2184#
```

```
TST2     012374         1676     1699#
TST20    014524         2185     2235#
TST21    014706         2236     2286#
TST22    015070         2287     2337#
TST23    015252         2338     2389#
TST24    015460         2390     2447#
TST25    015546         2448     2475#
TST26    015636         2476     2503#
TST27    015724         2504     2531#
TST3     012442         1700     1722#
TST30    016014         2532     2560#
TST31    016166         2561     2608#
TST32    016302         2609     2645#
TST33    016420         2646     2682#
TST34    016536         2683     2719#
TST35    016654         2720     2756#
TST36    017004         2757     2797#
TST37    017144         2798     2841#
TST4     012514         1723     1746#
TST40    017304         2842     2885#
TST41    017466         2886     2932#
TST42    017632         2933     2976#
TST43    020146         2977     3051#
TST44    020462         3052     3126#
TST45    020776         3127     3201#
TST46    021312         3202     3275#
TST47    021550         3276     3341#
TST5     012616         1747     1777#
TST50    022006         3342     3409#
TST51    022340         3410     3509#
TST52    022542         3510     3570#
TST53    023642         3571     3817#
TST54    025042         3818     4091#
TST55    025274         4092     4154#
TST56    025412         4155     4193#
TST57    026000         4194     4287#     4355
TST6     012756         1778     1819#
TST60 =  ****** U       4288
TST7     013116         1820     1861#
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     1329     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#
XCSR     003546          738      763#
XERR     003570          744      772#
XHEAD    006224          535     1187#
```

F10

```
DZDME    MACY11 30(1046)  11-JUL-77  11:40  PAGE 104                                                    PAGE:  0122
DZDME.P11    12-MAY-77 14:18          CROSS REFERENCE TABLE -- USER SYMBOLS

XLOC     003022            593*     611*     614      645      658#
XPASS    003562            742      769#
XPOLY    027316           2984*    3013*    3059*    3088*    3134*    3163*    3209*    3238*    3284*    3350*    3435*    3600*    3651*
                          3848*    3918*    4202*    4296*    4561     4563     4571#
XSTATQ   007454            544     1187#
XTSTN    005330           1060     1100#
XVEC     003554            740      766#
ZERO     001300            186#    4785
$COD   = ****** U            1
$CRAP  = 177777              1#    1665#    1668     1671#    1689#    1692     1695#    1712#    1715     1718#    1736#    1739     1742#
                          1767#    1770     1773#    1809#    1812     1815#    1851#    1854     1857#    1909#    1912     1915#    1961#
                          1964     1967#    1983#    1986     1989#    2005#    2008     2011#    2038#    2041     2045#    2071#    2074
                          2078#    2115#    2118     2122#    2172#    2175     2180#    2223#    2226     2231#    2274#    2277     2282#
                          2325#    2328     2333#    2376#    2379     2385#    2437#    2440     2443#    2465#    2468     2471#    2493#
                          2496     2499#    2521#    2524     2527#    2549#    2552     2556#    2598#    2601     2604#    2635#    2638
                          2641#    2672#    2675     2678#    2709#    2712     2715#    2746#    2749     2752#    2785#    2788     2793#
                          2829#    2832     2837#    2874#    2877     2881#    2922#    2925     2928#    2965#    2968     2972#    3040#
                          3043     3047#    3115#    3118     3122#    3190#    3193     3197#    3265#    3268     3271#    3331#    3334
                          3337#    3397#    3400     3405#    3498#    3501     3505#    3555#    3558     3566#    3801#    3804     3813#
                          4079#    4082     4087#    4144#    4147     4150#    4179#    4182     4189#    4274#    4277     4283#
$ENDAD   003522            123      509      755#    1079
$N     = 000057              1#    1665     1671     1673     1678#    1689     1695     1697     1702#    1712     1718     1720     1725
                          1726#    1736     1742     1744     1749     1750#    1767     1773     1775     1781     1782#    1809     1815
                          1817     1823     1824#    1851     1857     1859     1865     1866#    1909     1915     1917     1923     1924#
                          1961     1967     1969     1974     1975#    1983     1989     1991     1996     1997#    2005     2011     2013
                          2018     2019#    2038     2045     2047     2052     2053#    2071     2078     2080     2085     2086#    2115
                          2122     2124     2129     2130#    2172     2180     2182     2187     2188#    2223     2231     2233     2238
                          2239#    2274     2282     2284     2289     2290#    2325     2333     2335     2340     2341#    2376     2385
                          2387     2392     2393#    2437     2443     2445     2450     2451#    2465     2471     2473     2478     2479#
                          2493     2499     2501     2506     2507#    2521     2527     2529     2534     2535#    2549     2556     2558
                          2563     2564#    2598     2604     2606     2611     2612#    2635     2641     2643     2648     2649#    2672
                          2678     2680     2685     2686#    2709     2715     2717     2722     2723#    2746     2752     2754     2759
                          2760#    2785     2793     2795     2800     2801#    2829     2837     2839     2844     2845#    2874     2881
                          2883     2888     2889#    2922     2928     2930     2935     2936#    2965     2972     2974     2980     2981#
                          3040     3047     3049     3055     3056#    3115     3122     3124     3130     3131#    3190     3197     3199
                          3205     3206#    3265     3271     3273     3278     3279#    3331     3337     3339     3344     3345#    3397
                          3405     3407     3412     3413#    3498     3505     3507     3512     3513#    3555     3566     3568     3573
                          3574#    3801     3813     3815     3820     3821#    4079     4087     4089     4094     4095#    4144     4150
                          4152     4157     4158#    4179     4189     4191     4196     4197#    4274     4283     4285     4290     4291#
                          4355#
$S     = 000061              1#    1676     1678#    1700     1702#    1723     1726#    1747     1750#    1778     1782#    1820     1824#
                          1862     1866#    1920     1924#    1972     1975#    1994     1997#    2016     2019#    2050     2053#    2083
                          2086#    2127     2130#    2185     2188#    2236     2239#    2287     2290#    2338     2341#    2390     2393#
                          2448     2451#    2476     2479#    2504     2507#    2532     2535#    2561     2564#    2609     2612#    2646
                          2649#    2683     2686#    2720     2723#    2757     2760#    2798     2801#    2842     2845#    2886     2889#
                          2933     2936#    2977     2981#    3052     3056#    3127     3131#    3202     3206#    3276     3279#    3342
                          3345#    3410     3413#    3510     3513#    3571     3574#    3818     3821#    4092     4095#    4155     4158#
                          4194     4197#    4288     4291#
$Y     = 000017              1#     207#     215      217#     219#     221#     223#     225#     227#     229#     231#     233#     235#
                           237#     239#     241#     243#     245#
.      = 032202            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#
```

```
DZDME    MACY11 30(1046)  11-JUL-77  11:40  PAGE 105                                                         PAGE:  0123
DZDME.P11     12-MAY-77 14:18          CROSS REFERENCE TABLE -- USER SYMBOLS

                              357#    515     525     657#    675     1110    1120    1168#   1203#   1205#   1260    1263    1284
                              1378    1422    1542    1546    1593    1624    1656    1659    2890    2916    2937    3492    3708
                              3975
.BEGIN   003152               690#
.CNVRT   004472               234     955#
.CONVR   004466               232     954#
.DATAC   005552               242     1159#
.DELAY   005436               238     1132#
.EOP     003364               731#    4288
.ERRTA   031660               1046    4785#
.HLT     004750               115     1026#
.INSTE   004154               224     861#
.INSTR   004050               222     840#
.INST1   004070               844#    864
.MSG     004072               842*    845#
.MSTCL   005466               236     1143#
.PARAM   004174               226     872#
.PFAIL   005336               113     478     1107#   1115
.RESOS   004434               230     943#
.ROMCL   005504               240     1148#
.SAVOS   004374               228     929#
.SCOPE   003576               216     779#
.SCOP1   003736               218     811#
.START   002002               132     476#    492     1247
.TIMER   005616               244     1170#
.TRPSR   004716               117     1014#
.TRPTA   001330               214#    1019
.TYPE    003766               220     822#
```

H10

```
DZDME    MACY11 30(1046)  11-JUL-77  11:40  PAGE 107                                                        PAGE:  0124
DZDME.P11     12-MAY-77 14:18             CROSS REFERENCE TABLE -- MACRO NAMES

DMEND       1#     725
DMERNT      1#
HLT        75#    1687    1710    1734    1765    1793    1806    1835    1848    1881    1902    1936    1954    1981    2003
         2027     2036    2068    2102    2112    2150    2160    2163    2207    2211    2221    2258    2262    2272    2309
         2313     2323    2360    2364    2374    2419    2423    2463    2491    2519    2547    2577    2589    2596    2625
         2633     2662    2670    2699    2707    2736    2744    2775    2819    2856    2864    2872    2905    2920    2963
         2999     3003    3028    3032    3074    3078    3103    3107    3149    3153    3178    3182    3224    3228    3253
         3257     3308    3312    3374    3378    3447    3451    3468    3472    3484    3493    3531    3550    3612    3616
         3633     3637    3663    3667    3684    3688    3700    3709    3727    3747    3756    3770    3790    3799    3860
         3864     3881    3885    3901    3905    3930    3934    3951    3955    3967    3976    3994    4014    4023    4034
         4048     4068    4077    4114    4123    4133    4141    4177    4238    4260    4269    4323    4344    4353    4425
         4541     4746
$ABORT      1#
$AUTO       1#     547
$BCC        1#    2965    3040    3115    3190
$BINCR      1#    3265    3331
$BINWI      1#    2376
$BUFFE      1#    1199
$CDATA      1#    4179    4274
$CLOCK      1#    2005
$COMP       1#    2572    2584    2591    2620    2629    2657    2666    2694    2703    2731    2740    2852    2860    2868
         2900     2912    3545    3742    3785    4009    4063    4255    4339
$CRC        1#    2982    3011    3057    3086    3132    3161    3207    3236
$CRCSH      1#    3279    3345
$CYCLE      1#    1271
$EMPTY      1#    4079
$EOP        1#     725
$FINI       1#    4355
$FLAG       1#
$FLOAT      1#    1867    1887    1925    1942
$GETPA      1#
$HALF       1#    4144
$HEADE      1#
$INACT      1#    2437    2465    2493    2521
$INIT       1#
$LINEI      1#    1851    1909
$LU1        1#    1665    1689    1712    1736
$LU12       1#    1767
$LU17       1#    1809
$MARHI      1#
$MARK       1#    2829
$MATCH      1#    3534    3730    3773    3997    4051    4243    4327
$MOCK       1#
$MODEM      1#    2874
$MSG        1#    1187
$MULT       1#
$PATTE      1#    2746    2785
$PFAIL      1#    1103
$QOQI       1#    4606    4616
$QUEST      1#    1381    1394    1403    1482    1491
$RAMCL      1#    1131
$RCLK       1#    1134    1137    1174    1179    1680    1703    1727    1751    1784    1786    1797    1799    1826    1829
         1839     1841    1871    1873    1892    1894    1928    1930    1946    1948    1975    1997    2021    2030    2055
         2057     2061    2088    2090    2095    2105    2132    2134    2139    2143    2153    2163    2191    2196    2217
         2242     2247    2268    2293    2298    2319    2344    2349    2370    2399    2403    2408    2431    2456    2494
         2512     2540    2570    2580    2582    2618    2627    2655    2664    2692    2701    2729    2738    2769    2907
```

I10

```
DZDME   MACY11 30(1046)  11-JUL-77  11:40  PAGE 108                                                          PAGE:  0125
DZDME.P11    12-MAY-77 14:18          CROSS REFERENCE TABLE -- MACRO NAMES

         2813    2850    2858    2866    2895    2898    2907    2910    2942    2946    2953    3287    3291    3295    3325
         3353    3357    3361    3391    3525    3538    3543    3552    3721    3735    3740    3749    3764    3778    3783
         3792    3988    4002    4007    4016    4028    4042    4056    4061    4070    4110    4117    4127    4135    4161
         4170    4232    4248    4253    4262    4317    4332    4337    4346    4364    4377    4392    4396    4399    4405
         4408    4419    4437    4441    4444    4450    4463    4466    4483    4487    4490    4496    4516    4518    4535
         4587    4591    4594    4600    4610    4620    4633    4637    4640    4653    4656    4667    4669    4684    4698
         4700
$RCRC      1#    3498
$REC       1#    2549    2598    2635    2672    2709
$SCOPE     1#     775
$SIMBC     1#    4545
$SINAC     1#
$SOFTC     1#    1207
$STUFF     1#
$SWPAC     1#    1961    1983
$TCHAR     1#    3416    3513    3577    3824
$TCRC      1#    3397    3555    3801
$TRANW     1#    3435    3600    3651    3848    3918
$TRAN1     1#    2038    2071    2115
$TRPDE     1#     215     217     219     221     223     225     227     229     231     233     235     237     239     241
          243
$TSTN      1#    1673    1697    1720    1744    1775    1817    1859    1917    1969    1991    2013    2047    2080    2124
         2182    2233    2284    2335    2387    2445    2473    2501    2529    2558    2606    2643    2680    2717    2754
         2795    2839    2883    2930    2974    3049    3124    3199    3273    3339    3407    3507    3568    3815    4089
         4152    4191    4285
$VARIA     1#     134
$WINDO     1#    2172    2223    2274    2325
$XZ        1#    1665    1671    1689    1695    1712    1718    1736    1742    1767    1773    1809    1815    1851    1857
         1909    1915    1961    1967    1983    1989    2005    2011    2038    2045    2071    2078    2115    2122    2172
         2180    2223    2231    2274    2282    2325    2333    2376    2385    2437    2443    2465    2471    2493    2499
         2521    2527    2549    2556    2598    2604    2635    2641    2672    2678    2709    2715    2746    2752    2785
         2793    2829    2837    2874    2881    2922    2928    2965    2972    3040    3047    3115    3122    3190    3197
         3265    3271    3331    3337    3397    3405    3498    3505    3555    3566    3801    3813    4079    4087    4144
         4150    4179    4189    4274    4283
$ZEROS     1#

. ABS.  032202     000

 ERRORS DETECTED:  0

 DZDME,DZDME/SOL/CRF←IPLUTL,DZDME
 RUN-TIME: 16 22 1 SECONDS
 RUN-TIME RATIO: 779/39=19.5
 CORE USED:  31K  (61 PAGES)
```