# DMC11

BITSTUFF MODE LINE TESTS

MD-11-DZDMF-B

EP-DZDMF-B-DL-A

COPYRIGHT © 76-77

FICHE 1 OF 1

AUG 1977

digital

MADE IN USA

IDENTIFICATION

| | |
|---|---|
| PRODUCT CODE: | MAINDEC-11-DZDMF-B-C |
| PRODUCT NAME: | BITSTUFF MODE LINE UNIT TESTS |
| DATE: | MAY 1977 |
| MAINTAINER: | DIAGNCSTICS |
| AUTHOR: | FAY BASHAW |

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

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

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

1. ABSTRACT

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

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

DZDMF 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 BITSTUFF mode. The modem signals are also checked. DZDMF requires a DMC Micro-Processor (M8200 or M3204) 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 'SCOPI') 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

RE‾URN (1214) Contains the address where program will return when iteration count is reachěd 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. EXAMP E: (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 ir 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 a 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 DMC11-AR. Further tests are performed at this point to determine which line unit, if any, is installed, if a loop-back connector is installed and various switch settings on the line unit. THIS IS WHY THE STATUS TABLE MUST BE VERIFIED BY THE USER AND IF ANY OF THE INFORMATION DOES NOT AGREE WITH THE HARDWARE THE DIAGNOSTIC MUST BE RESTARTED AND THE QUESTIONS MUST BE ANSWERED. All DMC11's in the system will be found by the auto-sizer. If it does not find a DMC11 the diagnostic must be restarted and the questions answered.

### 8.5.2 FINDING THE VECTOR AND BR LEVEL

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

## 8.6 SOFTWARE SWITCH REGISTER

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

Control:

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

```
    SWR=XXXXXX NEW?
```

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

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

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

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

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

DOCUMENT
**************
DZDMF LST
**************

```
6       MAINDEC-11-DZDMF-B   DMC11 BITSTUFF 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

2077    *************************** TEST 15 ***************************
        BITSTUFF 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

2125    *************************** 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

2186    *************************** TEST 17 ***************************
        BITSTUFF TRANSMITTER TEST
        SINGLE CLOCK THE CHARACTER 0
        CHECK FLAG AND DATA IN THE BIT WINDOW
        VERIFY EACH BIT POSITION AS IT
        PASSES THE BIT WINDOW (SI BIT)
        ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE

2260    *************************** TEST 20 ***************************
        BITSTUFF TRANSMITTER TEST
        SINGLE CLOCK THE CHARACTER 125
        CHECK FLAG AND DATA IN THE BIT WINDOW
        VERIFY EACH BIT POSITION AS IT
        PASSES THE BIT WINDOW (SI BIT)
        ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE

2334    *************************** TEST 21 ***************************

2335    BITSTUFF TRANSMITTER TEST
        SINGLE CLOCK THE CHARACTER 252
        CHECK FLAG AND DATA IN THE BIT WINDOW
        VERIFY EACH BIT POSITION AS IT
        PASSES THE BIT WINDOW (SI BIT)
        ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
```

```
   2408     ************************** TEST 22 ***************************
            BIT STUFF TEST
            THIS TEST CHECKS ZERO BIT STUFFING OF
             THE TRANSMITTER IN THE BIT WINDOW

   2485     ************************** TEST 23 ***************************
            BITSTUFF TRANSMITTER TEST
            SINGLE CLOCK THE CHARACTER 377
            CHECK FLAG AND DATA IN THE BIT WINDOW
            VERIFY EACH BIT POSITION AS IT
            PASSES THE BIT WINDOW (SI BIT)
            ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE

   2565     ************************** TEST 24 ***************************
            BITSTUFF 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

   2654     ************************** TEST 25 ***************************
            MULTIPLE FLAG AND TRANSMITTER ABORT TEST
            LOAD SILO WITH 5 FLAGS AND A CHAR (000)
            VERIFIY IN THE BIT WINDOW THAT THE FLAGS
            AND DATA ARE CORRECT AND FOLLOWED BY AN ABORT
            SEQUENCE (8 CONTIGUOUS 1'S)

   2729     ************************** TEST 26 ***************************
            LEADING ZEROS TEST
            VERIFY THAT THE SETTING OF SOM AND EOM TOGETHER
            AND THEN SOM ALONE WILL GENERATE 16 LEADING ZEROS
            AND A FLAG,THE CHECK IS MADE USING THE BIT WINDOW

   2789     ************************** TEST 27 ***************************
            BITSTUFF STRIP FLAG TEST
            SET LU LOOP, SINGLE STEP 5 FLAGS,
            VERIFY THAT IN ACTIVE DOES NOT SET

   2821     ************************** TEST 30 ***************************
            BITSTUFF IN ACTIVE TEST
            SET LU LOOP, SINGLE STEP 5 FLAGS AND A NON-FLAG (301)
            VERIFY THAT IN ACTIVE IS SET

   2853     ************************** TEST 31 ***************************
            BITSTUFF IN ACTIVE TEST
            SET LINE UNIT LOOP,SINGLE STEP ONE FLAG AND A CHAR (301)
            VERIFY THAT IN ACTIVE IS SET
```

```
2893    ************************** TEST 32 ***************************
        BITSTUFF IN ACTIVE TEST

2895    SET LU LOOP, SINGLE STEP 2 FLAGS AND A NON-FLAG (301)
        VERIFY THAT IN ACTIVE IS SET

2925    ************************** TEST 33 ***************************
        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

2983    ************************** TEST 34 ***************************
        BITSTUFF BASIC RECEICER TEST
        SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 0
        VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED

3029    ************************** TEST 35 ***************************
        BITSTUFF BASIC RECEICER TEST
        SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 125
        VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED

3075    ************************** TEST 36 ***************************
        BITSTUFF BASIC RECEICER TEST
        SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 252
        VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED

3121    ************************** TEST 37 ***************************
        BITSTUFF BASIC RECEICER TEST
        SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 377
        VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED

3167    ************************** TEST 40 ***************************
        BITSTUFF DATA TEST
        THIS TEST SINGLE STEPS A BINARY COUNT PATTERN
        CHECKING EACH CHARACTER AS IT IS RECEIVED

3212    ************************** TEST 41 ***************************
        BITSTUFF 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

3262    ************************** TEST 42 ***************************
        RECEIVER ABORT TEST
        SINGLE CLOCK 3 FLAGS, A 301, ANOTHER 301 AND 10 EXTRA
        CLOCK TICKS, VERIFY THAT A 301 AND A BLOCK END
        WERE RECEIVED INDICATING THAT THE RECEIVER RECOGINIZED
        THE ABORT SEQUENCE (8 CONTIGUIOUS 1'S)
```

```
3307      *************************** TEST 43 ***************************
          CABLE TURNAROUND TEST
          CLEAR LINE UNIT LOOP, SET DTR
          VERIFY THAT MODEM READY IS SET
          CLEAR DTR, VERIFY THAT MRDY IS CLEARED

3355      *************************** TEST 44 ***************************
          CABLE TURNAROUND TEST
          CLEAR LINE UNIT LOOP, LOAD OUT DATA SILO
          VERIFY THAT ALL MODEM SIGNALS ARE SET

3398      *************************** TEST 45 ***************************

3399      TEST OF CRC OPERATION
          USING THE CRC.CCITT POLYNOMIAL, SINGLE CLOCK THE CHARACTER
          0, VERIFY THE LSB OF THE BCC ON EACH SHIFT
          TEST TRANSMITTER FIRST THEN THE RECEIVER BCC

3480      *************************** TEST 46 ***************************
          TEST OF CRC OPERATION
          USING THE CRC.CCITT POLYNOMIAL, SINGLE CLOCK THE CHARACTER
          377, VERIFY THE LSB OF THE BCC ON EACH SHIFT
          TEST TRANSMITTER FIRST THEN THE RECEIVER BCC

3568      *************************** TEST 47 ***************************
          TEST OF CRC OPERATION
          USING THE CRC.CCITT POLYNOMIAL, SINGLE CLOCK THE CHARACTER
          125, VERIFY THE LSB OF THE BCC ON EACH SHIFT
          TEST TRANSMITTER FIRST THEN THE RECEIVER BCC

3650      *************************** TEST 50 ***************************
          TEST OF CRC OPERATION
          USING THE CRC.CCITT POLYNOMIAL, SINGLE CLOCK THE CHARACTER
          252, VERIFY THE LSB OF THE BCC ON EACH SHIFT
          TEST TRANSMITTER FIRST THEN THE RECEIVER BCC

3732      *************************** TEST 51 ***************************
          TRANSMITTER CRC TEST
          USING THE CRC.CCITT POLYNOMINAL, SINGLE CLOCK A BINARY

3735      COUNT PATTERN, VERIFY THE LSB OF THE TRANSMITTER BCC ON EACH SHIFT

3815      *************************** TEST 52 ***************************
          RECEIVER CRC TEST
          USING THE CRC.CCITT POLYNOMINAL, SINGLE CLOCK A BINARY
          COUNT PATTERN, VERIFY THE LSB OF THE RECEIVER BCC ON EACH SHIFT

3901      *************************** TEST 53 ***************************
          TRANSMITTER BITSTUFF CRC TEST
```

```
3903    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

4038    *************************** TEST 54 ***************************
        RECEIVER BITSTUFF 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

4100    *************************** TEST 55 ***************************
        BITSTUFF EOM FUNCTION TEST
        THIS TEST LOADS OUT SILO WITH: 2 FLAGS,4 CHAR MESSAGE,EOM
        4 CHARACTER MESS,EOM. THE DATA STREAM IS CHECKED TO BE
        4 CHAR,BCC,FLAG,4 CHAR,BCC,FLAG,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

4413    *************************** TEST 56 ***************************
        BITSTUFF EOM FUNCTION TEST
        THIS TEST LOADS OUT SILO WITH: 2 FLAGS,4 CHAR MESSAGE,EOM
        SOM,4 CHAR MESS,EOM. THE DATA STREAM IS CHECKED TO BE
        4 CHAR,BCC,FLAG,4 CHAR,BCC,FLAG,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

4746    *************************** TEST 57 ***************************
        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 CHARACTERS  AND A BLOCK END WERE RECEIVED, AND IN ACTIVE IS CLEAR

4810    *************************** TEST 60 ***************************
        BITSTUFF CABLE DATA TEST
        THIS TEST LOADS OUT SILO WITH THE FOLLOWING:
        2 FLAGS,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

4913    *************************** TEST 61 ***************************
        BITSTUFF CABLE DATA TEST
        THIS TEST LOADS OUT SILO WITH THE FOLLOWING:
        2 FLAGS,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-DZDMF-B   DMC11 BITSTUFF 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-DZDMF-B   DMC11 BITSTUFF 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 STATJS 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-DZDMF-B/<377>
 001025      104  041515  030461           .ASCIZ  /DMC11 BITSTUFF 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 * 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 FRCM 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                                   XSTATO                     ;
002416  000762                                   BR      5$
002420  012700  001500                   1$:     MOV     #DM.MAP,R0         ;R0 POINTS TO STATUS TABLE

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

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

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

```
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                 ;DJP
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

 003364  000005                              .EOP:       RESET                        ;MAKE THE  WORLD CLEAN AGAIN.
 003366  005037  001234                                  CLR      LSTERR              ;CLEAR LAST ERROR PC
 003372  105037  001325                                  CLRB     ERRFLG              ;CLEAR ERROR FLAG
 003376  005237  001230                                  INC      PASCNT              ;UPDATE PASS COUNT
 003402  013777  001230  175570                          MOV      PASCNT,@DISPLAY     ;DISPLAY PASS COUNT
 003410  104402  005733                                  TYPE     ,MEPASS             ;TYPE END PASS
 003414  104402  006072                                  TYPE     ,MCSRX              ;TYPE CSR
 003420  104411  003546                                  CNVRT    ,XCSR               ;SHOW IT
 003424  104402  006100                                  TYPE     ,MVECX              ;TYPE VECTOR
 003430  104411  003554                                  CNVRT    ,XVEC               ;SHOW IT
 003434  104402  006106                                  TYPE     ,MPASSX             ;TYPE PASSES
 003440  104411  003562                                  CNVRT    ,XPASS              ;SHOW IT
 003444  104402  006117                                  TYPE     ,MERRX              ;TYPE ERRORS
 003450  104411  003570                                  CNVRT    ,XERR               ;SHOW IT
 003454  013700  001322                                  MOV      MILK,R0             ;GET POINTER TO PASS COUNT
 003460  013720  001230                                  MOV      PASCNT,(R0)+        ;STORE PASS COUNT FOR THIS DMC11
 003464  013720  001232                                  MOV      ERRCNT,(R0)+        ;STORE ERROR COUNT FOR THIS DMC11
 003470  005337  001314                                  DEC      SAVNUM              ;ARE ALL DEVICES TESTED?
 003474  001017                                          BNE      RESTRT              ;BR IF NO.
 003476  112737  000377  001327                          MOVB     #377,QV.FLG         ;SET THE QUICK VERIFY FLAG
 003504  013737  001310  001314                          MOV      DMNUM,SAVNUM        ;RESTORE THE COUNT
 003512  013701  000042                                  MOV      @#42,R1             ;CHECK FOR ACT-11 OR DDP
 003516  001406                                          BEQ      RESTRT              ;IF NOT, CONTINUE TESTING
 003520  000005                                          RESET                        ;STOP THE SHOW--CLEAR THE WORLD
 003522                                      $ENDAD:
 003522  004711                                          JSR      PC,(R1)
 003524  000240                                          NOP
 003526  000240                                          NOP
 003530  000240                                          NOF
 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  005656                  .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$:     SAV05                   ;SAVE ALL PROC REGISTERS
005030  011605                              MOV     (SP),R5         ;GET THE PC OF ERROR
005032  162705  000002                      SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
005036  011504                              MOV     (R5),R4         ;GET HLT INSTRUCTION
005040  006304                              ASL     R4              ;MULT BY TWO
005042  061504                              ADD     (R5),R4         ;DOUBLE IT
005044  006304                              ASL     R4              ;MULT AGAIN
005046  042704  177001                      BIC     #177001,R4      ;CLEAR JUNK
005052  062704  035372                      ADD     #.ERRTAB,R4     ;GET POINTER
005056  012437  005172                      MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
005062  012437  005204                      MOV     (R4)+,DATAHD    ;GET DATA HEADRER
005066  011437  005216                      MOV     (R4),DATABP     ;GET DATA TABLE
005072  105737  001325                      TSTB    ERRFLG          ;TYPE HEADREER
005076  001403                              BEQ     TYPMSG          ;BR IF YES
005100  005737  005216                      TST     DATABP          ;DOES DATA TABLE EXIST?
005104  001040                              BNE     TYPDAT          ;BR IF YES.
005106  104402  005672              TYPMSG: TYPE    ,MCRLF
005112  104402  005672                      TYPE    ,MCRLF
005116  005737  001220                      TST     LOCK
005122  001402                              BEQ     1$
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  /  ?/
005672  005015     000                  MCRLF:  .ASCIZ   <15><12>
005675     377  053520  020122          MPFAIL: .ASCIZ  <377>/PWR FAILED. RESTART AT TEST /
005733     377  047105  020104          MEPASS: .ASCIZ  <377>/END PASS DZDMF  /
005755     377  000122                  MR:     .ASCIZ  <377>/R/
005760  047377  020117  042504          MERR2:  .ASCIZ  <377>/NO DEVICES PRESENT./
006005     377  047111  052523          MERR3:  .ASCIZ  <377>/INSUFFICIENT DATA!/
006031     377  042524  052123          MTSTPC: .ASCIZ  <377>/TEST PC-/
006043     377  047514  045503          MLOCK:  .ASCIZ  <377>/LOCK ON SELECTED TEST/
006072  051503  035122  000040          MCSRX:  .ASCIZ   /CSR: /
006100  042526  C35103  000040          MVECX:  .ASCIZ  /VEC: /
006106  040520  051523  051505          MPASSX: .ASCIZ  /PASSES: /
006117     105  051122  051117          MERRX:  .ASCIZ  /ERRORS: /
006130  042524  052123  047040          MTSTN:  .ASCIZ  /TEST NO: /
006142  000052                          MASTEK: .ASCIZ  /*/
006144  051777  052105  051440          MNEW:   .ASCIZ  <377>/SET SWITCH REG TO DMC11'S DESIRED ACTIVE.
006217     120  035103  000040          MERRPC: .ASCIZ  /PC: /
006224  020212  020040  020040          XHEAD:  .ASCII  <212>/           MAP OF DMC11 STATUS/
006263     377  020040  020040                  .ASCII  <377>/           -------------------.
006322  020212  050040  020103                  .ASCII  <212>/  PC      CSR     STAT1    STAT2    STAT3
006374  026777  026455  026455                  .ASCIZ  <377>/------  ------  ------  ------   ------./
006450  044377  053517  046440          NUM:    .ASCIZ  <377>/HOW MANY DMC11'S TO BE TESTED?/
006510  041777  051123  040440          CSR:    .ASCIZ  <377>/CSR ADDRESS?/
006526  053377  041505  047524          VEC:    .ASCIZ  <377>/VECTOR ADDRESS?/
006547     377  051102  050040          PRIO:   .ASCIZ  <377>/BR PRIORITY LEVEL? (4,5,6,7)?/
006606  044777  020106  046504          CRAM:   .ASCIZ  <377>/IF DMC HAS CRAM (M8204) TYPE "Y", IF CROM (M8200) TYPE "N"
006704  053777  044510  044103          MODU:   .ASCIZ  <377>/WHICH LINE UNIT? IF NONE TYPE "N", IF M8201 TYPE "1", IF "
007016  051777  044527  041524          LINE:   .ASCIZ  <377>/SWITCH PAC#1 (DDCMP LINE #)?/
007054  051777  044527  041524          BM:     .ASCIZ  <377>/SWITCH PAC#2 (BM873 BOOT ADD)?
007114  044777  020123  044124          CONN:   .ASCIZ  <377>/IS THE LOOP BACK CONNECTOR ON?
007154  047377  020117  042504          NOACT:  .ASCIZ  <377>/NO DEVICES ARE SELECTED/
007205     377  051412  051127          SWMES:  .ASCIZ  <377><12>/SWR= /
007215     116  053505  020077          SWMES1: .ASCIZ  /NEW? /
007223     377  042377  041515          CONERR: .ASCIZ  <377><377>/DMC11 FOUND AT NON-STANDARD ADDRESS  PC:
007277     377  054105  042520          CNERR:  .ASCIZ  <377>/EXPECTED  FOUND/
007320  024040  046504  024503          DMCM:   .ASCIZ  / (DMC)
007330  024040  046513  024503          KMCM:   .ASCIZ  / (KMC) /
007340  042377  041515  030461          SPEED:  .ASCIZ  <377>/DMC11-AR(REMOTE,LOW SPEED) OR DMC11-AL(LOCAL,HIGH SPEED) "
                                        .EVEN
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                       CKSWRS: 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$:     TST     (R0)+           ;POP R0
010460  020027  031460                          CMP     R0,#TLAST+10    ;AT END YET?
010464  001351                                  BNE     5$              ;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 ROM0
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$:    MOV     #BIT9,(R1)        ;SET ROMI
011352  012761  100417  000006                  MOV     #100417,6(R1)     ;PUT INSTRUCTION IN SEL6
011360  012711  001400                          MOV     #BIT9!BIT8,(R1)   ;CLOCK INSTRUCTION (MICRO PROC PC TO 0)
011364  012711  002000                          MOV     #BIT10,(R1)       ;SET ROM0
011370  022761  000626  000006                  CMP     #626,6(R1)        ;IS IT LOCAL CROM
011376  001411                                  BEQ     23$               ;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                                  TEMPS

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  167344                          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     #65$,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$:
013630  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=53C4
013632  021364                                  021364                  ;PORT4←LU-17
013634  032761  000002  000004                  BIT     #2,4(R1)        ;IS CLOCK BIT SET?
013642  001004                                  BNE     2$              ;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=53C4
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  014022  001216                  MOV     #TST15,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
013722  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
013724  005061  000004                          CLR     4(R1)           ;CLEAR PORT4
013730  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=53C4
```

```
013732  122117                                   122117                       ;PUT LINE UNIT IN BITSTUFF MODE
013734  004737  033374                           JSR      P?,CLRIO            ;DO THIS AFTER MODE IS SET
013740  012711  004000                           MOV      #BIT11,(R1)         ;SET LINE UNIT LOOP
013744  012761  000001  000004                   MOV      #1,4(R1)            ;LOAD PORT4 WITH BIT0
013752  104414                                   ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013754  ?22111                                   122111                       ;SET SOM
013756  104414                                   ??MCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013760  122110                                      '110                      ;LOAD OUT DATA SILO
013762  104416  000002                             ER,    2                   ;WAIT FOR OCOR
013766  012702  000017                           MUV      #17,R2              ;SAVE ADDRESS FOR TYPEOUT
013772  104414                                   ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013774  021364                                   021364                       ;PORT4+LU 17
013776  016104  000004                           MOV      4(R1),R4            ;PUT "FOUND" IN R4
014002  042704  000357                           BIC      #357,R4             ;CLEAR UNWANTED BITS
014006  012705  000020                           MOV      #20,R5              ;PUT "EXPECTED" IN R5
014012  120504                                   CMPB     R5,R4               ;IS OCOR SET?
014014  001401                                   BEQ      1$                  ;BR IF YES
014016  104005                                   HLT      5
014020                                   1$:
014020  104400                                   SCOPE                        ;SCOPE THIS TEST


                                                 ;**************************** TEST 15 ****************************
                                                 ;*BITSTUFF 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
                                                 ;--------------
014022  012737  000015  001226           TST15:  MOV      #15,TSTNO
014030  012737  014174  001216                   MOV      #TST16,NEXT
                                                                              ;R1 CONTAINS BASE DMC11 ADDRESS
014036  104412                                   MSTCLR                       ;MASTER CLEAR DMC11
014040  005061  000004                           CLR      4(R1)               ;CLEAR PORT4
014044  104414                                   ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014046  122117                                   122117                       ;PUT LINE UNIT IN BITSTUFF MODE
014050  004737  033374                           JSR      PC,CLRIO            ;DO THIS AFTER MODE IS SET
014054  012711  004000                           MOV      #BIT11,(?1)         ;SET LINE UNIT LOOP
014060  012761  000001  000004                   MOV      #1,4(R1)            ;LOAD PORT4 WITH BIT0
014066  104414                                   ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014070  122111                                   122111                       ;SET SOM
014072  104414                                   ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014074  122110                                   122110                       ;LOAD OUT DATA SILO
014076  004737  032044                           JSR      PC,OCOR             ;WAIT FOR OCOR
014102  104415  000002                           DATACLK,     2               ;CLOCK DATA FOUR TIMES
014106  012702  000011                           MOV      #11,R2              ;SAVE ADDRESS FOR TYPEOUT
014112  104414                                   ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014114  021224                                   021224                       ;PORT4+LU 11
014116  016104  000004                           MOV      4(R1),R4            ;PUT "FOUND" IN R4
014122  042704  000257                           BIC      #257,R4             ;CLEAR UNWANTED BITS
014126  012705  000120                           MOV      #120,R5             ;PUT "EXPECTED" IN R5
014132  120504                                   CMPB     R5,R4               ;IS ACTIVE SET?
014134  001401                                   BEQ      1$                  ;BR IF YES
014136  104005                                   HLT      5
```

```
014140                                  1$:
014140  012702  000013                          MOV     #13,R2          ;SAVE ADDRESS FOR TYPEOUT
014144  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014146  021264                                  021264                  ;PORT4+LU 13
014150  016104  000004                          MOV     4(R1),R4        ;PUT EXPECTED IN R4
014154  042704  000337                          BIC     #337,R4         ;CLEAR UNWANTED BITS
014160  012705  000040                          MOV     #BIT5,R5        ;PUT "EXPECTED" IN R5, RTS SHOULD BE SET
014164  120504                                  CMPB    R5,R4           ;IS RTS OK?
014166  001401                                  BEQ     2$              ;BR IF YES
014170  104005                                  HLT     5               ;RTS ERROR
014172                                  2$:
014172  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
                                                ;--------------
014174  012737  000016  001226          TST16:  MOV     #16,TSTNO
014202  012737  014406  001216                  MOV     #TST17,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
014210  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
014212  005061  000004                          CLR     4(R1)           ;CLEAR PORT4
014216  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014220  122117                                  122117                  ;PUT LINE UNIT IN BITSTUFF MODE
014222  004737  033374                          JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
014226  012711  004000                          MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
014232  012761  000001  000004                  MOV     #1,4(R1)        ;LOAD PORT4 WITH BIT0
014240  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014242  122111                                  122111                  ;SET SOM
014244  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014246  122110                                  122110                  ;LOAD OUT DATA SILO
014250  004737  032044                          JSR     PC,OCOR         ;WAIT FOR OCOR
014254  104415  000002                          DATACLK,        2       ;CLOCK DATA FOUR TIMES
014260  012761  000200  000004                  MOV     #BIT7,4(R1)     ;SET BIT7 IN PORT4
014266  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014270  122111                                  122111                  ;SET OUT CLEAR
014272  104415  000001                          DATACLK,        1       ;GIVE A TICK TO CLEAR RTS
014276  012702  000017                          MOV     #17,R2          ;SAVE ADDRESS FOR TYPEOUT
014302  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014304  021364                                  021364                  ;PORT4+LU 17
014306  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
014312  042704  000357                          BIC     #357,R4         ;CLEAR UNWANTED BITS
014316  005005                                  CLR     R5              ;PUT "EXPECTED" IN R5
014320  120504                                  CMPB    R5,R4           ;IS OCOR CLEARED?
014322  001401                                  BEQ     1$              ;BR IF YES
014324  104005                                  HLT     5
014326                                  1$:
014326  012702  000013                          MOV     #13,R2          ;SAVE ADDRESS FOR TYPEOUT
014332  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014334  021264                                  021264                  ;PORT4+LU 13
```

```
014336  016104  000004                  MOV     4(R1),R4        ;PUT EXPECTED IN R4
014342  042704  000337                  BIC     #337,R4         ;CLEAR UNWANTED BITS
014346  005005                          CLR     R5              ;PUT "EXPECTED" IN R5. RTS SHOULD BE CLEARED
014350  120504                          CMPB    R5,R4           ;IS RTS OK?
014352  001401                          BEQ     2$              ;BR IF YES
014354  104005                          HLT     5               ;RTS ERROR
014356                          2$:
014356  012702  000011                  MOV     #11,R2          ;SAVE ADDRESS FOR TYPEOUT
014362  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014364  021224                          021224                  ;PORT4+LU11
014366  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
014372  012705  000020                  MOV     #BIT4,R5        ;ONLY OUT READY SHOULD BE SET
014376  120504                          CMPB    R5,R4           ;IS ACTIVE CLEAR?
014400  001401                          BEQ     3$              ;BR IF YES
014402  104005                          HLT     5               ;ERROR ACTIVE NOT CLEARED
014404                          3$:
014404  104400                          SCOPE                   ;SCOPE THIS TEST


                                        ;*************************** TEST 17 ***************************
                                        ;*BITSTUFF TRANSMITTER TEST
                                        ;*SINGLE CLOCK THE CHARACTER 0
                                        ;*CHECK FLAG AND DATA IN THE BIT WINDOW
                                        ;*VERIFY EACH BIT POSITION AS IT
                                        ;*PASSES THE BIT WINDOW (SI BIT)
                                        ;*ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
                                        ;:**************************************************************

                                        ;  TEST 17
                                        ;--------------
014406  012737  000017  001226  TST17:  MOV     #17,TSTNO
014414  012737  014670  001216          MOV     #TST20,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
014422  104412                          MSTCLR                  ;MASTER CLEAR DMC11
014424  005061  000004                  CLR     4(R1)           ;CLEAR PORT4
014430  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014432  122117                          122117                  ;PUT LINE UNIT IN BITSTUFF MODE
014434  004737  033374                  JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
014440  005037  033612                  CLR     BITCON          ;CONSECUTIVE 1'S COUNTER INIT TO 0
014444  012711  004000                  MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
014450  004737  032176                  JSR     PC,OUTRDY       ;WAIT FOR OUT-READY
014454  012761  000001  000004          MOV     #1,4(R1)        ;SET BIT0 IN PORT4
014462  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014464  122111                          122111                  ;SET SOM!
014466  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014470  122110                          122110                  ;LOAD GARBAGE CHAR
014472  012705  000000                  MOV     #0,R5   ;LOAD CHARACTER IN R5 FOR TYPEOUT
014476  004737  032176                  JSR     PC,OUTRDY       ;WAIT FOR OUT-READY
014502  010561  000004                  MOV     R5,4(R1)        ;LOAD PORT4 WITH CHARACTER
014506  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014510  122110                          122110                  ;LOAD OUT DATA
014512  004737  032044                  JSR     PC,OCOR         ;WAIT FOR OCOR TO SET
014516  005003                          CLR     R3              ;CLEAR BIT COUNTER
014520  010502                          MOV     R5,R2           ;LOAD CHARACTER IN R2
014522  104415  000002                  DATACLK,        2       ;2 TICKS TO SET UP TRANSMITTER
014526  012737  000176  001252          MOV     #↑B<01111110>,TEMP3 ;PUT FLAG CHARACTER IN TEMP3
```

```
014534  104415  000001                  64$:    DATACLK,        1       ;CLOCK FLAG ONCE
014540  106037  001252                          RORB    TEMP3           ;SHIFT SOFT FLAG
014544  103405                                  BCS     65$             ;BR IF BIT IS MARK
014546  004737  032012                          JSR     PC,GETSI        ;LOOK AT BIT WINDOW
014552  103006                                  BCC     66$             ;BR IF OK
014554  104026                                  HLT     26              ;ERROR IN FLAG CHAR
014556  000404                                  BR      66$
014560  004737  032012                  65$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
014564  103401                                  BCS     66$             ;BR IF OK
014566  104026                                  HLT     26              ;ERROR IN FLAG CHAR
014570  005203                          66$:    INC     R3              ;INC BIT COUNT
014572  022703  000010                          CMP     #10,R3          ;FLAG DONE YET?
014576  001356                                  BNE     64$             ;BR IF NO
014600  005003                                  CLR     R3              ;CLEAR BIT COUNT
014602  104415  000001                  1$:     DATACLK,        1       ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
014606  106002                                  RORB    R2              ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
014610  103005                                  BCC     2$              ;BR IF CARRY CLEAR
014612  004737  032012                          JSR     PC,GETSI        ;GET THE WINDOW
014616  103406                                  BCS     3$              ;BR IF BIT IS A MARK
014620  104006                                  HLT     6               ;ERROR BIT WAS A SPACE
014622  000404                                  BR      3$              ;CONTINE WITH TEST
014624  004737  032012                  2$:     JSR     PC,GETSI        ;GET THE WINDOW
014630  103001                                  BCC     3$              ;BR IF BIT IS A SPACE
014632  104006                                  HLT     6               ;ERROR BIT WAS A MARK
014634                                  3$:
014634  005203                                  INC     R3              ;NEXT BIT
014636  022703  000010                          CMP     #10,R3          ;DONE YET?
014642  001357                                  BNE     1$              ;BR IF NO
014644  104415  000014                          DATACLK,        14      ;CLOCK TRANSMITTER 14 MORE TICKS
014650  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014652  021264                                  021264                  ;PORT4+LU-13
014654  032761  000040  000004                  BIT     #BIT5,4(R1)     ;RTS SHOULD BE CLEAR NOW
014662  001401                                  BEQ     4$              ;BR IF YES
014664  104034                                  HLT     34              ;ERROR, RTS NOT CLEAR
014666  104400                          4$:     SCOPE                   ;SCOPE THIS TEST



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

                                                ;  TEST 20
                                                ;--------------
014670  012737  000020  001226          TST20:  MOV     #20,TSTNO
014676  012737  015152  001216                  MOV     #TST21,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
014704  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
014706  005061  000004                          CLR     4(R1)           ;CLEAR PORT4
014712  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014714  122117                                  122117                  ;PUT LINE UNIT IN BITSTUFF MODE
014716  004737  033374                          JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
```

```
014722  005037  033612                          CLR     BITCON                  ;CONSECUTIVE 1'S COUNTER INIT TO 0
014726  012711  004000                          MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
014732  004737  032176                          JSR     PC,OUTRDY               ;WAIT FOR OUT-READY
014736  012761  000001  000004                  MOV     #1,4(R1)                ;SET BIT0 IN PORT4
014744  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014746  122111                                  122111                          ;SET SOM!
014750  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014752  122110                                  122110                          ;LOAD GARBAGE CHAR
014754  012705  000125                          MOV     #125,R5 ;LOAD CHARACTER IN R5 FOR TYPEOUT
014760  004737  032176                          JSR     PC,OUTRDY               ;WAIT FOR OUT-READY
014764  010561  000004                          MOV     R5,4(R1)                ;LOAD PORT4 WITH CHARACTER
014770  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014772  122110                                  122110                          ;LOAD OUT DATA
014774  004737  032044                          JSR     PC,OCOR                 ;WAIT FOR OCOR TO SET
015000  005003                                  CLR     R3                      ;CLEAR BIT COUNTER
015002  010502                                  MOV     R5,R2                   ;LOAD CHARACTER IN R2
015004  104415  000002                          DATACLK,        2               ;2 TICKS TO SET UP TRANSMITTER
015010  012737  000176  001252                  MOV     #↑B<01111110>,TEMP3 ;PUT FLAG CHARACTER IN TEMP3
015016  104415  000001                  64$:    DATACLK,        1               ;CLOCK FLAG ONCE
015022  106037  001252                          RORB    TEMP3                   ;SHIFT SOFT FLAG
015026  103405                                  BCS     65$                     ;BR IF BIT IS MARK
015030  004737  032012                          JSR     PC,GETSI                ;LOOK AT BIT WINDOW
015034  103006                                  BCC     66$                     ;BR IF OK
015036  104026                                  HLT     26                      ;ERROR IN FLAG CHAR
015040  000404                                  BR      66$
015042  004737  032012                  65$:    JSR     PC,GETSI                ;LOOK AT BIT WINDOW
015046  103401                                  BCS     66$                     ;BR IF OK
015050  104026                                  HLT     26                      ;ERROR IN FLAG CHAR
015052  005203                          66$:    INC     R3                      ;INC BIT COUNT
015054  022703  000010                          CMP     #10,R3                  ;FLAG DONE YET?
015060  001356                                  BNE     64$                     ;BR IF NO
015062  005003                                  CLR     R3                      ;CLEAR BIT COUNT
015064  104415  000001                  1$:     DATACLK,        1               ;SHIFT NEXT BIT IN THE WINDOW (SI BIT
015070  106002                                  RORB    R2                      ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
015072  103005                                  BCC     2$                      ;BR IF CARRY CLEAR
015074  004737  032012                          JSR     PC,GETSI                ;GET THE WINDOW
015100  103406                                  BCS     3$                      ;BR IF BIT IS A MARK
015102  104006                                  HLT     6                       ;ERROR BIT WAS A SPACE
015104  000404                                  BR      3$                      ;CONTINE WITH TEST
015106  004737  032012                  2$:     JSR     PC,GETSI                ;GET THE WINDOW
015112  103001                                  BCC     3$                      ;BR IF BIT IS A SPACE
015114  104006                                  HLT     6                       ;ERROR BIT WAS A MARK
015116                                  3$:
015116  005203                                  INC     R3                      ;NEXT BIT
015120  022703  000010                          CMP     #10,R3                  ;DONE YET?
015124  001357                                  BNE     1$                      ;BR IF NO
015126  104415  000014                          DATACLK,        14              ;CLOCK TRANSMITTER 14 MORE TICKS
015132  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015134  021264                                  021264                          ;PORT4←LU-13
015136  032761  000040  000004                  BIT     #BIT5,4(R1)             ;RTS SHOULD BE CLEAR NOW
015144  001401                                  BEQ     4$                      ;BR IF YES
015146  104034                                  HLT     34                      ;ERROR, RTS NOT CLEAR
015150  104400                          4$:     SCOPE                           ;SCOPE THIS TEST



                                                ;****************************** TEST 21 ******************************
```

```
                                                     ;*BITSTUFF TRANSMITTER TEST
                                                     ;*SINGLE CLOCK THE CHARACTER 252
                                                     ;*CHECK FLAG AND DATA IN THE BIT WINDOW
                                                     ;*VERIFY EACH BIT POSITION AS IT
                                                     ;*PASSES THE BIT WINDOW (SI BIT)
                                                     ;*ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
                                                     ;:*************************************************************

                                                     ;  TEST 21
                                                     ;--------------
015152  012737  000021  001226          TST21:  MOV     #21,TSTNO
015160  012737  015434  001216                  MOV     #TST22,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
015166  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
015170  005061  000004                          CLR     4(R1)           ;CLEAR PORT4
015174  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015176  122117                                  122117                  ;PUT LINE UNIT IN BITSTUFF MODE
015200  004737  033374                          JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
015204  005037  033612                          CLR     BITCON          ;CONSECUTIVE 1'S COUNTER INIT TO 0
015210  012711  004000                          MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
015214  004737  032176                          JSR     PC,OUTRDY       ;WAIT FOR OUT-READY
015220  012761  000001  000004                  MOV     #1,4(R1)        ;SET BIT0 IN PORT4
015226  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015230  122111                                  122111                  ;SET SOM!
015232  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015234  122110                                  122110                  ;LOAD GARBAGE CHAR
015236  012705  000252                          MOV     #252,R5 ;LOAD CHARACTER IN R5 FOR TYPEOUT
015242  004737  032176                          JSR     PC,OUTRDY       ;WAIT FOR OUT-READY
015246  010561  000004                          MOV     R5,4(R1)        ;LOAD PORT4 WITH CHARACTER
015252  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015254  122110                                  122110                  ;LOAD OUT DATA
015256  004737  032044                          JSR     PC,OCOR         ;WAIT FOR OCOR TO SET
015262  005003                                  CLR     R3              ;CLEAR BIT COUNTER
015264  010502                                  MOV     R5,R2           ;LOAD CHARACTER IN R2
015266  104415  000002                          DATACLK,        2       ;2 TICKS TO SET UP TRANSMITTER
015272  012737  000176  001252                  MOV     #↑B<01111110>,TEMP3 ;PUT FLAG CHARACTER IN TEMP3
015300  104415  000001                  64$:    DATACLK,        1       ;CLOCK FLAG ONCE
015304  106037  001252                          RORB    TEMP3           ;SHIFT SOFT FLAG
015310  103405                                  BCS     65$             ;BR IF BIT IS MARK
015312  004737  032012                          JSR     PC,GETSI        ;LOOK AT BIT WINDOW
015316  103006                                  BCC     66$             ;BR IF OK
015320  104026                                  HLT     26              ;ERROR IN FLAG CHAR
015322  000404                                  BR      66$
015324  004737  032012                  65$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
015330  103401                                  BCS     66$             ;BR IF OK
015332  104026                                  HLT     26              ;ERROR IN FLAG CHAR
015334  005203                          66$:    INC     R3              ;INC BIT COUNT
015336  022703  000010                          CMP     #10,R3          ;FLAG DONE YET?
015342  001356                                  BNE     64$             ;BR IF NO
015344  005003                                  CLR     R3              ;CLEAR BIT COUNT
015346  104415  000001                  1$:     DATACLK,        1       ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
015352  106002                                  RORB    R2              ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
015354  103005                                  BCC     2$              ;BR IF CARRY CLEAR
015356  004737  032012                          JSR     PC,GETSI        ;GET THE WINDOW
015362  103406                                  BCS     3$              ;BR IF BIT IS A MARK
015364  104006                                  HLT     6               ;ERROR BIT WAS A SPACE
```

```
015366  000404                            BR        3$                ;CONTINE WITH TEST
015370  004737  032012            2$:     JSR       PC,GETSI          ;GET THE WINDOW
015374  103001                            BCC       3$                ;BR IF BIT IS A SPACE
015376  104006                            HLT       6                 ;ERROR BIT WAS A MARK
015400                            3$:
015400  005203                            INC       R3                ;NEXT BIT
015402  022703  000010                    CMP       #10,R3            ;DONE YET?
015406  001357                            BNE       1$                ;BR IF NO
015410  104415  000014                    DATACLK,          14        ;CLOCK TRANSMITTER 14 MORE TICKS
015414  104414                            ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015416  021264                            021264                      ;PORT4+LU-13
015420  032761  000040  000004            BIT       #BIT5,4(R1)       ;RTS SHOULD BE CLEAR NOW
015426  001401                            BEQ       4$                ;BR IF YES
015430  104034                            HLT       34                ;ERROR, RTS NOT CLEAR
015432  104400                    4$:     SCOPE                       ;SCOPE THIS TEST


                                          ;*************************** TEST 22 ****************************
                                          ;*BIT STUFF TEST
                                          ;*THIS TEST CHECKS ZERO BIT STUFFING OF
                                          ;* THE TRANSMITTER IN THE BIT WINDOW
                                          ;:****************************************************************

                                          :  TEST 22
                                          ;---------------
015434  012737  000022  001226    TST22:  MOV       #22,TSTNO
015442  012737  015744  001216            MOV       #TST23,NEXT
                                                                      ;R1 CONTAINS BASE DMC11 ADDRESS
015450  104412                            MSTCLR                      ;MASTER CLEAR DMC1.
015452  005061  000004                    CLR       4(R1)             ;CLEAR PORT4
015456  104414                            ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015460  122117                            122117                      ;PUT LINE UNIT IN BITSTUFF MODE
015462  004737  033374                    JSR       PC,CLRIO          ;DO THIS AFTER MODE IS SET
015466  012711  004000                    MOV       #BIT11,(R1)       ;SET LU LOOP
015472  004737  032176                    JSR       PC,OUTRDY         ;WAIT FOR OUT-READY
015476  012761  000001  000004            MOV       #1,4(R1)          ;SET BIT0 IN PORT4
015504  104414                            ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015506  122111                            122111                      ;SET SOM!
015510  104414                            ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015512  122110                            122110                      ;LOAD GARBAGE CHAR
015514  004537  033332                    JSR       R5,MESLD          ;LOAD OUT SILO DATA
015520  033640                            STUFDT              ;MESSAGE ADDRESSS
015522  000024                            20.                         ;NUMBER OF CHARACTERS
015524  012704  033640                    MOV       #STUFDT,R4        ;R4=CHARACTER POINTER
015530  005003                            CLR       R3                ;R3= BIT COUNTER
015532  012700  000006                    MOV       #6,R0             ;BIT COUNTER FOR FLAG CHARACTER
015536  104415  000002                    DATACLK,          2         ;SET UP TRANSMITTER
015542  012737  000176  001252            MOV       #↑B<01111110>,TEMP3 ;PUT FLAG CHARACTER IN TEMP3
015550  104415  000001            64$:    DATACLK,          1         ;CLOCK FLAG ONCE
015554  106037  001252                    RORB      TEMP3             ;SHIFT SOFT FLAG
015560  103405                            BCS       65$               ;BR IF BIT IS MARK
015562  004737  032012                    JSR       PC,GETSI          ;LOOK AT BIT WINDOW
015566  103006                            BCC       66$               ;BR IF OK
015570  104026                            HLT       26                ;ERROR IN FLAG CHAR
015572  000404                            BR        66$
015574  004737  032012            65$:    JSR       PC,GETSI          ;LOOK AT BIT WINDOW
```

```
015600  103401                                  BCS     66$             ;BR IF OK
015602  104026                                  HLT     26              ;ERROR IN FLAG CHAR
015604  005203                          66$:    INC     R3              ;INC BIT COUNT
015606  022703  000010                          CMP     #10,R3          ;FLAG DONE YET?
015612  001356                                  BNE     64$             ;BR IF NO
015614  005003                                  CLR     R3              ;CLEAR BIT COUNT
015616  012700  000024                          MOV     #20.,R0         ;R0=CHARACTER COUNTER
015622  005037  033612                          CLR     BITCON          ;CLEAR BIT STUFF COUNTER
015626  112405                          3$:     MOVB    (R4)+,R5        ;LOAD CHARACTER IN R5
015630  010502                                  MOV     R5,R2           ;LOAD CHARACTER IN R2
015632  104415  000001                  4$:     DATACLK,        1       ;SHIFT DTAT ONCE
015636  106002                                  RORB    R2              ;SHIFT SOFT DATA
015640  103407                                  BCS     5$              ;BR IF CARRY SET
015642  005037  033612                          CLR     BITCON          ;CLEAR BIT STUFF COUNTER
015646  004737  032012                          JSR     PC,GETSI        ;LOOK AT WINDOW
015652  103010                                  BCC     6$              ;BR IF SPACE
015654  104006                                  HLT     6               ;ERROR, WINDOW WAS A MARK
015656  000406                                  BR      6$              ;CONTINUE
015660  005237  033612                  5$:     INC     BITCON          ;ADD 1 TO BIT STUFF COUNTER
015664  004737  032012                          JSR     PC,GETSI        ;LOOK AT WINDOW
015670  103401                                  BCS     6$              ;BR IF MARK
015672  104006                                  HLT     6               ;ERROR, WINDOW WAS A SPACE
015674  022737  000005  033612          6$:     CMP     #5,BITCON       ;HAVE THERE BEEN 5 1'S IN A ROW
015702  001010                                  BNE     7$              ;BR IF NO
015704  005037  033612                          CLR     BITCON          ;IF YES CLR BIT STUFF COUNTER
015710  104415  000001                          DATACLK,        1       ;AND CLOCK TRANSMITTER ONCE
015714  004737  032012                          JSR     PC,GETSI        ;CHECK WINDOW FOR A ZEOR STUFF!!
015720  103001                                  BCC     7$              ;BR IF WINDOW IS A SPACE
015722  104030                                  HLT     30              ;ERROR, TRANSMITTER DID NOT STUFF A ZERO
015724  005203                          7$:     INC     R3              ;BUMP BIT COUNTER
015726  022703  000010                          CMP     #10,R3          ;DONE THIS CHARACTER YET?
015732  001337                                  BNE     4$              ;BR IF NO
015734  005003                                  CLR     R3              ;RESTART BIT COUNTER AT ZERO
015736  005300                                  DEC     R0              ;DEC CHARACTER COUNTER
015740  001332                                  BNE     3$              ;BR IF NOT DONE YET
015742  104400                          9$:     SCOPE                   ;SCOPE THIS TEST



                                        ;*************************** TEST 23 ***************************
                                        ;*BITSTUFF TRANSMITTER TEST
                                        ;*SINGLE CLOCK THE CHARACTER 377
                                        ;*CHECK FLAG AND DATA IN THE BIT WINDOW
                                        ;*VERIFY EACH BIT POSITION AS IT
                                        ;*PASSES THE BIT WINDOW (SI BIT)
                                        ;*ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
                                        ;;**************************************************************

                                        ;  TEST 23
                                        ;---------------

015744  012737  000023  001226          TST23:  MOV     #23,TSTNO
015752  012737  016252  001216                  MOV     #TST24,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
015760  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
015762  005061  000004                          CLR     4(R1)           ;CLEAR PORT4
015766  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015770  122117                                  122117                  ;PUT LINE UNIT IN BITSTUFF MODE
```

```
015772  004737  033374                  JSR     PC,CLRIO          ;DO THIS AFTER MODE IS SET
015776  005037  033612                  CLR     BITCON            ;CONSECUTIVE 1'S COUNTER INIT TO 0
016002  012711  004000                  MOV     #BIT11,(R1)       ;SET LINE UNIT LOOP
016006  004737  032176                  JSR     PC,OUTRDY         ;WAIT FOR OUT-READY
016012  012761  000001  000004          MOV     #1,4(R1)          ;SET BIT0 IN PORT4
016020  104414                          ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016022  122111                          122111                    ;SET SOM!
016024  104414                          ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016026  122110                          122110                    ;LOAD GARBAGE CHAR
016030  012705  000377                  MOV     #377,R5 ;LOAD CHARACTER IN R5 FOR TYPEOUT
016034  010537  016206                  MOV     R5,5$             ;LOAD CHAR FOR STUFF CHECK
016040  004737  032176                  JSR     PC,OUTRDY         ;WAIT FOR OUT-READY
016044  010561  000004                  MOV     R5,4(R1)          ;LOAD PORT4 WITH CHARACTER
016050  104414                          ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016052  122110                          122110                    ;LOAD OUT DATA
016054  004737  032044                  JSR     PC,OCOR           ;WAIT FOR OCOR TO SET
016060  005003                          CLR     R3                ;CLEAR BIT COUNTER
016062  010502                          MOV     R5,R2             ;LOAD CHARACTER IN R2
016064  104415  000002                  DATACLK,        2         ;2 TICKS TO SET UP TRANSMITTER
016070  012737  000176  001252          MOV     #↑B<01111110>,TEMP3 ;PUT FLAG CHARACTER IN TEMP3
016076  104415  000001          64$:    DATACLK,        1         ;CLOCK FLAG ONCE
016102  106037  001252                  RORB    TEMP3             ;SHIFT SOFT FLAG
016106  103405                          BCS     65$               ;BR IF BIT IS MARK
016110  004737  032012                  JSR     PC,GETSI          ;LOOK AT BIT WINDOW
016114  103006                          BCC     66$               ;BR IF OK
016116  104026                          HLT     26                ;ERROR IN FLAG CHAR
016120  000404                          BR      66$
016122  004737  032012          65$:    JSR     PC,GETSI          ;LOOK AT BIT WINDOW
016126  103401                          BCS     66$               ;BR IF OK
016130  104026                          HLT     26                ;ERROR IN FLAG CHAR
016132  005203                  66$:    INC     R3                ;INC BIT COUNT
016134  022703  000010                  CMP     #10,R3            ;FLAG DONE YET?
016140  001356                          BNE     64$               ;BR IF NO
016142  005003                          CLR     R3                ;CLEAR BIT COUNT
016144  005037  033612                  CLR     BITCON            ;CLEAR STUFF COUNT
016150  104415  000001          1$:     DATACLK,        1         ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
016154  106002                          RORB    R2                ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
016156  103005                          BCC     2$                ;BR IF CARRY CLEAR
016160  004737  032012                  JSR     PC,GETSI          ;GET THE WINDOW
016164  103406                          BCS     3$                ;BR IF BIT IS A MARK
016166  104006                          HLT     6                 ;ERROR BIT WAS A SPACE
016170  000404                          BR      3$                ;CONTINE WITH TEST
016172  004737  032012          2$:     JSR     PC,GETSI          ;GET THE WINDOW
016176  103001                          BCC     3$                ;BR IF BIT IS A SPACE
016200  104006                          HLT     6                 ;ERROR BIT WAS A MARK
016202                          3$:
016202  004537  033474                  JSR     R5,STFFCK         ;CHECK FOR BIT STUFF
016206  000377                  5$:     377                       ;DATA CHARACTER
016210  000001                          1                         ;SHIFT COUNT
016212  010237  016206                  MOV     R2,5$             ;LOAD CHAR FOR STUFF CHECK
016216  005203                          INC     R3                ;NEXT BIT
016220  022703  000010                  CMP     #10,R3            ;DONE YET?
016224  001351                          BNE     1$                ;BR IF NO
016226  104415  000014                  DATACLK,        14        ;CLOCK TRANSMITTER 14 MORE TICKS
016232  104414                          ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016234  021264                          021264                    ;PORT4+LU-13
```

```
016236  032761  000040  000004                  BIT     #BIT5,4(R1)         ;RTS SHOULD BE CLEAR NOW
016244  001401                                  BEQ     4$                  ;BR IF YES
016246  104034                                  HLT     34                  ;ERROR, RTS NOT CLEAR
016250  104400                          4$:     SCOPE                       ;SCOPE THIS TEST


                                                ;************************* TEST 24 ***************************
                                                ;*BITSTUFF 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 24
                                                ;--------------
016252  012737  000024  001226          TST24:  MOV     #24,TSTNO
016260  012737  016604  001216                  MOV     #TST25,NEXT
                                                                            ;R1 CONTAINS BASE DMC11 ADDRESS
016266  104412                                  MSTCLR                      ;MASTER CLEAR DMC11
016270  005061  000004                          CLR     4(R1)               ;CLEAR PORT4
016274  104414                                  ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016276  122117                                  122117                      ;PUT LINE UNIT IN BITSTUFF MODE
016300  004737  033374                          JSR     PC,CLRIO            ;DO THIS AFTER MODE IS SET
016304  005037  033612                          CLR     BITCON              ;CONSECUTIVE 1'S COUNTER INIT TO 0
016310  012711  004000                          MOV     #BIT11,(R1)         ;SET LINE UNIT LOOP
016314  005003                                  CLR     R3                  ;R3 CONTAINS BIT COUNT
016316  005004                                  CLR     R4                  ;R4 CONTAINS CHAR TO BE LOADED IN SILO
016320  005005                                  CLR     R5                  ;R5 CONTAINS CHARACTER CURRENTLY BEING SHIFTED C
016322  004737  032176                          JSR     PC,OUTRDY           ;WAIT FOR OUT-READY
016326  012761  000001  000004                  MOV     #1,4(R1)            ;SET BIT0 IN PORT4
016334  104414                                  ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016336  122111                                  122111                      ;SET SOM!
016340  104414                                  ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016342  122110                                  122110                      ;LOAD GARBAGE CHAR
016344  004737  032176                          JSR     PC,OUTRDY           ;WAIT FOR OUT-READY
016350  010461  000004                          MOV     R4,4(R1)            ;LOAD PORT4 WITH CHARACTER
016354  104414                                  ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016356  122110                                  122110                      ;LOAD OUT DATA
016360  005204                                  INC     R4                  ;INCREMENT TO NEXT CHARACTER
016362  004737  032176                          JSR     PC,OUTRDY           ;WAIT FOR OUT-READY
016366  010461  000004                          MOV     R4,4(R1)            ;LOAD PORT4 WITH CHARACTER
016372  104414                                  ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016374  122110                                  122110                      ;LOAD OUT DATA
016376  004737  032044                          JSR     PC,OCOR             ;WAIT FOR OCOR TO SET
016402  104415  000002                          DATACLK,        2           ;2 TICKS TO SET UP TRANSMITTER
016406  012737  000176  001252                  MOV     #↑B<01111110>,TEMP3 ;PUT FLAG CHARACTER IN TEMP3
016414  104415  000001                  64$:    DATACLK,        1           ;CLOCK FLAG ONCE
016420  106037  001252                          RORB    TEMP3               ;SHIFT SOFT FLAG
016424  103405                                  BCS     65$                 ;BR IF BIT IS MARK
016426  004737  032012                          JSR     PC,GETSI            ;LOOK AT BIT WINDOW
016432  103006                                  BCC     66$                 ;BR IF OK
016434  104026                                  HLT     26                  ;ERROR IN FLAG CHAR
016436  000404                                  BR      66$
016440  004737  032012                  65$:    JSR     PC,GETSI            ;LOOK AT BIT WINDOW
```

```
016444  103401                             BCS     66$             ;BR IF OK
016446  104026                             HLT     26              ;ERROR IN FLAG CHAR
016450  005203                     66$:    INC     R3              ;INC BIT COUNT
016452  022703  000010                     CMP     #10,R3          ;FLAG DONE YET?
016456  001356                             BNE     64$             ;BR IF NO
016460  005003                             CLR     R3              ;CLEAR BIT COUNT
016462  005037  033612                     CLR     BITCON          ;CLEAR BIT STUFF COUNTER
016466  005003                     4$:     CLR     R3              ;CLEAR BIT COUNTER
016470  010502                             MOV     R5,R2           ;LOAD CHARACTER IN R2
016472  010237  016534                     MOV     R2,6$                   ;LOAD CHAR FOR STUFF CHECK
016476  104415  000001             1$:     DATACLK,          1     ;SHIFT NEXT BIT IN THE WINDOW  SI BIT)
016502  106002                             RORB    R2              ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
016504  103005                             BCC     2$              ;BR IF CARRY CLEAR
016506  004737  032012                     JSR     PC,GETSI        ;GET THE WINDOW
016512  103406                             BCS     3$              ;BR IF BIT IS A MARK
016514  104006                             HLT     6               ;ERROR BIT WAS A SPACE
016516  000404                             BR      3$              ;CONTINE WITH TEST
016520  004737  032012             2$:     JSR     PC,GETSI        ;GET THE WINDOW
016524  103001                             BCC     3$              ;BR IF BIT IS A SPACE
016526  104006                             HLT     6               ;ERROR BIT WAS A MARK
016530                             3$:
016530  004537  033474                     JSR     R5,STFFCK       ;CHECK FOR BIT STUFF
016534  000000                     6$:     0                       ;DATA CHARACTER
016536  000001                             1                       ;SHIFT COUNT
016540  010237  016534                     MOV     R2,6$           ;LOAD CHAR FOR STUFF CHECK
016544  005203                             INC     R3              ;NEXT BIT
016546  022703  000010                     CMP     #10,R3          ;DONE YET?
016552  001351                             BNE     1$              ;BR IF NO
016554  005204                             INC     R4              ;NEXT CHARACTER
016556  004737  032176                     JSR     PC,OUTRDY       ;WAIT FOR OUT-READY
016562  010461  000004                     MOV     R4,4(R1)        ;LOAD PORT4 WITH CHARACTER
016566  104414                             ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016570  122110                             122110                  ;LOAD OUT DATA
016572  005205                             INC     R5              ;NEXT CHARACTER
016574  022705  000400                     CMP     #400,R5         ;DONE YET?
016600  001332                             BNE     4$              ;BR IF NO
016602  104400                     5$:     SCOPE                   ;SCOPE THIS TEST



                                           ;*************************** TEST 25 **************************
                                           ;*MULTIPLE FLAG AND TRANSMITTER ABORT TEST
                                           ;*LOAD SILO WITH 5 FLAGS AND A CHAR (000)
                                           ;*VERIFIY IN THE BIT WINDOW THAT THE FLAGS
                                           ;*AND DATA ARE CORRECT AND FOLLOWED BY AN ABORT
                                           ;*SEQUENCE (8 CONTIGUOUS 1'S)
                                           ;;*************************************************************


                                           ;  TEST 25
                                           ;---------------
016604  012737  000025  001226     TST25:  MOV      #25,TSTNO
016612  012737  017072  001216             MOV      #TST26,NEXT
                                                                   ;R1 CONTAINS BASE DMC11 ADDRESS
016620  104412                             MSTCLR                  ;MASTER CLEAR DMC11
016622  005061  000004                     CLR      4(R1)          ;CLEAR PORT4
016626  104414                             ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016630  122117                             122117                  ;PUT LINE UNIT IN BITSTUFF MODE
```

```
016632  004737  033374                           JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
016636  012711  004000                           MOV     #BIT11,(R1)     ;SET LU LOOP
016642  012700  000005                           MOV     #5,R0           ;FLAG COUNT
016646  005003                                   CLR     R3              ;CLEAR BIT COUNTER
016650  004737  032176                   1$:     JSR     PC,OUTRDY       ;WAIT FOR OUT-READY
016654  012761  000001  000004                   MOV     #1,4(R1)        ;SET BIT0 IN PORT4
016662  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016664  122111                                   122111                  ;SET SOM!
016666  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016670  122110                                   122110                  ;LOAD GARBAGE CHAR
016672  005300                                   DEC     R0              ;DEC COUNT
016674  001365                                   BNE     1$              ;LOAD ANOTHER
016676  004737  032176                           JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
016702  005061  000004                           CLR     4(R1)           ;CLEAR PORT4
016706  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016710  122110                                   122110                  ;LOAD A ZERO
016712  004737  032044                           JSR     PC,OCOP         ;WAIT
016716  012700  000005                           MOV     #5,R0           ;R0 = FLAG COUNT
016722  104415  000002                           DATACLK,        2       ;SET UP TRANSMITTER
016726                                   2$:
016726  012737  000176  001252                   MOV     #^B<01111110>,TEMP3 ;PUT FLAG CHARACTER IN TEMP3
016734  104415  000001                   64$:    DATACLK,        1       ;CLOCK FLAG ONCE
016740  106037  001252                           RORB    TEMP3           ;SHIFT SOFT FLAG
016744  103405                                   BCS     65$             ;BR IF BIT IS MARK
016746  004737  032012                           JSR     PC,GETSI        ;LOOK AT BIT WINDOW
016752  103006                                   BCC     66$             ;BR IF OK
016754  104026                                   HLT     26              ;ERROR IN FLAG CHAR
016756  000404                                   BR      66$
016760  004737  032012                   65$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
016764  103401                                   BCS     66$             ;BR IF OK
016766  104026                                   HLT     26              ;ERROR IN FLAG CHAR
016770  005203                           66$:    INC     R3              ;INC BIT COUNT
016772  022703  000010                           CMP     #10,R3          ;FLAG DONE YET?
016776  001356                                   BNE     64$             ;BR IF NO
017000  005003                                   CLR     R3              ;CLEAR BIT COUNT
017002  005300                                   DEC     R0              ;DEC COUNT
017004  001350                                   BNE     2$              ;BR IF NOT DONE
017006  005003                                   CLR     R3              ;R3 = BIT COUNT
017010  005005                                   CLR     R5              ;R5 = "EXPECTED"
017012  104415  000001                   3$:     DATACLK,        1       ;CLOCK ONCE
017016  004737  032012                           JSR     PC,GETSI        ;GO LOOK AT WINDOW
017022  103001                                   BCC     4$              ;BR IF A SPACE
017024  104006                                   HLT     6               ;ERROR, A MARK WAS SEEN
017026  005203                           4$:     INC     R3              ;INC BIT COUNT
017030  022703  000010                           CMP     #10,R3          ;DONE YET?
017034  001366                                   BNE     3$              ;BR IF NO
017036  005003                                   CLR     R3              ;CLEAR BIT COUNT
017040  012705  000377                           MOV     #377,R5         ;R5 = "EXPECTED"
017044  104415  000001                   5$:     DATACLK,        1       ;CLOCK ONCE
017050  004737  032012                           JSR     PC,GETSI        ;LOOK AT WINDOW
017054  103401                                   BCS     6$              ;BR IF A MARY
017056  104033                                   HLT     33              ;ERROR, A SPACE WAS SEEN
017060  005203                           6$:     INC     R3              ;INC BIT COUNT
017062  022703  000010                           CMP     #10,R3          ;DONE YET?
017066  001366                                   BNE     5$              ;BR IF NO
017070  104400                                   SCOPE                   ;SCOPE THIS TEST
```

```


                                                ;*************************** TEST 26 ***************************
                                                ;*LEADING ZEROS TEST
                                                ;*VERIFY THAT THE SETTING OF SOM AND EOM TOGETHER
                                                ;*AND THEN SOM ALONE WILL GENERATE 16 LEADING ZEROS
                                                ;*AND A FLAG,THE CHECK IS MADE USING THE BIT WINDOW
                                                ;:****************************************************************

                                                ;  TEST 26
                                                ;---------------
017072  012737  000026  001226  TST26:  MOV       #26,TSTNO
017100  012737  017312  001216          MOV       #TST27,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
017106  104412                          MSTCLR                          ;MASTER CLEAR DMC11
017110  005061  000004                  CLR       4(R1)                 ;CLEAR PORT4
017114  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017116  122117                          122117                          ;SET TO BITSTUFF MODE
017120  004737  033374                  JSR       PC,CLRIO              ;DO THIS AFTER MODE IS SET
017124  012711  004000                  MOV       #BIT11,(R1)           ;SET LU LOOP
017130  004737  032176                  JSR       PC,OUTRDY             ;WAIT FOR OUTRDY
017134  012761  000003  000004          MOV       #3,4(R1)              ;LOAD PORT4
017142  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017144  122111                          122111                          ;SET SOM & EOM
017146  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017150  122110                          122110                          ;GARBAGE CHARACTER
017152  012761  000001  000004          MOV       #1,4(R1)              ;LOAD PORT4
017160  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017162  122111                          122111                          ;SET SOM
017164  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017166  122110                          122110                          ;GARBAGE CHAR
017170  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017172  122110                          122110                          ;GARBAGE CHAR
017174  004737  032044                  JSR       PC,OCOR               ;WAIT FOR OCOR
017200  005000                          CLR       R0                    ;R0 = BIT COUNT
017202  104415  000002                  DATACLK,2                       ;SET UP TRANSMITTER
017206  104415  000001          1$:     DATACLK,1                       ;SINGLE CLOCK TRANSMITTER
017212  004737  032012                  JSR       PC,GETSI              ;LOOK AT BITWINDOW
017216  103001                          BCC       .+4
017220  104041                          HLT       41                    ;ERROR WINDOW WAS A MARK
017222  005200                          INC       R0
017224  022700  000020                  CMP       #16.,R0               ;16 ZEROS YET?
017230  001366                          BNE       1$                    ;BR IF NO
017232  005003                          CLR       R3                    ;R3 = BIT COUNT
017234  012737  000176  001252          MOV       #↑B<01111110>,TEMP3 ;PUT FLAG CHARACTER IN TEMP3
017242  104415  000001          64$:    DATACLK,       1                ;CLOCK FLAG ONCE
017246  106037  001252                  RORB      TEMP3                 ;SHIFT SOFT FLAG
017252  103405                          BCS       65$                   ;BR IF BIT IS MARK
017254  004737  032012                  JSR       PC,GETSI              ;LOOK AT BIT WINDOW
017260  103006                          BCC       66$                   ;BR IF OK
017262  104026                          HLT       26                    ;ERROR IN FLAG CHAR
017264  000404                          BR        66$
017266  004737  032012          65$:    JSR       PC,GETSI              ;LOOK AT BIT WINDOW
017272  103401                          BCS       66$                   ;BR IF OK
017274  104026                          HLT       26                    ;ERROR IN FLAG CHAR
017276  005203                  66$:    INC       R3                    ;INC BIT COUNT
```

```
017300  022703  000010                    CMP       #10,R3          ;FLAG DONE YET?
017304  001356                            BNE       64$             ;BR IF NO
017306  005003                            CLR       R3              ;CLEAR BIT COUNT
017310  104400                            SCOPE                     ;SCOPE THIS TEST


                                          ;*************************** TEST 27 ***************************
                                          ;*BITSTUFF STRIP FLAG TEST
                                          ;*SET LU LOOP, SINGLE STEP 5 FLAGS,
                                          ;*VERIFY THAT IN ACTIVE DOES NOT SET
                                          ;;*************************************************************

                                          ;  TEST 27
                                          ;---------------
017312  012737  000027  001226   TST27:   MOV       #27,TSTNO
017320  012737  017414  001216            MOV       #TST30,NEXT
                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
017326  104412                            MSTCLR                    ;MASTER CLEAR DMC11
017330  005061  000004                    CLR       4(R1)           ;CLEAR PORT4
017334  104414                            ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017336  122117                            122117                    ;PUT LINE UNIT IN BITSTUFF MODE
017340  004737  033374                    JSR       PC,CLRIO        ;DO THIS AFTER MODE IS SET
017344  012711  004000                    MOV       #BIT11,(R1)     ;SET LU LOOP
017350  012702  000012                    MOV       #12,R2          ;SAVE LU REG FOR TYPEOUT
017354  004737  032062                    JSR       PC,SYNC         ;SINGLE CLOCK 5 SYNC CHARACTERS
017360  000005                            5
017362  104415  000054                    DATACLK,         54
017366  104414                            ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017370  021244                            021244                    ;PORT4←LU12
017372  016104  000004                    MOV       4(R1),R4        ;PUT "FOUND" IN R4
017376  042704  000277                    BIC       #277,R4         ;CLEAR UNWANTED BITS
017402  005005                            CLR       R5              ;PUT "EXPECTED" IN R5
017404  120504                            CMPB      R5,R4           ;IS ACTIVE CLEAR?
017406  001401                            BEQ       1$              ;BR IF YES
017410  104040                            HLT       40              ;ERROR ACTIVE IS NOT CLEAR
017412  104400                   1$:      SCOPE                     ;SCOPE THIS TEST


                                          ;*************************** TEST 30 ***************************
                                          ;*BITSTUFF IN ACTIVE TEST
                                          ;*SET LU LOOP, SINGLE STEP 5 FLAGS AND A NON-FLAG (301)
                                          ;*VERIFY THAT IN ACTIVE IS SET
                                          ;;*************************************************************

                                          ;  TEST 30
                                          ;---------------
017414  012737  000030  001226   TST30:   MOV       #30,TSTNO
017422  012737  017520  001216            MOV       #TST31,NEXT
                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
017430  104412                            MSTCLR                    ;MASTER CLEAR DMC11
017432  005061  000004                    CLR       4(R1)           ;CLEAR PORT4
017436  104414                            ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017440  122117                            122117                    ;PUT LINE UNIT IN BITSTUFF MODE
017442  004737  033374                    JSR       PC,CLRIO        ;DO THIS AFTER MODE IS SET
017446  012711  004000                    MOV       #BIT11,(R1)     ;SET LU LOOP
017452  012702  000012                    MOV       #12,R2          ;SAVE LU REG FOR TYPEOUT
```

```
017456  004737  032062                      JSR     PC,SYNC         ;SINGLE CLOCK 5 SYNC CHARACTERS
017462  000005                              5
017464  104415  000064                      DATACLK,        64
017470  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017472  021244                              021244                  ;PORT4+LU12
017474  016104  000004                      MOV     4(R1),R4        ;PUT "FOUND" IN R4
017500  042704  000277                      BIC     #277,R4         ;CLEAR UNWANTED BITS
017504  012705  000100                      MOV     #BIT6,R5              ;PUT "EXPECTED" IN R5
017510  120504                              CMPB    R5,R4           ;IS ACTIVE SET?
017512  001401                              BEQ     1$              ;BR IF YES
017514  104040                              HLT     40              ;ERROR ACTIVE IS NOT SET
017516  104400                      1$:     SCOPE                   ;SCOPE THIS TEST


                                            ;************************** TEST 31 **************************
                                            ;*BITSTUFF IN ACTIVE TEST
                                            ;*SET LINE UNIT LOOP,SINGLE STEP ONE FLAG AND A CHAR (301)
                                            ;*VERIFY THAT IN ACTIVE IS SET
                                            ;;*************************************************************

                                            ;  TEST 31
                                            ;--------------
017520  012737  000031  001226      TST31:  MOV     #31,TSTNO
017526  012737  017656  001216              MOV     #TST32,NEXT
                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
017534  104412                              MSTCLR                  ;MASTER CLEAR DMC11
017536  005061  000004                      CLR     4(R1)           ;CLEAR PORT4
017542  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017544  122117                              122117                  ;PUT LINE UNIT IN BITSTUFF MODE
017546  004737  033374                      JSR     PC,CLRIO        ;MUST DO THIS AFTER MODE IS SET
017552  012711  004000                      MOV     #BIT11,(R1)
017556  012702  000012                      MOV     #12,R2          ;SAVE REG ADDRESS FOR TYPEOUT
017562  004737  032176                      JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
017566  012761  000001  000004              MOV     #1,4(R1)        ;LOAD PORT4
017574  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017576  122111                              122111                  ;SET SOM
017600  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017602  122110                              122110                  ;LOAD GARBAGE CHAR
017604  012761  000301  000004              MOV     #301,4(R1)      ;LOAD PORT4
017612  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017614  122110                              122110                  ;LOAD OUT DATA
017616  004737  032044                      JSR     PC,OCOR         ;WAIT FOR OCOR
017622  104415  000023                      DATACLK,        23      ;SINGLE CLOCK THE DATA
017626  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017630  021244                              021244                  ;PORT4+LU-12
017632  016104  000004                      MOV     4(R1),R4        ;PUT "FOUND" IN R4
017636  042704  000277                      BIC     #277,R4         ;CLEAR UNWANTED BITS
017642  012705  000100                      MOV     #BIT6,R5              ;PUT "EXPECTED" IN R5
017646  120504                              CMPB    R5,R4           ;IS IN ACTIVE SET?
017650  001401                              BEQ     1$
017652  104040                              HLT     40              ;ERROR, IN ACTIVE NOT SET
017654  104400                      1$:     SCOPE                   ;SCOPE THIS TEST


                                            ;************************** TEST 32 **************************
                                            ;*BITSTUFF IN ACTIVE TEST
```

```
                                                        ;*SET LU LOOP, SINGLE STEP 2 FLAGS AND A NON-FLAG (301)
                                                        ;*VERIFY THAT IN ACTIVE IS SET
                                                        ;;*************************************************************


                                                        ;  TEST 32
                                                        ;--------------
017656  012737  000032  001226                  TST32:  MOV     #32,TSTNO
017664  012737  017762  001216                          MOV     #TST33,NEXT
                                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
017672  104412                                          MSTCLR                  ;MASTER CLEAR DMC11
017674  005061  000004                                  CLR     4(R1)           ;CLEAR PORT4
017700  104414                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017702  122117                                          122117                  ;PUT LINE UNIT IN BITSTUFF MODE
017704  004737  033374                                  JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
017710  012711  004000                                  MOV     #BIT11,(R1)     ;SET LU LOOP
017714  012702  000012                                  MOV     #12,R2          ;SAVE LU REG FOR TYPEOUT
017720  004737  032062                                  JSR     PC,SYNC         ;SINGLE CLOCK 2 SYNC CHARACTERS
017724  000002                                          2
017726  104415  000033                                  DATACLK,        33
017732  104414                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017734  021244                                          021244                  ;PORT4+LU12
017736  016104  000004                                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
017742  042704  000277                                  BIC     #277,R4         ;CLEAR UNWANTED BITS
017746  012705  000100                                  MOV     #BIT6,R5                ;PUT "EXPECTED" IN R5
017752  120504                                          CMPB    R5,R4           ;IS ACTIVE SET?
017754  001401                                          BEQ     1$              ;BR IF YES
017756  104040                                          HLT     40              ;ERROR ACTIVE IS NOT SET
017760  104400                                  1$:     SCOPE                   ;SCOPE THIS TEST


                                                        ;*************************** TEST 33 ***************************
                                                        ;*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 33
                                                        ;--------------
017762  012737  000033  001226                  TST33:  MOV     #33,TSTNO
017770  012737  020166  001216                          MOV     #TST34,NEXT
                                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
017776  104412                                          MSTCLR                  ;MASTER CLEAR DMC11
020000  005061  000004                                  CLR     4(R1)           ;CLEAR PORT4
020004  104414                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020006  122117                                          122117                  ;PUT LINE UNIT IN BITSTUFF MODE
020010  004737  033374                                  JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
020014  012702  000012                                  MOV     #12,R2          ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
020020  012711  004000                                  MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
020024  012761  000001  000004                          MOV     #1,4(R1)        ;SET BIT0 IN PORT4
020032  104414                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020034  122111                                          122111                  ;SET SOM!
020036  104414                                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020040  122110                                          122110                  ;LOAD GARBAGE CHAR
020042  004737  032342                                  JSR     PC,CHARSC       ;LOAD SILO WITH CHARACTER
020046  000026                                          26                      ;CHARACTER
```

```
020050  104415  000033                   DATACLK,        33      ;SINGLE CLOCK THE DATA
020054  104416  000002                   TIMER,   2              ;WAIT FOR INRDY
020060  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020062  021244                           021244                  ;PORT4←LU 12
020064  016104  000004                   MOV      4(R1),R4       ;PUT "FOUND" IN R4
020070  042704  000357                   BIC      #357,R4        ;CLEAR UNWANTED BITS
020074  012705  000020                   MOV      #BIT4,R5               ;PUT "EXPECTED" IN R5
020100  120504                           CMPB     R5,R4          ;IS INRDY SET?
020102  001401                           BEQ      1$
020104  104040                           HLT      40             ;ERROR, INRDY IS NOT SET
020106                           1$:
020106  012761  000200  000004           MOV      #BIT7 4(R1)    ;LOAD PORT4
020114  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020116  122112                           122112                  ;SET IN CLEAR
020120  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020122  021244                           021244                  ;PORT4←LU 12
020124  016104  000004                   MOV      4(R1),R4       ;PUT "FOUND" IN R4
020130  042704  000277                   BIC      #277,R4        ;CLEAR UNWANTED BITS
020134  005005                           CLR      R5             ;PUT "EXPECTED" IN R5
020136  120504                           CMPB     R5,R4          ;IS IN ACTIVE CLEAR?
020140  001401                           BEQ      2$
020142  104040                           HLT      40             ;ERROR, IN ACTIVE IS NOT CLEAR
020144                           2$:
020144  016104  000004                   MOV      4(R1),R4       ;PUT "FOUND" IN R4
020150  042704  000357                   BIC      #357,R4        ;CLEAR UNWANTED BITS
020154  005005                           CLR      R5             ;PUT "EXPECTED" IN R5
020156  120504                           CMPB     R5,R4          ;IS INRDY CLEARED?
020160  001401                           BEQ      3$
020162  104040                           HLT      40             ;ERROR, INRDY IS NOT CLEARED
020164  104400                   3$:     SCOPE                   ;SCOPE THIS TEST



                                         ;*************************** TEST 34 ***************************
                                         ;*BITSTUFF 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 34
                                         ;---------------
020166  012737  000034  001226   TST34:  MOV      #34,TSTNO
020174  012737  020334  001216           MOV      #TST35,NEXT
                                                                 ;R1 CONTAINS BASE DMC11 ADDRESS
020202  104412                           MSTCLR                  ;MASTER CLEAR DMC11
020204  005061  000004                   CLR      4(R1)          ;CLEAR PORT4
020210  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020212  122117                           122117                  ;PUT LINE UNIT IN BITSTUFF MODE
020214  004737  033374                   JSR      PC,CLRIO       ;DO THIS AFTER MODE IS SET
020220  012702  000012                   MOV      #12,R2         ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
020224  012711  004000                   MOV      #BIT11,(R1)    ;SET LINE UNIT LOOP
020230  012761  000001  000004           MOV      #1,4(R1)       ;SET BIT0 IN PORT4
020236  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020240  122111                           122111                  ;SET SOM!
020242  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020244  122110                           122110                  ;LOAD GARBAGE CHAR
020246  004737  032342                   JSR      PC,CHARSD      ;LOAD SILO WITH CHARACTER
```

```
020252  000000                                  0                       ;CHARACTER
020254  104415  000033                          DATACLK,        33      ;SINGLE CLOCK THE DATA
020260  104416  000002                          TIMER,  2               ;WAIT FOR INRDY
020264  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020266  021244                                  021244                  ;PORT4+LU 12
020270  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
020274  042704  000357                          BIC     #357,R4         ;CLEAR UNWANTED BITS
020300  012705  000020                          MOV     #BIT4,R5                ;PUT "EXPECTED" IN R5
020304  120504                                  CMPB    R5,R4           ;IS INRDY SET?
020306  001401                                  BEQ     1$
020310  104040                                  HLT     40              ;ERROR, INRDY IS NOT SET
020312                                  1$:
020312  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020314  021204                                  021204                  ;PORT4+IN DATA
020316  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
020322  005005                                  CLR     R5              ;PUT "EXPECTED" IN R5
020324  120504                                  CMPB    R5,R4           ;WAS A 0 RECEIVED?
020326  001401                                  BEQ     2$
020330  104010                                  HLT     10              ;ERROR, RECEIVED DATA IS WRONG
020332  104400                          2$:     SCOPE                   ;SCOPE THIS TEST


                                                ;************************** TEST 35 **************************
                                                ;*BITSTUFF 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 35
                                                ;--------------
020334  012737  000035  001226          TST35:  MOV     #35,TSTNO
020342  012737  020504  001216                  MOV     #TST36,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
020350  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
020352  005061  000004                          CLR     4(R1)           ;CLEAR PORT4
020356  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020360  122117                                  122117                  ;PUT LINE UNIT IN BITSTUFF MODE
020362  004737  033374                          JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
020366  012702  000012                          MOV     #12,R2          ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
020372  012711  004000                          MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
020376  012761  000001  000004                  MOV     #1,4(R1)        ;SET BIT0 IN PORT4
020404  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020406  122111                                  122111                  ;SET SOM!
020410  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020412  122110                                  122110                  ;LOAD GARBAGE CHAR
020414  004737  032342                          JSR     PC,CHARSD       ;LOAD SILO WITH CHARACTER
020420  000125                                  125                     ;CHARACTER
020422  104415  000033                          DATACLK,        33      ;SINGLE CLOCK THE DATA
020426  104416  000002                          TIMER,  2               ;WAIT FOR INRDY
020432  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020434  021244                                  021244                  ;PORT4+LU 12
020436  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
020442  042704  000357                          BIC     #357,R4         ;CLEAR UNWANTED BITS
020446  012705  000020                          MOV     #BIT4,R5                ;PUT "EXPECTED" IN R5
020452  120504                                  CMPB    R5,R4           ;IS INRDY SET?
020454  001401                                  BEQ     1$
```

```
020456  104040                                HLT     40                 ;ERROR, INRDY IS NOT SET
020460                                1$:
020460  104414                                ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020462  021204                                021204                     ;PORT4+IN DATA
020464  016104  000004                        MOV     4(R1),R4           ;PUT "FOUND" IN R4
020470  012705  000125                        MOV     #125,R5            ;PUT "EXPECTED" IN R5
020474  120504                                CMPB    R5,R4              ;WAS A 125 RECEIVED?
020476  001401                                BEQ     2$
020500  104010                                HLT     10                 ;ERROR, RECEIVED DATA IS WRONG
020502  104400                        2$:     SCOPE                      ;SCOPE THIS TEST


                                              ;*************************** TEST 36 ***************************
                                              ;*BITSTUFF 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 36
                                              ;---------------
020504  012737  000036  001226        TST36:  MOV     #36,TSTNO
020512  012737  020654  001216                MOV     #TST37,NEXT
                                                                         ;R1 CONTAINS BASE DMC11 ADDRESS
020520  104412                                MSTCLR                     ;MASTER CLEAR DMC11
020522  005061  000004                        CLR     4(R1)              ;CLEAR PORT4
020526  104414                                ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020530  122117                                122117                     ;PUT LINE UNIT IN BITSTUFF MODE
020532  004737  033374                        JSR     PC,CLRIO           ;DO THIS AFTER MODE IS SET
020536  012702  000012                        MOV     #12,R2             ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
020542  012711  004000                        MOV     #BIT11,(R1)        ;SET LINE UNIT LOOP
020546  012761  000001  000004                MOV     #1,4(R1)           ;SET BIT0 IN PORT4
020554  104414                                ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020556  122111                                122111                     ;SET SOM!
020560  104414                                ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020562  122110                                122110                     ;LOAD GARBAGE CHAR
020564  004737  032342                        JSR     PC,CHARSD          ;LOAD SILO WITH CHARACTER
020570  000252                                252                        ;CHARACTER
020572  104415  000033                        DATACLK,        33         ;SINGLE CLOCK THE DATA
020576  104416  000002                        TIMER,  2                  ;WAIT FOR INRDY
020602  104414                                ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020604  021244                                021244                     ;PORT4+LU 12
020606  016104  000004                        MOV     4(R1),R4           ;PUT "FOUND" IN R4
020612  042704  000357                        BIC     #357,R4            ;CLEAR UNWANTED BITS
020616  012705  000020                        MOV     #BIT4,R5                   ;PUT "EXPECTED" IN R5
020622  120504                                CMPB    R5,R4              ;IS INRDY SET?
020624  001401                                BEQ     1$
020626  104040                                HLT     40                 ;ERROR, INRDY IS NOT SET
020630                                1$:
020630  104414                                ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020632  021204                                021204                     ;PORT4+IN DATA
020634  016104  000004                        MOV     4(R1),R4           ;PUT "FOUND" IN R4
020640  012705  000252                        MOV     #252,R5            ;PUT "EXPECTED" IN R5
020644  120504                                CMPB    R5,R4              ;WAS A 252 RECEIVED?
020646  001401                                BEQ     2$
020650  104010                                HLT     10                 ;ERROR, RECEIVED DATA IS WRONG
020652  104400                        2$:     SCOPE                      ;SCOPE THIS TEST
```

```


                                                                ;*************************** TEST 37 ***************************
                                                                ;*BITSTUFF 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 37
                                                                ;--------------
 020654  012737  000037  001226                         TST37:  MOV     #37,TSTNO
 020662  012737  021024  001216                                 MOV     #TST40,NEXT
                                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
 020670  104412                                                 MSTCLR                  ;MASTER CLEAR DMC11
 020672  005061  000004                                         CLR     4(R1)           ;CLEAR PORT4
 020676  104414                                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 020700  122117                                                 122117                  ;PUT LINE UNIT IN BITSTUFF MODE
 020702  004737  033374                                         JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
 020706  012702  000012                                         MOV     #12,R2          ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
 020712  012711  004000                                         MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
 020716  012761  000001  000004                                 MOV     #1,4(R1)        ;SET BIT0 IN PORT4
 020724  104414                                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 020726  122111                                                 122111                  ;SET SOM!
 020730  104414                                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 020732  122110                                                 122110                  ;LOAD GARBAGE CHAR
 020734  004737  032342                                         JSR     PC,CHARSC       ;LOAD SILO WITH CHARACTER
 020740  000377                                                 377                     ;CHARACTER
 020742  104415  000034                                         DATACLK,        34      ;SINGLE CLOCK THE DATA
 020746  104416  000002                                         TIMER,  2               ;WAIT FOR INRDY
 020752  104414                                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 020754  021244                                                 021244                  ;PORT4←LU 12
 020756  016104  000004                                         MOV     4(R1),R4        ;PUT "FOUND" IN R4
 020762  042704  000357                                         BIC     #357,R4         ;CLEAR UNWANTED BITS
 020766  012705  000020                                         MOV     #BIT4,R5                ;PUT "EXPECTED" IN R5
 020772  120504                                                 CMPB    R5,R4           ;IS INRDY SET?
 020774  001401                                                 BEQ     1$
 020776  104040                                                 HLT     40              ;ERROR, INRDY IS NOT SET
 021000                                                 1$:
 021000  104414                                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 021002  021204                                                 021204                  ;PORT4←IN DATA
 021004  016104  000004                                         MOV     4(R1),R4        ;PUT "FOUND" IN R4
 021010  012705  000377                                         MOV     #377,R5         ;PUT "EXPECTED" IN R5
 021014  120504                                                 CMPB    R5,R4           ;WAS A 377 RECEIVED?
 021016  001401                                                 BEQ     2$
 021020  104010                                                 HLT     10              ;ERROR, RECEIVED DATA IS WRONG
 021022  104400                                         2$:     SCOPE                   ;SCOPE THIS TEST


                                                                ;*************************** TEST 40 ***************************
                                                                ;*BITSTUFF DATA TEST
                                                                ;*THIS TEST SINGLE STEPS A BINARY COUNT PATTERN
                                                                ;*CHECKING EACH CHARACTER AS IT IS RECEIVED
                                                                ;;**************************************************************


                                                                ;  TEST 40
                                                                ;--------------
```

```
021024  012737  000040  001226          TST40:  MOV     #40,TSTNO
021032  012737  021200  001216                  MOV     #TST41,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
021040  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
021042  005061  000004                          CLR     4(R1)           ;CLEAR PORT4
021046  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021050  122117                                  122117                  ;PUT LINE UNIT IN BITSTUFF MODE
021052  004737  033374                          JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
021056  005037  032646                          CLR     SCHAR           ;START BINARY COUNT AT ZERO
021062  005137  032646                          COM     SCHAR           ;IF BITSTUFF SCHAR IS MINUS NUMBER
021066  005037  033612                          CLR     BITCON          ;START  1'S COUNT AT 0
021072  005037  032650                          CLR     STUFLG          ;CLEAR BITSTUFF FLAG
021076  005002                                  CLR     R2              ;R2 IS "EXPECTED" DATA
021100  012703  000073                          MOV     #73,R3          ;R3 IS CHARACTER COUNT
021104  012711  004000                          MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
021110  004737  032406                          JSR     PC,SILOLD       ;LOAD SILO WITH COUNT PATTERN
021114  104415  000023                          DATACLK,        23      ;SYNC RECEIVER AND GET IT ACTIVE
021120  104415  000730                  1$:     DATACLK,        730     ;CLOCK IN 73 CHARACTERS
021124  004737  032652                  4$:     JSR     PC,INRDY        ;WAIT FOR INRDY
021130  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021132  021204                                  021204                  ;PORT4←IN DATA
021134  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
021140  010205                                  MOV     R2,R5           ;PUT "EXPECTED" IN R5
021142  120504                                  CMPB    R5,R4           ;IS DATA CORRECT?
021144  001401                                  BEQ     2$              ;BR IF YES
021146  104010                                  HLT     10              ;DATA ERROR
021150  005202                          2$:     INC     R2              ;NEXT CHARACTER
021152  022702  000400                          CMP     #400,R2         ;ALL DONE?
021156  001407                                  BEQ     3$              ;BR IF YES
021160  005303                                  DEC     R3              ;DECREMENT CHARACTER COUNT
021162  001360                                  BNE     4$              ;BR IF SILO NOT EMPTY
021164  004737  032406                          JSR     PC,SILOLD       ;LOAD SILO WITH MORE OF COUNT PATTERN
021170  012703  000073                          MOV     #73,R3          ;RELOAD CHARACTER COUNT
021174  000751                                  BR      1$              ;CONTINUE
021176  104400                          3$:     SCOPE                   ;SCOPE THIS TEST



                                                ;*************************** TEST 41 ***************************
                                                ;*BITSTUFF 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 41
                                                ;--------------
021200  012737  000041  001226          TST41:  MOV     #41,TSTNO
021206  012737  021364  001216                  MOV     #TST42,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
021214  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
021216  005061  000004                          CLR     4(R1)           ;CLEAR PORT4
021222  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021224  122117                                  122117                  ;PUT LINE UNIT IN BITSTUFF MODE
021226  004737  033374                          JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
021232  005037  032646                          CLR     SCHAR           ;START BINARY COUNT AT ZERO
```

```
021236  005137  032646                      COM     SCHAR           ;IF BITSTUFF SCHAR IS MINUS NUMBER
021242  005037  033612                      CLR     BITCON          ;START  1'S COUNT AT 0
021246  005037  032650                      CLR     STUFLG          ;CLEAR BITSTUFF FLAG
021252  005002                              CLR     R2              ;R2 IS "EXPECTED" DATA
021254  012703  000073                      MOV     #73,R3          ;R3 IS CHARACTER COUNT
021260  005011                              CLR     (R1)            ;CLEAR LU LOOP IN MAINT REG
021262  012761  000040  000004              MOV     #BITS,4(R1)     ;LOAD PORT4
021270  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021272  122112                              122112                  ;SET LU LOOP IN LU REG 12
021274  004737  032406                      JSR     PC,SILOLD       ;LOAD SILO WITH COUNT PATTERN
021300  104415  000023                      DATACLK,        23      ;SYNC RECEIVER AND GET IT ACTIVE
021304  104415  000730              1$:     DATACLK,        730     ;CLOCK IN 73 CHARACTERS
021310  004737  032652              4$:     JSR     PC,INRDY        ;WAIT FOR INRDY
021314  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021316  021204                              021204                  ;PORT4+IN DATA
021320  016104  000004                      MOV     4(R1),R4        ;PUT "FOUND" IN R4
021324  010205                              MOV     R2,R5           ;PUT "EXPECTED" IN R5
021326  120504                              CMPB    R5,R4           ;IS DATA CORRECT?
021330  001401                              BEQ     2$              ;BR IF YES
021332  104010                              HLT     10              ;DATA ERROR
021334  005202                      2$:     INC     R2              ;NEXT CHARACTER
021336  022702  000400                      CMP     #400,R2         ;ALL DONE?
021342  001407                              BEQ     3$              ;BR IF YES
021344  005303                              DEC     R3              ;DECREMENT CHARACTER COUNT
021346  001360                              BNE     4$              ;BR IF SILO NOT EMPTY
021350  004737  032406                      JSR     PC,SILOLD       ;LOAD SILO WITH MORE OF COUNT PATTERN
021354  012703  000073                      MOV     #73,R3          ;RELOAD CHARACTER COUNT
021360  000751                              BR      1$              ;CONTINUE
021362  104400                      3$:     SCOPE                   ;SCOPE THIS TEST



                                            ;*************************** TEST 42 ***************************
                                            ;*RECEIVER ABORT TEST
                                            ;*SINGLE CLOCK 3 FLAGS, A 301, ANOTHER 301 AND 10 EXTRA
                                            ;*CLOCK TICKS, VERIFY THAT A 301 AND A BLOCK END
                                            ;*WERE RECEIVED INDICATING THAT THE RECEIVER RECOGINIZED
                                            ;*THE ABORT SEQUENCE (8 CONTIGUIOUS 1'S)
                                            ;;**************************************************************


                                            ;  TEST 42
                                            ;--------------
021364  012737  000042  001226      TST42:  MOV     #42,TSTNO
021372  012737  021526  001216              MOV     #TST43,NEXT
                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
021400  104412                              MSTCLR                  ;MASTER CLEAR DMC11
021402  005061  000004                      CLR     4(R1)
021406  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021410  122117                              122117                  ;PUT LINE UNIT IN BITSTUFF MODE
021412  004737  033374                      JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
021416  012711  004000                      MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
021422  004737  032230                      JSR     PC,CHAR         ;LOAD SILO WITH 3 FLAGS
021426  000301                              301                     ;AND A 301
021430  004737  032176                      JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
021434  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021436  122110                              122110                  ;LOAD 2ND 301 CHARACTER
021440  104415  000073                      DATACLK,        73      ;CLOCK THE 301 IN AND 10 EXTRA TICKS
```

```
021444  004737  032652                   JSR     PC,INRDY        ;WAIT FOR INRDY
021450  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021452  021204                           021204                  ;PORT4←IN DATA
021454  016104  000004                   MOV     4(R1),R4        ;PUT "FOUND" IN R4
021460  012705  000301                   MOV     #301,R5         ;PUT "EXPECTED" IN R5
021464  120504                           CMPB    R5,R4           ;WAS A 301 RECEIVED?
021466  001401                           BEQ     1$
021470  104010                           HLT     10              ;ERROR FIRST CHARACTER INCORRECT
021472  004737  032652           1$:     JSR     PC,INRDY        ;WAIT FOR INRDY
021476  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021500  021244                           021244                  ;READ LU-12
021502  016104  000004                   MOV     4(R1),R4        ;PUT "FOUND" IN R4
021506  042704  000375                   BIC     #375,R4         ;CLEAR UNWANTED BITS
021512  012705  000002                   MOV     #2,R5           ;PUT "EXPECTED" IN R5
021516  120504                           CMPB    R5,R4           ;IS BLOCK END SET?
021520  001401                           BEQ     3$              ;BR IF YES
021522  104032                           HLT     32              ;ERROR, BLOCK END NOT SET
021524  104400                   3$:     SCOPE                   ;SCOPE THIS TEST


                                         ;*************************** TEST 43 ***************************
                                         ;*CABLE TURNAROUND TEST
                                         ;*CLEAR LINE UNIT LOOP, SET DTR
                                         ;*VERIFY THAT MODEM READY IS SET
                                         ;*CLEAR DTR, VERIFY THAT MRDY IS CLEARED
                                         ;;**************************************************************

                                         ;  TEST 43
                                         ;--------------
021526  012737  000043  001226   TST43:  MOV     #43,TSTNO
021534  012737  021710  001216           MOV     #TST44,NEXT
                                                                 ;R1 CONTAINS BASE DMC11 ADDRESS
021542  104412                           MSTCLR                  ;MASTER CLEAR DMC11
021544  032737  020000  001366           BIT     #BIT13,STAT1    ;IS LINE UNIT M8202?
021552  001004                           BNE     .+12            ;BR IF YES (DO TEST EVEN IF NO LOOP-BACK CONN.
021554  032737  040000  001366           BIT     #BIT14,STAT1    ;IS TURNAROUND CONNECTOR ON?
021562  001451                           BEQ     2$              ;SKIP TEST IF NO
021564  005011                           CLR     (R1)            ;CLEAR LINE UNIT LOOP
021566  012761  000100  000004           MOV     #100,4(R1)      ;LOAD PORT4
021574  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021576  122113                           122113                  ;SET DTR
021600  104416  000002                   TIMER,  2               ;WAIT
021604  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021606  021264                           021264                  ;PORT4←LU13
021610  016104  000004                   MOV     4(R1),R4        ;PUT "FOUND" IN R4
021614  042704  000223                   BIC     #223,R4         ;CLEAR UNWANTED BITS
021620  012705  000110                   MOV     #110,R5         ;PUT "EXPECTED" IN R5
021624  120504                           CMPB    R5,R4           ;IS MRDY SET?
021626  001401                           BEQ     1$
021630  104011                           HLT     11              ;ERROR, MRDY NOT SET
021632  005061  000004           1$:     CLR     4(R1)           ;CLEAR PORT4
021636  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021640  122113                           122113                  ;CLEAR DTR
021642  104416  000002                   TIMER,  2
021646  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021650  021264                           021264                  ;PORT4←LU13
```

```
021652  016104  000004                       MOV     4(R1),R4          ;PUT "FOUND" IN R4
021656  042704  000223                       BIC     #223,R4           ;CLEAR UNWANTED BITS
021662  005005                               CLR     R5                ;PUT "EXPECTED" IN R5
021664  032737  020000  001366               BIT     #BIT13,STAT1      ;IS LINE UNIT M8202?
021672  001402                               BEQ     .+6               ;BR IF NO
021674  052705  000010                       BIS     #BIT3,R5          ;MRDY SET ON M8202
021700  120504                               CMPB    R5,R4             ;IS MRDY CLEAR?
021702  001401                               BEQ     2$
021704  104011                               HLT     11                ;ERROR, MRDY NOT CLEAR
021706  104400                       2$:     SCOPE                     ;SCOPE THIS TEST



                                             ;*************************** TEST 44 **************************
                                             ;*CABLE TURNAROUND TEST
                                             ;*CLEAR LINE UNIT LOOP, LOAD OUT DATA SILO
                                             ;*VERIFY THAT ALL MODEM SIGNALS ARE SET
                                             ;;*************************************************************

                                             ;  TEST 44
                                             ;---------------
021710  012737  000044  001226       TST44:  MOV     #44,TSTNO
021716  012737  022054  001216               MOV     #TST45,NEXT
                                                                       ;R1 CONTAINS BASE DMC11 ADDRESS
021724  104412                               MSTCLR                    ;MASTER CLEAR DMC11
021726  032737  020000  001366               BIT     #BIT13,STAT1      ;IS LINE UNIT M8202?
021734  001004                               BNE     .+12              ;BR IF YES (DO TEST EVEN IF NO LOOP-BACK CONN
021736  032737  040000  001366               BIT     #BIT14,STAT1      ;IS TURNAROUND CONNECTOR ON?
021744  001442                               BEQ     1$                ;SKIP TEST IF NO
021746  012711  004000                       MOV     #BIT11,(R1)       ;SET LINE UNIT LOOP
021752  012761  000100  000004               MOV     #100,   4(R1)          ;LOAD PORT4
021760  104414                               ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021762  122113                               122113                    ;CLEAR ALL MODEM SIGNALS,EXCEPT DTR
021764  104416  000002                       TIMER,  2                 ;WAIT
021770  012761  000001  000004               MOV     #1,4(R1)          ;LOAD PORT4
021776  104414                               ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022000  122111                               122111                    ;SET SOM
022002  004537  033332                       JSR     R5,MESLD          ;FILL OUT DATA SILO
022006  033614                               MESDAT                    ;WITH 64 CHARACTERS
022010  000100                               64.
022012  012700  000050                       MOV     #50,R0            ;PREPARE FOR DELAY
022016  005011                               CLR     (R1)              ;CLEAR LINE UNIT LOOP
022020                               2$:
022020  104414                               ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022022  021264                               021264                    ;PORT4←LU13
022024  016104  000004                       MOV     4(R1),R4          ;PUT "FOUND" IN R4
022030  042704  000223                       BIC     #223,R4           ;CLEAR UNWANTED BITS
022034  012705  000154                       MOV     #154,R5           ;PUT "EXPECTED" IN R5
022040  120504                               CMPB    R5,R4             ;COMPARE EXPECTED AND FOUND
022042  001403                               BEQ     1$                ;BR IF OK
022044  005300                               DEC     R0                ;DEC DELAY COUNT
022046  001364                               BNE     2$                ;BR IF NOT ZERO
022050  104011                               HLT     11                ;ERROR, ALL SIGNALS ARE NOT SET
022052  104400                       1$:     SCOPE                     ;SCOPE THIS TEST


                                             ;*************************** TEST 45 **************************
```

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

                                                       ;  TEST 45
                                                       ;---------------
022054  012737  000045  001226        TST45:  MOV     #45,TSTNO
022062  012737  022420  001216                MOV     #TST46,NEXT
022070  012737  022124  001220                MOV     #64$,LOCK
                                                                               ;R1 CONTAINS BASE DMC11 ADDRESS
022076  104412                                MSTCLR                           ;MASTER CLEAR DMC11
022100  005061  000004                        CLR     4(R1)                    ;CLEAR PORT4
022104  104414                                ROMCLK                           ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022106  122117                                122117                           ;PUT LINE UNIT IN BITSTUFF MODE
022110  004737  033374                        JSR     PC,CLRIO                 ;DO THIS AFTER MODE IS SET
022114  005037  033612                        CLR     BITCON                   ;CONSECUTIVE 1'S COUNTER INIT TO 0
022120  012711  004000                        MOV     #BIT11,(R1)              ;SET LU LOOP
022124  004737  033374                64$:    JSR     PC,CLRIO                 ;CLEAR BCC REGISTERS
022130  005000                                CLR     R0                       ;START SHIFT COUNTER AT ZERO
022132  012737  102010  033030                MOV     #CRC.CCITT,XPOLY;LOAD POLYNOMIAL FOR SOFTWARE BCC
022140  012737  000000  022204                MOV     #0,66$                   ;LOAD CHAR FOR SOFTWARE BCC
022146  005037  022206                        CLR     67$                      ;CLEAR OLD SOFTWARE BCC
022152  005137  022206                        COM     67$                      ;START AT -1
022156  004737  033034                        JSR     PC,BCCLD                 ;LOAD OUT SILO WITH 2 SYNCS
022162  000000                                0                                ;AND THE CHARACTER 0
022164  104415  000021                        DATACLK,        21               ;GET TRANSMITTER ACTIVE
022170  104415  000001                65$:    DATACLK,        1                ;SHIFT BCC ONCE
022174  005200                                INC     R0                       ;BUMP SHIFT COUNT
022176  004537  032706                        JSR     R5,SIMBCC                ;CALCULATE SOFTWARE BCC LSB
022202  000001                                1                                ;ONE SHIFT
022204  000000                        66$:    0                                ;DATA CHARACTER
022206  000000                        67$:    0                                ;OLD BCC
022210  103405                                BCS     68$                      ;BR IF SOFT BCC LSB IS SET
022212  004737  033146                        JSR     PC,GETQO                 ;GET HARDWARE TRANSMITTER BCC LSB
022216  103006                                BCC     69$                      ;BR IF HARD BCC LSB IS CLEAR
022220  104012                                HLT     12                       ;ERROR, BCC LSB IS SET
022222  000404                                BR      69$                      ;CONTINUE
022224  004737  033146                68$:    JSR     PC,GETQO                 ;GET HARDWARE TRANSMITTER BCC LSB
022230  103401                                BCS     69$                      ;BR IF HARD BCC LSB IS SET
022232  104016                                HLT     16                       ;ERROR, HARD BCC LSB IS CLEAR
022234                                69$:
022234  006037  022204                        ROR     66$                      ;SHIFT SOFT DATA
022240  013737  033032  022206                MOV     CALBCC,67$               ;LOAD OLD SOFT BCC
022246  022700  000010                        CMP     #10,R0                   ;DONE YET?
022252  001346                                BNE     65$                      ;BR IF NOT DONE
022254  104401                                SCOP1                            ;SCOPE SUBTEST (SW09=1)
022256  012737  022264  001220                MOV     #71$,LOCK                ;NEW SCOPE1
022264  004737  033374                71$:    JSR     PC,CLRIO                 ;CLEAR BCC REGISTERS
022270  005000                                CLR     R0                       ;START SHIFT COUNTER AT ZERO
022272  012737  102010  033030                MOV     #CRC.CCITT,XPOLY;LOAD POLYNOMIAL FOR SOFTWARE BCC
022300  012737  000000  022344                MOV     #0,73$                   ;LOAD CHAR FOR SOFTWARE BCC
022306  005037  022346                        CLR     74$                      ;CLEAR OLD SOFTWARE BCC
022312  005137  022346                        COM     74$                      ;START AT -1
022316  004737  033034                        JSR     PC,BCCLD                 ;LOAD OUT SILO WITH 2 SYNCS
```

```
022322  000000                                  0                          ;AND THE CHARACTER 0
022324  104415  000032                          DATACLK,        32         ;GET RECEIVER ACTIVE
022330  104415  000001                  72$:    DATACLK,        1          ;SHIFT BCC ONCE
022334  005200                                  INC     R0                 ;BUMP SHIFT COUNT
022336  004537  032706                          JSR     R5,SIMBCC          ;CALCULATE SOFTWARE BCC LSB
022342  000001                                  1                          ;ONE SHIFT
022344  000000                          73$:    0                          ;DATA CHARACTER
022346  000000                          74$:    0                          ;OLD BCC
022350  103405                                  BCS     75$                ;BR IF SOFT BCC LSB IS SET
022352  004737  033160                          JSR     PC,GETQI           ;GET HARDWARE RECEIVER BCC LSB
022356  103006                                  BCC     76$                ;BR IF HARD BCC LSB IS CLEAR
022360  104013                                  HLT     13                 ;ERROR, BCC LSB IS SET
022362  000404                                  BR      76$                ;CONTINUE
022364  004737  033160                  75$:    JSR     PC,GETQI           ;GET HARDWARE RECEIVER BCC LSB
022370  103401                                  BCS     76$                ;BR IF HARD BCC LSB IS SET
022372  104017                                  HLT     17                 ;ERROR, BCC LSB IS CLEAR
022374                                  76$:
022374  006037  022344                          ROR     73$                ;SHIFT SOFT DATA
022400  013737  033032  022346                  MOV     CALBCC,74$         ;LOAD OLD SOFT BCC
022406  022700  000010                          CMP     #10,R0             ;DONE YET?
022412  001346                                  BNE     72$                ;BR IF NOT DONE
022414  104401                                  SCOP1                      ;SCOPE SUBTEST (SW09=1)
022416  104400                          77$:    SCOPE                      ;SCOPE THIS TEST


                                                ;*************************** TEST 46 ***************************
                                                ;*TEST OF CRC OPERATION
                                                ;*USING THE CRC.CCITT POLYNOMIAL, SINGLE CLOCK THE CHARACTER
                                                ;*377, VERIFY THE LSB OF THE BCC ON EACH SHIFT
                                                ;*TEST TRANSMITTER FIRST THEN THE RECEIVER BCC
                                                ;;*************************************************************

                                                ;  TEST 46
                                                ;--------------
022420  012737  000046  001226          TST46:  MOV     #46,TSTNO
022426  012737  023012  001216                  MOV     #TST47,NEXT
022434  012737  022470  001220                  MOV     #64$,LOCK
                                                                           ;R1 CONTAINS BASE DMC11 ADDRESS
022442  104412                                  MSTCLR                     ;MASTER CLEAR DMC11
022444  005061  000004                          CLR     4(R1)              ;CLEAR PORT4
022450  104414                                  ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022452  122117                                  122117                     ;PUT LINE UNIT IN BITSTUFF MODE
022454  004737  033374                          JSR     PC,CLRIO           ;DO THIS AFTER MODE IS SET
022460  005037  033612                          CLR     BITCON             ;CONSECUTIVE 1'S COUNTER INIT TO 0
022464  012711  004000                          MOV     #BIT11,(R1)        ;SET LU LOOP
022470  004737  033374                  64$:    JSR     PC,CLRIO           ;CLEAR BCC REGISTERS
022474  005000                                  CLR     R0                 ;START SHIFT COUNTER AT ZERO
022476  012737  102010  033030                  MOV     #CRC.CCITT,XPOLY;LOAD POLYNOMIAL FOR SOFTWARE BCC
022504  012737  000377  022570                  MOV     #377,66$;          ;LOAD CHAR FOR SOFTWARE BCC
022512  005037  022572                          CLR     67$                ;CLEAR OLD SOFTWARE BCC
022516  005137  022572                          COM     67$                ;START AT -1
022522  004737  033034                          JSR     PC,BCCLD           ;LOAD OUT SILO WITH 2 SYNCS
022526  000377                                  377                        ;AND THE CHARACTER 377
022530  104415  000021                          DATACLK,        21         ;GET TRANSMITTER ACTIVE
022534  005037  033612                          CLR     BITCON             ;CLEAR BIT COUNTER
022540  005037  022554                          CLR     60$
```

```
022544  104415  000001                   65$:    DATACLK,         1      ;SHIFT BCC ONCE
022550  004537  033474                           JSR     R5,STFFCK       ;CHECK FOR STUFFING ZEROS
022554  000000                           60$:    0                       ;CHARACTER
022556  000001                                   1                       ;SHIFT COUNT
022560  005200                                   INC     R0              ;BUMP SHIFT COUNT
022562  004537  032706                           JSR     R5,SIMBCC       ;CALCULATE SOFTWARE BCC LSB
022566  000001                                   1                       ;ONE SHIFT
022570  000000                           66$:    0                       ;DATA CHARACTER
022572  000000                           67$:    0                       ;OLD BCC
022574  103405                                   BCS     68$             ;BR IF SOFT BCC LSB IS SET
022576  004737  033146                           JSR     PC,GETQ0        ;GET HARDWARE TRANSMITTER BCC LSB
022602  103006                                   BCC     69$             ;BR IF HARD BCC LSB IS CLEAR
022604  104012                                   HLT     12              ;ERROR, BCC LSB IS SET
022606  000404                                   BR      69$             ;CONTINUE
022610  004737  033146                   68$:    JSR     PC,GETQ0        ;GET HARDWARE TRANSMITTER BCC LSB
022614  103401                                   BCS     69$             ;BR IF HARD BCC LSB IS SET
022616  104016                                   HLT     16              ;ERROR, HARD BCC LSB IS CLEAR
022620                                   69$:
022620  013737  022570  022554                   MOV     66$,60$
022626  006037  022570                           ROR     66$             ;SHIFT SOFT DATA
022632  013737  033032  022572                   MOV     CALBCC,67$      ;LOAD OLD SOFT BCC
022640  022700  000010                           CMP     #10,R0          ;DONE YET?
022644  001337                                   BNE     65$             ;BR IF NOT DONE
022646  104401                                   SCOP1                   ;SCOPE SUBTEST (SW09=1)
022650  012737  022656  001220                   MOV     #71$,LOCK       ;NEW SCOPE1
022656  004737  033374                   71$:    JSR     PC,CLRIO        ;CLEAR BCC REGISTERS
022662  005000                                   CLR     R0              ;START SHIFT COUNTER AT ZERO
022664  012737  102010  033030                   MOV     #CRC.CCITT,XPOLY;LOAD POLYNOMIAL FOR SOFTWARE BCC
022672  012737  000377  022736                   MOV     #377,73$:       ;LOAD CHAR FOR SOFTWARE BCC
022700  005037  022740                           CLR     74$             ;CLEAR OLD SOFTWARE BCC
022704  005137  022740                           COM     74$             ;START AT -1
022710  004737  033034                           JSR     PC,BCCLD        ;LOAD OUT SILO WITH 2 SYNCS
022714  000377                                   377                     ;AND THE CHARACTER 377
022716  104415  000033                           DATACLK,        33      ;GET RECEIVER ACTIVE
022722  104415  000001                   72$:    DATACLK,         1      ;SHIFT BCC ONCE
022726  005200                                   INC     R0              ;BUMP SHIFT COUNT
022730  004537  032706                           JSR     R5,SIMBCC       ;CALCULATE SOFTWARE BCC LSB
022734  000001                                   1                       ;ONE SHIFT
022736  000000                           73$:    0                       ;DATA CHARACTER
022740  000000                           74$:    0                       ;OLD BCC
022742  103405                                   BCS     75$             ;BR IF SOFT BCC LSB IS SET
022744  004737  033160                           JSR     PC,GETQI        ;GET HARDWARE RECEIVER BCC LSB
022750  103006                                   BCC     76$             ;BR IF HARD BCC LSB IS CLEAR
022752  104013                                   HLT     13              ;ERROR, BCC LSB IS SET
022754  000404                                   BR      76$             ;CONTINUE
022756  004737  033160                   75$:    JSR     PC,GETQI        ;GET HARDWARE RECEIVER BCC LSB
022762  103401                                   BCS     76$             ;BR IF HARD BCC LSB IS SET
022764  104017                                   HLT     17              ;ERROR, BCC LSB IS CLEAR
022766                                   76$:
022766  006037  022736                           ROR     73$             ;SHIFT SOFT DATA
022772  013737  033032  022740                   MOV     CALBCC,74$      ;LOAD OLD SOFT BCC
023000  022700  000010                           CMP     #10,R0          ;DONE YET?
023004  001346                                   BNE     72$             ;BR IF NOT DONE
023006  104401                                   SCOP1                   ;SCOPE SUBTEST (SW09=1)
023010  104400                           77$:    SCOPE                   ;SCOPE THIS TEST
```

```

                                                      ;*************************** TEST 47 ***************************
                                                      ;*TEST OF CRC OPERATION
                                                      ;*USING THE CRC.CCITT POLYNOMIAL, SINGLE CLOCK THE CHARACTER
                                                      ;*125, VERIFY THE LSB OF THE BCC ON EACH SHIFT
                                                      ;*TEST TRANSMITTER FIRST THEN THE RECEIVER BCC
                                                      ;;*****************************************************************

                                                      ;  TEST 47
                                                      ;---------------
023012  012737  000047  001226          TST47:  MOV     #47,TSTNO
023020  012737  023356  001216                  MOV     #TST50,NEXT
023026  012737  023062  001220                  MOV     #64$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
023034  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
023036  005061  000004                          CLR     4(R1)           ;CLEAR PORT4
023042  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=53C4
023044  122117                                  122117                  ;PUT LINE UNIT IN BITSTUFF MODE
023046  004737  033374                          JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
023052  005037  033612                          CLR     BITCON          ;CONSECUTIVE 1'S COUNTER INIT TO 0
023056  012711  004000                          MOV     #BIT11,(R1)     ;SET LU LOOP
023062  004737  033374                  64$:    JSR     PC,CLRIO        ;CLEAR BCC REGISTERS
023066  005000                                  CLR     R0              ;START SHIFT COUNTER AT ZERO
023070  012737  102010  033030                  MOV     #CRC.CCITT,XPOLY;LOAD POLYNOMIAL FOR SOFTWARE BCC
023076  012737  000125  023142                  MOV     #125,66$;       ;LOAD CHAR FOR SOFTWARE BCC
023104  005037  023144                          CLR     67$             ;CLEAR OLD SOFTWARE BCC
023110  005137  023144                          COM     67$             ;START AT -1
023114  004737  033034                          JSR     PC,BCCLD        ;LOAD OUT SILO WITH 2 SYNCS
023120  000125                                  125                     ;AND THE CHARACTER 125
023122  104415  000021                          DATACLK,        21      ;GET TRANSMITTER ACTIVE
023126  104415  000001                  65$:    DATACLK,        1       ;SHIFT BCC ONCE
023132  005200                                  INC     R0              ;BUMP SHIFT COUNT
023134  004537  032706                          JSR     R5,SIMBCC       ;CALCULATE SOFTWARE BCC LSB
023140  000001                                  1                       ;ONE SHIFT
023142  000000                          66$:    0                       ;DATA CHARACTER
023144  000000                          67$:    0                       ;OLD BCC
023146  103405                                  BCS     68$             ;BR IF SOFT BCC LSB IS SET
023150  004737  033146                          JSR     PC,GETQ0        ;GET HARDWARE TRANSMITTER BCC LSB
023154  103006                                  BCC     69$             ;BR IF HARD BCC LSB IS CLEAR
023156  104012                                  HLT     12              ;ERROR, BCC LSB IS SET
023160  000404                                  BR      69$             ;CONTINUE
023162  004737  033146                  68$:    JSR     PC,GETQ0        ;GET HARDWARE TRANSMITTER BCC LSB
023166  103401                                  BCS     69$             ;BR IF HARD BCC LSB IS SET
023170  104016                                  HLT     16              ;ERROR, HARD BCC LSB IS CLEAR
023172                                  69$:
023172  006037  023142                          ROR     66$             ;SHIFT SOFT DATA
023176  013737  033032  023144                  MOV     CALBCC,67$      ;LOAD OLD SOFT BCC
023204  022700  000010                          CMP     #10,R0          ;DONE YET?
023210  001346                                  BNE     65$             ;BR IF NOT DONE
023212  104401                                  SCOP1                   ;SCOPE SUBTEST (SW09=1)
023214  012737  023222  001220                  MOV     #71$,LOCK       ;NEW SCOPE1
023222  004737  033374                  71$:    JSR     PC,CLRIO        ;CLEAR BCC REGISTERS
023226  005000                                  CLR     R0              ;START SHIFT COUNTER AT ZERO
023230  012737  102010  033030                  MOV     #CRC.CCITT,XPOLY;LOAD POLYNOMIAL FOR SOFTWARE BCC
023236  012737  000125  023302                  MOV     #125,73$;       ;LOAD CHAR FOR SOFTWARE BCC
023244  005037  023304                          CLR     74$             ;CLEAR OLD SOFTWARE BCC
```

```
023250  005137  023304                     COM     74$             ;START AT -1
023254  004737  033034                     JSR     PC,BCCLD        ;LOAD OUT SILO WITH 2 SYNCS
023260  000125                             125                     ;AND THE CHARACTER 125
023262  104415  000032                     DATACLK,        32      ;GET RECEIVER ACTIVE
023266  104415  000001             72$:    DATACLK,        1       ;SHIFT BCC ONCE
023272  005200                             INC     R0              ;BUMP SHIFT COUNT
023274  004537  032706                     JSR     R5,SIMBCC       ;CALCULATE SOFTWARE BCC LSB
023300  000001                             1                       ;ONE SHIFT
023302  000000                     73$:    0                       ;DATA CHARACTER
023304  000000                     74$:    0                       ;OLD BCC
023306  103405                             BCS     75$             ;BR IF SOFT BCC LSB IS SET
023310  004737  033160                     JSR     PC,GETQI        ;GET HARDWARE RECEIVER BCC LSB
023314  103006                             BCC     76$             ;BR IF HARD BCC LSB IS CLEAR
023316  104013                             HLT     13              ;ERROR, BCC LSB IS SET
023320  000404                             BR      76$             ;CONTINUE
023322  004737  033160             75$:    JSR     PC,GETQI        ;GET HARDWARE RECEIVER BCC LSB
023326  103401                             BCS     76$             ;BR IF HARD BCC LSB IS SET
023330  104017                             HLT     17              ;ERROR, BCC LSB IS CLEAR
023332                             76$:
023332  006037  023302                     ROR     73$             ;SHIFT SOFT DATA
023336  013737  033032  023304             MOV     CALBCC,74$      ;LOAD OLD SOFT BCC
023344  022700  000010                     CMP     #10,R0          ;DONE YET?
023350  001346                             BNE     72$             ;BR IF NOT DONE
023352  104401                             SCOP1                   ;SCOPE SUBTEST (SW09=1)
023354  104400                     77$:    SCOPE                   ;SCOPE THIS TEST


                                           ;**************************** TEST 50 *****************************
                                           ;*TEST OF CRC OPERATION
                                           ;*USING THE CRC.CCITT POLYNOMIAL, SINGLE CLOCK THE CHARACTER
                                           ;*252, VERIFY THE LSB OF THE BCC ON EACH SHIFT
                                           ;*TEST TRANSMITTER FIRST THEN THE RECEIVER BCC
                                           ;:*******************************************************************

                                           ;  TEST 50
                                           ;---------------
023356  012737  000050  001226     TST50:  MOV     #50,TSTNO
023364  012737  023722  001216             MOV     #TST51,NEXT
023372  012737  023426  001220             MOV     #64$,LOCK
                                                                   ;R1 CONTAINS BASE DMC11 ADDRESS
023400  104412                             MSTCLR                  ;MASTER CLEAR DMC11
023402  005061  000004                     CLR     4(R1)           ;CLEAR PORT4
023406  104414                             ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023410  122117                             122117                  ;PUT LINE UNIT IN BITSTUFF MODE
023412  004737  033374                     JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
023416  005037  033612                     CLR     BITCON          ;CONSECUTIVE 1'S COUNTER INIT TO 0
023422  012711  004000                     MOV     #BIT11,(R1)     ;SET LU LOOP
023426  004737  033374             64$:    JSR     PC,CLRIO        ;CLEAR BCC REGISTERS
023432  005000                             CLR     R0              ;START SHIFT COUNTER AT ZERO
023434  012737  102010  033030             MOV     #CRC.CCITT,XPOLY;LOAD POLYNOMIAL FOR SOFTWARE BCC
023442  012737  000252  023506             MOV     #252,66$;       ;LOAD CHAR FOR SOFTWARE BCC
023450  005037  023510                     CLR     67$             ;CLEAR OLD SOFTWARE BCC
023454  005137  023510                     COM     67$             ;START AT -1
023460  004737  033034                     JSR     PC,BCCLD        ;LOAD OUT SILO WITH 2 SYNCS
023464  000252                             252                     ;AND THE CHARACTER 252
023466  104415  000021                     DATACLK,        21      ;GET TRANSMITTER ACTIVE
```

```
023472  104415  000001               65$:   DATACLK,          1      ;SHIFT BCC ONCE
023476  005200                              INC     R0               ;BUMP SHIFT COUNT
023500  004537  032706                      JSR     R5,SIMBCC        ;CALCULATE SOFTWARE BCC LSB
023504  000001                              1                        ;ONE SHIFT
023506  000000                       66$:   0                        ;DATA CHARACTER
023510  000000                       67$:   0                        ;OLD BCC
023512  103405                              BCS     68$              ;BR IF SOFT BCC LSB IS SET
023514  004737  033146                      JSR     PC,GETQ0         ;GET HARDWARE TRANSMITTER BCC LSB
023520  103006                              BCC     69$              ;BR IF HARD BCC LSB IS CLEAR
023522  104012                              HLT     12               ;ERROR, BCC LSB IS SET
023524  000404                              BR      69$              ;CONTINUE
023526  004737  033146               68$:   JSR     PC,GETQ0         ;GET HARDWARE TRANSMITTER BCC LSB
023532  103401                              BCS     69$              ;BR IF HARD BCC LSB IS SET
023534  104016                              HLT     16               ;ERROP, HARD BCC LSB IS CLEAP
023536                               69$:
023536  006037  023506                      ROR     66$              ;SHIFT SOFT DATA
023542  013737  033032  023510              MOV     CALBCC,67$       ;LOAD OLD SOFT BCC
023550  022700  000010                      CMP     #10,R0           ;DONE YET?
023554  001346                              BNE     65$              ;BR IF NOT DONE
023556  104401                              SCOP1                    ;SCOPE SUBTEST (SW09=1)
023560  012737  023566  001220              MOV     #71$,LOCK        ;NEW SCOPE1
023566  004737  033374               71$:   JSR     PC,CLRIO         ;CLEAR BCC REGISTERS
023572  005000                              CLR     R0               ;START SHIFT COUNTER AT ZERO
023574  012737  102010  033030              MOV     #CRC.CCITT,XPOLY;LOAD POLYNOMIAL FOR SOFTWARE BCC
023602  012737  000252  023646              MOV     #252,73$;        ;LOAD CHAR FOR SOFTWARE BCC
023610  005037  023650                      CLR     74$              ;CLEAR OLD SOFTWARE BCC
023614  005137  023650                      COM     74$              ;START AT -1
023620  004737  033034                      JSR     PC,BCCLD         ;LOAD OUT SILO WITH 2 SYNCS
023624  000252                              252                      ;AND THE CHARACTER 252
023626  104415  000032                      DATACLK,          32     ;GET RECEIVER ACTIVE
023632  104415  000001               72$:   DATACLK,          1      ;SHIFT BCC ONCE
023636  005200                              INC     R0               ;BUMP SHIFT COUNT
023640  004537  032706                      JSR     R5 SIMBCC        ;CALCULATE SOFTWARE BCC LSB
023644  000001                              1                        ;ONE SHIFT
023646  000000                       73$:   0                        ;DATA CHARACTER
023650  000000                       74$:   0                        ;OLD BCC
023652  103405                              BCS     75$              ;BR IF SOFT BCC LSB IS SET
023654  004737  033160                      JSR     PC,GETQI         ;GET HARDWARE RECEIVER BCC LSB
023660  103006                              BCC     76$              ;BR IF HARD BCC LSB IS CLEAR
023662  104013                              HLT     13               ;ERROR, BCC LSB IS SET
023664  000404                              BR      76$              ;CONTINUE
023666  004737  033160               75$:   JSR     PC,GETQI         ;GET HARDWARE RECEIVER BCC LSB
023672  103401                              BCS     76$              ;BR IF HARD BCC LSB IS SET
023674  104017                              HLT     17               ;ERROR, BCC LSB IS CLEAR
023676                               76$:
023676  006037  023646                      ROR     73$              ;SHIFT SOFT DATA
023702  013737  033032  023650              MOV     CALBCC,74$       ;LOAD OLD SOFT BCC
023710  022700  000010                      CMP     #10,R0           ;DONE YET?
023714  001346                              BNE     72$              ;BR IF NOT DONE
023716  104401                              SCOP1                    ;SCOPE SUBTEST (SW09=1)
023720  104400                       77$:   SCOPE                    ;SCOPE THIS TEST



                                            ;**************************** TEST 51 ****************************
                                            ;*TRANSMITTER CRC TEST
                                            ;*USING THE CRC.CCITT POLYNOMINAL, SINGLE CLOCK A BINARY
```

```
                                                      ;*COUNT PATTERN, VERIFY THE LSB OF THE TRANSMITTER BCC ON EACH SHIFT
                                                      ;;******************************************************************


                                                      ;  TEST 51
                                                      ;--------------
023722  012737  000051  001226           TSTS1:       MOV      #51,TSTNO
023730  012737  024244  001216                        MOV      #TSTS2,NEXT
                                                                                 ;R1 CONTAINS BASE DMC11 ADDRESS
023736  104412                                        MSTCLR                     ;MASTER CLEAR DMC11
023740  005061  000004                                CLR      4(R1)             ;CLEAR PORT4
023744  104414                                        ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023746  122117                                        122117                     ;PUT LINE UNIT IN BITSTUFF MODE
023750  004737  033374                                JSR      PC,CLRIO          ;DO THIS AFTER MODE IS SET
023754  005037  033612                                CLR      BITCON            ;CONSECUTIVE 1'S COUNTER INIT TO 0
023760  012711  004000                                MOV      #BIT11,(R1)       ;SET LINE UNIT LOOP
023764  005003                                        CLR      R3                .ZERO BIT COUNT
023766  005004                                        CLR      R4                ;R4 CONTAINS CHAR TO BE LOADED IN SILO
023770  005005                                        CLR      R5                ;R5 CONTAINS CHAR CURRENTLY BEING SHIFTED OUT
023772  005037  024120                                CLR      4$                ;CLEAR SOFT BCC
023776  005137  024120                                COM      4$                ;START AT -1
024002  012737  102010  033030                        MOV      #CRC.CCITT,XPOLY        ;LOAD POLYNOMINAL
024010  004737  033176                                JSR      PC,SYNLD          ;LOAD SILO WITH 2 SYNCS, SOM SET
024014  010461  000004                                MOV      R4,4(R1)          ;PORT4+CHAR
024020  104414                                        ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024022  122110                                        122110                     ;LOAD OUT DATA
024024  005204                                        INC      R4                ;INCREMENT TO NEXT CHARACTER
024026  010461  000004                                MOV      R4,4(R1)          ;PORT4+CHAR
024032  104414                                        ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024034  122110                                        122110                     ;LOAD OUT DATA
024036  005204                                        INC      R4                ;INCREMENT TO NEXT CHARACTER
024040  010461  000004                                MOV      R4,4(R1)          ;PORT4+CHAR
024044  104414                                        ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024046  122110                                        122110                     ;LOAD OUT DATA
024050  004737  032044                                JSR      PC,OCOR           ;WAIT FOR OCOR
024054  104415  000021                                DATACLK,21                 ;CLOCK DATA
024060  010537  024104                                MOV      R5,10$            ;START WITH ZERO
024064  012700  000001                                MOV      #1,R0             ;START COUNT AT 1
024070  010537  024116                   1$:          MOV      R5,3$             ;LOAD CHAR FOR SOFT CRC
024074  104415  000001                   2$:          DATACLK,1                  ;SHIFT BCC ONCE
024100  004537  033474                                JSR      R5,STFFCK         'CHECK BIT STUFFING
024104  000000                           10$:         0                          ;CHARACTER
024106  000001                                        1                          ;SHIFT COUNT
024110  004537  032706                                JSR      R5,SIMBCC         ;CALCULATE SOFT BCC
024114  000001                                        1                          ;SOFT SHIFT COUNT
024116  000000                           3$:          0                          ;SOFT CHARACTER
024120  000000                           4$:          0                          ;OLD SOFT BCC
024122  103405                                        BCS      5$                ;BR IF SOFT BCC LSB IS SET
024124  004737  033146                                JSR      PC,GETQO                ;GET HARDWARE TRANSMITTER BCC LSB
024130  103006                                        BCC      6$                ;BR IF OK (CLEARED)
024132  104020                                        HLT      20                ;ERROR, BCC LSB WAS SET
024134  000404                                        BR       6$                ;CONTINUE WITH TEST
024136  004737  033146                   5$:          JSR      PC,GETQO                ;GET HARDWARE TRANSMITTER BCC LSB
024142  103401                                        BCS      6$                ;BR IF OK (SET)
024144  104021                                        HLT      21                ;ERROR, BCC LSB WAS CLEAR

024146                                   6$:
```

```
024146  006037  024104                      ROR     10$             ;SHIFT CHAR FOR STUFF CHECK
024152  005300                              DEC     R0              ;DEC STUFF CHECK SHIFT COUNT
024154  001004                              BNE     11$             ;BR IF NOT DONE THIS CHARACTER
024156  012700  000010                      MOV     #10,R0          ;RESET BIT COUNT TO 10
024162  010537  024104                      MOV     R5,10$          ;LOAD NEXT CHAR FOR STUFF CHECK
024166                              11$:
024166  006037  024116                      ROR     3$              ;SHIFT SOFT DATA
024172  013737  033032  024120              MOV     CALBCC,4$       ;LOAD OLD SOFT BCC
024200  005203                              INC     R3              ;INCREMENT BIT COUNTER
024202  022703  000010                      CMP     #10,R3          ;DONE A FULL CHARACTER YET?
024206  001332                              BNE     2$              ;BR IF NO
024210  005003                              CLR     R3              ;RESTART BIT COUNTER
024212  005204                              INC     R4              ;INCREMENT DATA FOR SILO
024214  022704  000400                      CMP     #400,R4         ;DONE BINARY COUNT YET?
024220  003404                              BLE     9$              ;BR IF YES
024222  010461  000004                      MOV     R4,4(R1)        ;PORT4←DATA
024226  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024230  122110                              122110                  ;LOAD OUT DATA
024232  005205                      9$:     INC     R5              ;INCREMENT DATA
024234  022705  000400                      CMP     #400,R5         ;DONE BINARY PATTERN YET?
024240  001313                              BNE     1$              ;BR IF NO
024242  104400                      7$:     SCOPE                   ;SCOPE THIS TEST


                                            ;************************** TEST 52 **************************
                                            ;*RECEIVER CRC TEST
                                            ;*USING THE CRC.CCITT POLYNOMINAL, SINGLE CLOCK A BINARY
                                            ;*COUNT PATTERN, VERIFY THE LSB OF THE RECEIVER BCC ON EACH SHIFT
                                            ;:*************************************************************

                                            :  TEST 52
                                            ;--------------
024244  012737  000052  001226      TST52:  MOV     #52,TSTNO
024252  012737  024602  001216              MOV     #TST53,NEXT
                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
024260  104412                              MSTCLR                  ;MASTER CLEAR DMC11
024262  005061  000004                      CLR     4(R1)           ;CLEAR PORT4
024266  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024270  122117                              122117                  ;PUT LINE UNIT IN BITSTUFF MODE
024272  004737  033374                      JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
024276  005037  033612                      CLR     BITCON          ;CONSECUTIVE 1'S COUNTER INIT TO 0
024302  012711  004000                      MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
024306  005003                              CLR     R3              ;ZERO BIT COUNT
024310  005004                              CLR     R4              ;R4 CONTAINS CHAR TO BE LOADED IN SILO
024312  005005                              CLR     R5              ;R5 CONTAINS CHAR CURRENTLY BEING SHIFTED OUT
024314  005037  024446                      CLR     4$              ;CLEAR SOFT BCC
024320  005137  024446                      COM     4$              ;START AT -1
024324  012737  102010  033030              MOV     #CRC.CCITT,XPOLY        ;LOAD POLYNOMINAL
024332  004737  033176                      JSR     PC,SYNLD        ;LOAD SILO WITH 2 SYNCS, SOM SET
024336  010461  000004                      MOV     R4,4(R1)        ;PORT4←CHAR
024342  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024344  122110                              122110                  ;LOAD OUT DATA
024346  005204                              INC     R4              ;INCREMENT TO NEXT CHARACTER
024350  010461  000004                      MOV     R4,4(R1)        ;PORT4←CHAR
024354  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024356  122110                              122110                  ;LOAD OUT DATA
```

```
024360  005204                                   INC     R4              ;INCREMENT TO NEXT CHARACTER
024362  010461  000004                           MOV     R4,4(R1)        ;PORT4+CHAR
024366  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024370  122110                                   122110                  ;LOAD OUT DATA
024372  004737  032044                           JSR     PC,OCOR         ;WAIT FOR OCOR
024376  104415  000032                           DATACLK,32              ;CLOCK DATA
024402  010537  024432                           MOV     R5,10$          ;START WITH ZERO
024406  005237  024432                           INC     10$             ;TRANSMITTER IS ONE CHAR AHEAD
024412  012700  000010                           MOV     #10,R0          ;R0 = CHAR COUNT
024416  010537  024444                  1$:      MOV     R5,3$           ;LOAD CHAR FOR SOFT CRC
024422  104415  000001                  2$:      DATACLK,1               ;SHIFT BCC ONCE
024426  004537  033474                           JSR     R5,STFFCK       ;CHECK BIT STUFFING
024432  000000                          10$:     0                       ;CHARACTER
024434  000001                                   1                       ;SHIFT COUNT
024436  004537  032706                           JSR     R5,SIMBCC       ;CALCULATE SOFT BCC
024442  000001                                   1                       ;SOFT SHIFT COUNT
024444  000000                          3$:      0                       ;SOFT CHARACTER
024446  000000                          4$:      0                       ;OLD SOFT BCC
024450  103405                                   BCS     5$              ;BR IF SOFT BCC LSB IS SET
024452  004737  033160                           JSR     PC,GETQI                ;GET HARDWARE RECEIVER BCC LSB
024456  103006                                   BCC     6$              ;BR IF OK (CLEARED)
024460  104022                                   HLT     22              ;ERROR, BCC LSB WAS SET
024462  000404                                   BR      6$              ;CONTINUE WITH TEST
024464  004737  033160                  5$:      JSR     PC,GETQI                ;GET HARDWARE RECEIVER BCC LSB
024470  103401                                   BCS     6$              ;BR IF OK (SET)
024472  104023                                   HLT     23              ;ERROR, BCC LSB WAS CLEAR

024474                                  6$:
024474  006037  024432                           ROR     10$             ;SHIFT CHAR FOR STUFF CHECK
024500  005300                                   DEC     R0              ;DEC STUFF CHECK SHIFT COUNT
024502  001010                                   BNE     11$             ;BR IF NOT DONE THIS CHARACTER
024504  012700  000010                           MOV     #10,R0          ;RESET BIT COUNT TO 10
024510  010537  024432                           MOV     R5,10$          ;LOAD NEXT CHAR FOR STUFF CHECK
024514  005237  024432                           INC     10$             ;TRANSMITTER IS 2 CHAR AHEAD
024520  005237  024432                           INC     10$             ;
024524                                  11$:
024524  006037  024444                           ROR     3$              ;SHIFT SOFT DATA
024530  013737  033032  024446                   MOV     CALBCC,4$       ;LOAD OLD SOFT BCC
024536  005203                                   INC     R3              ;INCREMENT BIT COUNTER
024540  022703  000010                           CMP     #10,R3          ;DONE A FULL CHARACTER YET?
024544  001326                                   BNE     2$              ;BR IF NO
024546  005003                                   CLR     R3              ;RESTART BIT COUNTER
024550  005204                                   INC     R4              ;INCREMENT DATA FOR SILO
024552  022704  000400                           CMP     #400,R4         ;DONE BINARY COUNT YET?
024556  003404                                   BLE     9$              ;BR IF YES
024560  010461  000004                           MOV     R4,4(R1)        ;PORT4+DATA
024564  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024566  122110                                   122110                  ;LOAD OUT DATA
024570  005205                          9$:      INC     R5              ;INCREMENT DATA
024572  022705  000400                           CMP     #400,R5         ;DONE BINARY PATTERN YET?
024576  001307                                   BNE     1$              ;BR IF NO
024600  104400                          7$:      SCOPE                   ;SCOPE THIS TEST



                                                 ;**************************** TEST 53 ****************************
                                                 ;*TRANSMITTER BITSTUFF 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 53
                                                        ;--------------
024602  012737  000053  001226          TST53:  MOV     #53,TSTNO
024610  012737  025304  001216                  MOV     #TST54,NEXT
                                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
024616  104412                                  MSTCLR                          ;MASTER CLEAR DMC11
024620  005061  000004                          CLR     4(R1)                   ;CLEAR PORT4
024624  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024626  122117                                  122117                          ;PUT LINE UNIT IN BITSTUFF MODE
024630  004737  033374                          JSR     PC,CLRIO                ;DO THIS AFTER MODE IS SET
024634  005037  033612                          CLR     BITCON                  ;CONSECUTIVE 1'S COUNTER INIT TO 0

                                                        ;LOAD OUT DATA SILO

024640  012711  004000                          MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
024644  012704  033614                          MOV     #MESDAT,R4              ;LOAD POINTER TO DATA
024650  005037  024760                          CLR     10$                     ;CLEAR SOFT BCC
024654  005137  024760                          COM     10$                     ;START AT -1
024660  012700  000004                          MOV     #4,R0                   ;LOAD CHARACTER COUNT
024664  004737  033176                          JSR     PC,SYNLD                ;LOAD 2 FLAG CHARACTERS IN OUT SILO
024670  004737  032176                          JSR     PC,OUTRDY               ;WAIT FOR OUTRDY
024674  004537  033332                          JSR     R5,MESLD                ;LOAD SILO WITH 4 CHAR MESS
024700  033614                                  MESDAT                          ;ADDRESS OF MESSAGE
024702  000004                                  4                               ;NUMBER OF CHARACTERS
024704  004737  033306                          JSR     PC,EOM                  ;LOAD GARBAGE CHARACTER, WITH EOM SET
024710  004737  033306                          JSR     PC,EOM
024714  004737  032044                          JSR     PC,OCOR                 ;WAIT FOR OCOR
024720  005003                                  CLR     R3                      ;CLEAR BIT COUNTER
024722  104415  000022                          DATACLK,22                      ;CLOCK DATA
024726  112405                          12$:    MOVB    (R4)+,R5                ;LOAD R5 WITH CHAR
024730  010502                                  MOV     R5,R2                   ;LOAD R2 WITH CHAR

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

024732  010537  025026                          MOV     R5,71$                  ;LOAD FOR STUFF CHECK
024736  012737  102010  033030                  MOV     #CRC.CCITT,XPOLY                ;LOAD POLYNOMIAL
024744  010537  024756                          MOV     R5,67$                  ;LOAD SOFT CHAR FOR BCC
024750  004537  032706                          JSR     R5,SIMBCC               ;CALCULATE SOFT BCC
024754  000010                                  10                              ;SHIFT COUNT
024756  000000                          67$:    0                               ;CHARACTER
024760  000000                          10$:    0                               ;OLD BCC
024762  013737  033032  024760                  MOV     CALBCC,10$              ;LOAD SOFT BCC FOR NEXT SHIFT
024770  104415  000001                  64$:    DATACLK,        1               ;SHIFT DATA IN TO BIT WINDOW
024774  106002                                  RORB    R2                      ;SHIFT SOFT DATA
024776  103005                                  BCC     65$                     ;BR IF A SPACE
025000  004737  032012                          JSR     PC,GETSI                ;LOOK AT BIT WINDOW
025004  103406                                  BCS     66$                     ;BR IF OK (MARK)
025006  104006                                  HLT     6                       ;ERROR, BIT WINDOW WAS A SPACE
025010  000404                                  BR      66$                     ;CONTINUE
```

```
025012  004737  032012                      65$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
025016  103001                                      BCC     66$             ;BR IF OK (SPACE)
025020  104006                                      HLT     6               ;ERROR, BIT WINDOW WAS A MARK
025022                                      66$:
025022  004537  033474                              JSR     R5,STFFCK
025026  000000                              71$:    0
025030  000001                                      1
025032  110237  025026                              MOVB    R2,71$          ;SHIFT FOR NEXT STUFF CHECK
025036  005203                                      INC     R3              ;BUMP BIT COUNTER
025040  022703  000010                              CMP     #10,R3          ;DONE FULL 8 BITS YET
025044  001351                                      BNE     64$             ;BR IF NO
025046  005003                                      CLR     R3              ;CLEAR BIT COUNTER
025050  005300                                      DEC     R0              ;DEC CHARACTER COUNT
025052  001325                                      BNE     12$             ;BR IF NOT DONE YET

                                                    ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

025054  005137  033032                              COM     CALBCC          ;ADJUST BCC FOR SDLC
025060  013700  033032                              MOV     CALBCC,R0       ;PUT BCC IN R0
025064  010037  025126                              MOV     R0,72$          ;LOAD BCC FOR STUFF CHECK
025070  104415  000001                      68$:    DATACLK,1               ;SHIFT HARDWARE BCC
025074  006000                                      ROR     R0              ;SHIFT SOFT BCC
025076  103005                                      BCC     69$             ;BR IF CARRY CLEAR
025100  004737  032012                              JSR     PC,GETSI        ;LOOK AT BIT WINDOW
025104  103406                                      BCS     70$             ;BR IF OK (MARK)
025106  104014                                      HLT     14              ;ERROR, CRC WRONG (SPACE)
025110  000404                                      BR      70$             ;CONTINUE
025112  004737  032012                      69$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
025116  103001                                      BCC     70$             ;BR IF OK (SPACE)
025120  104014                                      HLT     14              ;ERROR, CRC WRONG (MARK)
025122                                      70$:
025122  004537  033474                              JSR     R5,STFFCK       ;CHECK BCC CHAR FOR ZERO STUFFS
025126  000000                              72$:    0                       ;CHARACTER
025130  000001                                      1                       ;SHIFT COUNT
025132  010037  025126                              MOV     R0,72$          ;SHIFT SOFTBCC ONCE
025136  005203                                      INC     R3              ;BUMP BIT COUNTER
025140  022703  000020                              CMP     #20,R3          ;FINISHED BCC YET?
025144  001351                                      BNE     68$             ;BR IF NO
025146  005003                                      CLR     R3              ;CLEAR BIT COUNTER

                                                    ;CHECK FOR FLAG TO FOLLOW BCC

025150  012737  000176  001252                      MOV     #↑B<01111110>,TEMP3 ;PUT FLAG CHARACTER IN TEMP3
025156  104415  000001                      73$:    DATACLK,        1       ;CLOCK FLAG ONCE
025162  106037  001252                              RORB    TEMP3           ;SHIFT SOFT FLAG
025166  103405                                      BCS     74$             ;BR IF BIT IS MARK
025170  004737  032012                              JSR     PC,GETSI        ;LOOK AT BIT WINDOW
025174  103006                                      BCC     75$             ;BR IF OK
025176  104026                                      HLT     26              ;ERROR IN FLAG CHAR
025200  000404                                      BR      75$
025202  004737  032012                      74$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
025206  103401                                      BCS     75$             ;BR IF OK
025210  104026                                      HLT     26              ;ERROR IN FLAG CHAR
025212  005203                              75$:    INC     R3              ;INC BIT COUNT
025214  022703  000010                              CMP     #10,R3          ;FLAG DONE YET?
025220  001356                                      BNE     73$             ;BR IF NO
```

```
025222  005003                                 CLR     R3                  ;CLEAR BIT COUNT

                                               ;CHECK TO SEE IF TRANSMITTER IS MARKING

025224  104415  000001                2$:      DATACLK,     1              ;CLOCK TRANSMITTER
025230  004737  032012                         JSR     PC,GETSI            ;LOOK AT WINDOW
025234  103401                                 BCS     3$                  ;IT SHOULD BE MARKING
025236  104024                                 HLT     24                  ;ERROR, BIT WAS A SPACE
025240  005203                        3$:      INC     R3                  ;BUMP BIT COUNTER
025242  022703  000007                         CMP     #7,R3               ;DONE YET
025246  001366                                 BNE     2$                  ;BR IF NO
025250  104415  000010                         DATACLK,        10          ;GIVE ENOUGH TICKS TO CLEAR OUT ACTIVE
025254  005003                                 CLR     R3                  ;CLEAR BIT COUNTER
025256  104415  000001                4$:      DATACLK,     1              ;SHIFT OUT NEXT BIT
025262  004737  032012                         JSR     PC,GETSI            ;LOOK AT BIT WINDOW
025266  103401                                 BCS     .+4                 ;BR IF IT IS A MARK
025270  104024                                 HLT     24                  ;ERROR, TRANSMITTER IS NOT MARKING
025272  005203                                 INC     R3                  ;INC BIT COUNT
025274  022703  000020                         CMP     #20,R3              ;DONE YET?
025300  001366                                 BNE     4$                  ;BR IF NO
025302  104400                        5$:      SCOPE                       ;SCOPE THIS TEST



                                               ;*************************** TEST 54 ***************************
                                               ;*RECEIVER BITSTUFF 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 54
                                               ;---------------
025304  012737  000054  001226        TST54:   MOV     #54,TSTNO
025312  012737  025526  001216                 MOV     #TST55,NEXT
                                                                           ;R1 CONTAINS BASE DMC11 ADDRESS
025320  104412                                 MSTCLR                      ;MASTER CLEAR DMC11
025322  005061  000004                         CLR     4(R1)               ;CLEAR PORT4
025326  104414                                 ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025330  122117                                 122117                      ;PUT LINE UNIT IN BITSTUFF MODE
025332  004737  033374                         JSR     PC,CLRIO            ;DO THIS AFTER MODE IS SET
025336  012711  004000                         MOV     #BIT11,(R1)         ;SET LINE UNIT LOOP
025342  012702  033614                         MOV     #MESDAT,R2          ;LOAD POINTER TO DATA
025346  012700  000004                         MOV     #4,R0               ;LOAD CHARACTER COUNT
025352  004737  033176                         JSR     PC,SYNLD            ;LOAD 2 FLAG CHARACTERS IN OUT SILO
025356  004737  032176                         JSR     PC,OUTRDY           ;WAIT FOR OUTRDY
025362  004537  033332                         JSR     R5,MESLD            ;LOAD SILO WITH 4 CHAR MESS
025366  033614                                 MESDAT                      ;ADDRESS OF MESSAGE
025370  000004                                 4                           ;NUMBER OF CHARACTERS
025372  004737  033306                         JSR     PC,EOM              ;LOAD GARBAGE CHARACTER, WITH EOM SET
025376  004737  033306                         JSR     PC,EOM
025402  004737  032044                         JSR     PC,OCOR             ;WAIT FOR OCOR
025406  104415  000115                         DATACLK,115                 ;CLOCK DATA
025412  004737  032652                3$:      JSR     PC,INRDY            ;WAIT FOR INRDY
025416  104414                                 ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025420  021204                                 021204                      ;GET IN DATA
025422  016104  000004                         MOV     4(R1),R4            ;PUT "FOUND" IN R4
```

```
025426  112205                                  MOVB    (R2)+,R5        ;PUT "EXPECTED" IN R5
025430  120504                                  CMPB    R5,R4           ;COMPARE RECEIVED DATA
025432  001401                                  BEQ     1$              ;BR IF OK
025434  104010                                  HLT     10              ;DATA ERROR
025436  005300                          1$:     DEC     R0              ;DEC CHARACTER COUNT
025440  001364                                  BNE     3$              ;BR IF NOT DONE YET

                                        ;CHECK TO SEE THAT IN BCC MATCH IS SET

025442  004737  032652                          JSR     PC,INRDY        ;WAIT FOR INRDY
025446  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025450  021204                                  021204                  ;GET FIRST HALF OF CRC
025452  116137  000004  001252                  MOVB    4(R1),TEMP3     ;PUT IN TEMP3
025460  042737  177400  001252                  BIC     #177400,TEMP3   ;CLEAR HI BYTE
025466  004737  032652                          JSR     PC,INRDY        ;WAIT FOR INRDY
025472  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025474  021244                                  021244
025476  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
025502  042704  000374                          BIC     #374,R4         ;CLEAR UNWANTED BITS
025506  012705  000003                          MOV     #3,R5           ;PUT "EXPECTED" IN R5
025512  120504                                  CMPB    R5,R4           ;ARE IN BCC MATCH AND BLOCK END SET?
025514  001401                                  BEQ     25$
025516  104042                                  HLT     42              ;IN BCC MATCH ERROR
025520                                  25$:
025520  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025522  021204                                  021204                  ;GET LAST HALF
025524  104400                          2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*************************** TEST 55 ***************************
                                        ;*BITSTUFF EOM FUNCTION TEST
                                        ;*THIS TEST LOADS OUT SILO WITH: 2 FLAGS,4 CHAR MESSAGE,EOM
                                        ;*4 CHARACTER MESS,EOM. THE DATA STREAM IS CHECKED TO BE
                                        ;*4 CHAR,BCC,FLAG,4 CHAR,BCC,FLAG,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 55
                                        ;--------------
025526  012737  000055  001226  TST55:  MOV     #55,TSTNO
025534  012737  027126  001216          MOV     #TST56,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
025542  104412                          MSTCLR                  ;MASTER CLEAR DMC11
025544  005061  000004                  CLR     4(R1)           ;CLEAR PORT4
025550  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025552  122117                          122117                  ;PUT LINE UNIT IN BITSTUFF MODE
025554  004737  033374                  JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
025560  005037  033612                  CLR     BITCON          ;CONSECUTIVE 1'S COUNTER INIT TO 0

                                        ;LOAD OUT DATA SILO

025564  012711  004000                  MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
025570  012704  033614                  MOV     #MESDAT,R4      ;LOAD POINTER TO DATA
```

```
025574  005037  025724                       CLR     10$             ;CLEAR SOFT BCC
025600  005137  025724                       COM     10$             ;START AT -1
025604  012700  000004                       MOV     #4,R0           ;LOAD CHARACTER COUNT
025610  004737  033176                       JSR     PC,SYNLD        ;LOAD 2 FLAG CHARACTERS IN OUT SILO
025614  004737  032176                       JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
025620  004537  033332                       JSR     R5,MESLD        ;LOAD SILO WITH 4 CHAR MESS
025624  033614                               MESDAT                  ;ADDRESS OF MESSAGE
025626  000004                               4                       ;NUMBER OF CHARACTERS
025630  004737  033306                       JSR     PC,EOM          ;LOAD GARBAGE CHARACTER, WITH EOM SET
025634  004737  033306                       JSR     PC,EOM
025640  004537  033332                       JSR     R5,MESLD        ;LOAD FOUR MORE CHARACTERS
025644  033614                               MESDAT                  ;ADDRESS OF MESSAGE
025646  000004                               4                       ;NUMBER OF CHACTERS
025650  004737  033306                       JSR     PC,EOM          ;SET EOM
025654  004737  033306                       JSR     PC,EOM          ;SET EOM
025660  004737  032044                       JSR     PC,OCOR         ;WAIT FOR OCOR
025664  005003                               CLR     R3              ;CLEAR BIT COUNTER
025666  104415  000022                       DATACLK,22              ;CLOCK DATA
025672  112405                       12$:    MOVB    (R4)+,R5        ;LOAD R5 WITH CHAR
025674  010502                               MOV     R5,R2           ;LOAD R2 WITH CHAR


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

025676  010537  025772                       MOV     R5,71$          ;LOAD FOR STUFF CHECK
025702  012737  102010  033030               MOV     #CRC.CCITT,XPOLY        ;LOAD POLYNOMIAL
025710  010537  025722                       MOV     R5,67$          ;LOAD SOFT CHAR FOR BCC
025714  004537  032706                       JSR     R5,SIMBCC       ;CALCULATE SOFT BCC
025720  000010                               10                      ;SHIFT COUNT
025722  000000                       67$:    0                       ;CHARACTER
025724  000000                       10$:    0                       ;OLD BCC
025726  013737  033032  025724               MOV     CALBCC,10$      ;LOAD SOFT BCC FOR NEXT SHIFT
025734  104415  000001               64$:    DATACLK,        1       ;SHIFT DATA IN TO BIT WINDOW
025740  106002                               RORB    R2              ;SHIFT SOFT DATA
025742  103005                               BCC     65$             ;BR IF A SPACE
025744  004737  032012                       JSR     PC,GETSI        ;LOOK AT BIT WINDOW
025750  103406                               BCS     66$             ;BR IF OK (MARK)
025752  104006                               HLT     6               ;ERROR, BIT WINDOW WAS A SPACE
025754  000404                               BR      66$             ;CONTINUE
025756  004737  032012               65$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
025762  103001                               BCC     66$             ;BR IF OK (SPACE)
025764  104006                               HLT     6               ;ERROR, BIT WINDOW WAS A MARK
025766                               66$:
025766  004537  033474                       JSR     R5,STFFCK
025772  000000                       71$:    0
025774  000001                               1
025776  110237  025772                       MOVB    R2,71$          ;SHIFT FOR NEXT STUFF CHECK
026002  005203                               INC     R3              ;BUMP BIT COUNTER
026004  022703  000010                       CMP     #10,R3          ;DONE FULL 8 BITS YET
026010  001351                               BNE     64$             ;BR IF NO
026012  005003                               CLR     R3              ;CLEAR BIT COUNTER
026014  005300                               DEC     R0              ;DEC CHARACTER COUNT
026016  001325                               BNE     12$             ;BR IF NOT DONE YET

                                             ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

```

```
026020  005137  033032                    COM     CALBCC          ;ADJUST BCC FOR SDLC
026024  013700  033032                    MOV     CALBCC,R0       ;PUT BCC IN R0
026030  010037  026072                    MOV     R0,72$          ;LOAD BCC FOR STUFF CHECK
026034  104415  000001            68$:    DATACLK,1               ;SHIFT HARDWARE BCC
026040  006000                            ROR     R0              ;SHIFT SOFT BCC
026042  103005                            BCC     69$             ;BR IF CARRY CLEAR
026044  004737  032012                    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
026050  103406                            BCS     70$             ;BR IF OK (MARK)
026052  104014                            HLT     14              ;ERROR, CRC WRONG (SPACE)
026054  000404                            BR      70$             ;CONTINUE
026056  004737  032012            69$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
026062  103001                            BCC     70$             ;BR IF OK (SPACE)
026064  104014                            HLT     14              ;ERROR, CRC WRONG (MARK)
026066                            70$:
026066  004537  033474                    JSR     R5,STFFCK       ;CHECK BCC CHAR FOR ZERO STUFFS
026072  000000                    72$:    0                       ;CHARACTER
026074  000001                            1                       ;SHIFT COUNT
026076  010037  026072                    MOV     R0,72$          ;SHIFT SOFTBCC ONCE
026102  005203                            INC     R3              ;BUMP BIT COUNTER
026104  022703  000020                    CMP     #20,R3          ;FINISHED BCC YET?
026110  001351                            BNE     68$             ;BR IF NO
026112  005003                            CLR     R3              ;CLEAR BIT COUNTER

                                          ;CHECK FOR FLAG TO FOLLOW BCC

026114  012737  000176  001252            MOV     #↑B<01111110>,TEMP3 ;PUT FLAG CHARACTER IN TEMP3
026122  104415  000001            73$:    DATACLK,        1       ;CLOCK FLAG ONCE
026126  106037  001252                    RORB    TEMP3           ;SHIFT SOFT FLAG
026132  103405                            BCS     74$             ;BR IF BIT IS MARK
026134  004737  032012                    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
026140  103006                            BCC     75$             ;BR IF OK
026142  104026                            HLT     26              ;ERROR IN FLAG CHAR
026144  000404                            BR      75$
026146  004737  032012            74$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
026152  103401                            BCS     75$             ;BR IF OK
026154  104026                            HLT     26              ;ERROR IN FLAG CHAR
026156  005203                    75$:    INC     R3              ;INC BIT COUNT
026160  022703  000010                    CMP     #10,R3          ;FLAG DONE YET?
026164  001356                            BNE     73$             ;BR IF NO
026166  005003                            CLR     R3              ;CLEAR BIT COUNT
026170  012700  000004                    MOV     #4,R0           ;RESET CHARACTER COUNTER
026174  012704  033614                    MOV     #MESDAT,R4      ;LOAD MESSAGE POINTER
026200  005037  026242                    CLR     11$             ;CLR SOFT BCC
026204  005137  026242                    COM     11$             ;ADJUST TO -1 FOR SDLC
026210  112405                    13$:    MOVB    (R4)+,R5        ;LOAD CHAR IN R5
026212  010502                            MOV     R5,R2           ;LOAD CHAR IN R2

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

026214  010537  026310                    MOV     R5,83$          ;LOAD FOR STUFF CHECK
026220  012737  102010  033030            MOV     #CRC.CCITT,XPOLY        ;LOAD POLYNOMIAL
026226  010537  026240                    MOV     R5,79$          ;LOAD SOFT CHAR FOR BCC
026232  004537  032706                    JSR     R5,SIMBCC       ;CALCULATE SOFT BCC
026236  000010                            10                      ;SHIFT COUNT
026240  000000                    79$:    0                       ;CHARACTER
026242  000000                    11$:    0                       ;OLD BCC
```

```
026244  013737  033032  026242                MOV     CALBCC,11$        ;LOAD SOFT BCC FOR NEXT SHIFT
026252  104415  000001                76$:    DATACLK,         1        ;SHIFT DATA IN TO BIT WINDOW
026256  106002                                RORB    R2                ;SHIFT SOFT DATA
026260  103005                                BCC     77$               ;BR IF A SPACE
026262  004737  032012                        JSR     PC,GETSI          ;LOOK AT BIT WINDOW
026266  103406                                BCS     78$               ;BR IF OK (MARK)
026270  104006                                HLT     6                 ;ERROR, BIT WINDOW WAS A SPACE
026272  000404                                BR      78$               ;CONTINUE
026274  004737  032012                77$:    JSR     PC,GETSI          ;LOOK AT BIT WINDOW
026300  103001                                BCC     78$               ;BR IF OK (SPACE)
026302  104006                                HLT     6                 ;ERROR, BIT WINDOW WAS A MARK
026304                                78$:
026304  004537  033474                        JSR     R5,STFFCK
026310  000000                        83$:    0
026312  000001                                1
026314  110237  026310                        MOVB    R2,83$            ;SHIFT FOR NEXT STUFF CHECK
026320  005203                                INC     R3                ;BUMP BIT COUNTER
026322  022703  000010                        CMP     #10,R3            ;DONE FULL 8 BITS YET
026326  001351                                BNE     76$               ;BR IF NO
026330  005003                                CLR     R3                ;CLEAR BIT COUNTER
026332  005300                                DEC     R0                ;DEC CHARACTER COUNT
026334  001325                                BNE     13$               ;BR IF NOT DONE YET

                                      ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

026336  005137  033032                        COM     CALBCC            ;ADJUST BCC FOR SDLC
026342  013700  033032                        MOV     CALBCC,R0         ;PUT BCC IN R0
026346  010037  026410                        MOV     R0,84$            ;LOAD BCC FOR STUFF CHECK
026352  104415  000001                80$:    DATACLK,1                 ;SHIFT HARDWARE BCC
026356  006000                                ROR     R0                ;SHIFT SOFT BCC
026360  103005                                BCC     81$               ;BR IF CARRY CLEAR
026362  004737  032012                        JSR     PC,GETSI          ;LOOK AT BIT WINDOW
026366  103406                                BCS     82$               ;BR IF OK (MARK)
026370  104014                                HLT     14                ;ERROR, CRC WRONG (SPACE)
026372  000404                                BR      82$               ;CONTINUE
026374  004737  032012                81$:    JSR     PC,GETSI          ;LOOK AT BIT WINDOW
026400  103001                                BCC     82$               ;BR IF OK (SPACE)
026402  104014                                HLT     14                ;ERROR, CRC WRONG (MARK)
026404                                82$:
026404  004537  033474                        JSR     R5,STFFCK         ;CHECK BCC CHAR FOR ZERO STUFFS
026410  000000                        84$:    0                         ;CHARACTER
026412  000001                                1                         ;SHIFT COUNT
026414  010037  026410                        MOV     R0,84$            ;SHIFT SOFTBCC ONCE
026420  005203                                INC     R3                ;BUMP BIT COUNTER
026422  022703  000020                        CMP     #20,R3            ;FINISHED BCC YET?
026426  001351                                BNE     80$               ;BR IF NO
026430  005003                                CLR     R3                ;CLEAR BIT COUNTER

                                      ;CHECK FOR FLAG TO FOLLOW BCC

026432  012737  000176  001252                MOV     #↑B(01111110),TEMP3 ;PUT FLAG CHARACTER IN TEMP3
026440  104415  000001                85$:    DATACLK,         1        ;CLOCK FLAG ONCE
026444  106037  001252                        RORB    TEMP3             ;SHIFT SOFT FLAG
026450  103405                                BCS     86$               ;BR IF BIT IS MARK
026452  004737  032012                        JSR     PC,GETSI          ;LOOK AT BIT WINDOW
026456  103006                                BCC     97$               ;BR IF OK
```

```
 026460  104026                                   HLT     26              ;ERROR IN FLAG CHAR
 026462  000404                                   BR      87$
 026464  004737  032012                   86$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
 026470  103401                                   BCS     87$             ;BR IF OK
 026472  104026                                   HLT     26              ;ERROR IN FLAG CHAR
 026474  005203                           87$:    INC     R3              ;INC BIT COUNT
 026476  022703  000010                           CMP     #10,R3          ;FLAG DONE YET?
 026502  001356                                   BNE     85$             ;BR IF NO
 026504  005003                                   CLR     R3              ;CLEAR BIT COUNT

                                                  ;CHECK TO SEE IF TRANSMITTER IS MARKING

 026506  104415  000001                   2$:     DATACLK,       1        ;CLOCK TRANSMITTER
 026512  004737  032012                           JSR    PC,GETSI         ;LOOK AT WINDOW
 026516  103401                                   BCS     3$              ;IT SHOULD BE MARKING
 026520  104024                                   HLT     24              ;ERROR, BIT WAS A SPACE
 026522  005203                           3$:     INC     R3              ;BUMP BIT COUNTER
 026524  022703  000007                           CMP     #7,R3           ;DONE YET
 026530  001366                                   BNE     2$              ;BR IF NO
 026532  104415  000010                           DATACLK,        10      ;GIVE ENOUGH TICKS TO CLEAR OUT ACTIVE
 026536  005003                                   CLR    R3               ;CLEAR BIT COUNTER
 026540  104415  000001                   4$:     DATACLK,       1        ;SHIFT OUT NEXT BIT
 026544  004737  032012                           JSR    PC,GETSI         ;LOOK AT BIT WINDOW
 026550  103401                                   BCS    .+4              ;BR IF IT IS A MARK
 026552  104024                                   HLT     24              ;ERROR, TRANSMITTER IS NOT MARKING
 026554  005203                                   INC     R3              ;INC BIT COUNT
 026556  022703  000020                           CMP     #20,R3          ;DONE YET?
 026562  001366                                   BNE     4$              ;BR IF NO

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

 026564  104415  000001                           DATACLK,       1        ;GET LAST BIT IN RECEIVER
 026570  012703  000004                           MOV     #4,R3           ;R3=CHARACTER COUNT
 026574  012702  033614                           MOV     #MESDAT,R2      ;LOAD MESSAGE POINTER IN R2
 026600  004737  032652                   40$:    JSR     PC,INRDY        ;WAIT FOR INRDY
 026604  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 026606  021204                                   021204
 026610  016104  000004                           MOV     4(R1),R4        ;PUT "FOUND" IN R4
 026614  112205                                   MOVB    (R2)+,R5        ;PUT "EXPECTED" IN R5
 026616  120504                                   CMPB    R5,R4           ;IS RECEIVED DATA CORRECT?
 026620  001401                                   BEQ     41$             ;BR IF YES
 026622  104010                                   HLT     10              ;RECEIVE DATA ERROR
 026624  005303                           41$:    DEC     R3              ;DEC CHARACTER COUNT
 026626  001364                                   BNE     40$             ;BR IF NOT DONE YET

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

 026630  004737  032652                           JSR    PC,INRDY         ;WAIT FOR INRDY
 026634  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 026636  021204                                   021204                  ;GET FIRST HALF OF CRC
 026640  116137  000004  001252                   MOVB    4(R1),TEMP3     ;PUT IN TEMP3
 026646  042737  177400  001252                   BIC     #177400,TEMP3   ;CLEAR HI BYTE
 026654  004737  032652                           JSR     PC,INRDY        ;WAIT FOR INRDY
 026660  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
026662  021244                                   021244
026664  016104  000004                           MOV     4(R1),R4         ;PUT "FOUND" IN R4
026670  042704  000374                           BIC     #374,R4          ;CLEAR UNWANTED BITS
026674  012705  000003                           MOV     #3,R5            ;PUT "EXPECTED" IN R5
026700  120504                                   CMPB    R5,R4            ;ARE IN BCC MATCH AND BLOCK END SET?
026702  001401                                   BEQ     50$
026704  104042                                   HLT     42               ;IN BCC MATCH ERROR
026706                                   50$:
026706  104414                                   ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026710  021204                                   021204                   ;GET LAST HALF
026712  116137  000004  001251                   MOVB    4(R1),TEMP2+1    ;PUT IN TEMP2
026720  042737  000377  001250                   BIC     #377,TEMP2       ;CLEAR LO BYTE
026726  053737  001250  001252                   BIS     TEMP2,TEMP3      ;16 BIT BCC NOW IN TEMP3
026734  023737  033032  001252                   CMP     CALBCC,TEMP3     ;IS IT CORRECT?
026742  001401                                   BEQ     42$              ;BR IF OK
026744  104027                                   HLT     27


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

026746  012703  000004                   42$:    MOV     #4,R3            ;R3=CHARACTER COUNT
026752  012702  033614                           MOV     #MESDAT,R2       ;LOAD MESSAGE POINTER IN R2
026756  004737  032652                   43$:    JSR     PC,INRDY         ;WAIT FOR INRDY
026762  104414                                   ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026764  021204                                   021204
026766  016104  000004                           MOV     4(R1),R4         ;PUT "FOUND" IN R4
026772  112205                                   MOVB    (R2)+,R5         ;PUT "EXPECTED" IN R5
026774  120504                                   CMPB    R5,R4            ;IS RECEIVED DATA CORRECT?
026776  001401                                   BEQ     44$              ;BR IF YES
027000  104010                                   HLT     10               ;RECEIVE DATA ERROR
027002  005303                           44$:    DEC     R3               ;DEC CHARACTER COUNT
027004  001364                                   BNE     43$              ;BR IF NOT DONE YET

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

027006  004737  032652                           JSR     PC,INRDY         ;WAIT FOR INRDY
027012  104414                                   ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027014  021204                                   021204                   ;GET FIRST HALF OF CRC
027016  116137  000004  001252                   MOVB    4(R1),TEMP3      ;PUT IN TEMP3
027024  042737  177400  001252                   BIC     #177400,TEMP3    ;CLEAR HI BYTE
027032  004737  032652                           JSR     PC,INRDY         ;WAIT FOR INRDY
027036  104414                                   ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027040  021244                                   021244
027042  016104  000004                           MOV     4(R1),R4         ;PUT "FOUND" IN R4
027046  042704  000374                           BIC     #374,R4          ;CLEAR UNWANTED BITS
027052  012705  000003                           MOV     #3,R5            ;PUT "EXPECTED" IN R5
027056  120504                                   CMPB    R5,R4            ;ARE IN BCC MATCH AND BLOCK END SET?
027060  001401                                   BEQ     51$
027062  104042                                   HLT     42               ;IN BCC MATCH ERROR
027064                                   51$:
027064  104414                                   ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027066  021204                                   021204                   ;GET LAST HALF
027070  116137  000004  001251                   MOVB    4(R1),TEMP2+1    ;PUT IN TEMP2
027076  042737  000377  001250                   BIC     #377,TEMP2       ;CLEAR LO BYTE
027104  053737  001250  001252                   BIS     TEMP2,TEMP3      ;16 BIT BCC NOW IN TEMP3
```

```
027112  023737  033032  001252                  CMP     CALBCC,TEMP3        ;IS IT CORRECT?
027120  001401                                  BEQ     5$                  ;BR IF OK
027122  104027                                  HLT     27
027124  104400                          5$:     SCOPE                       ;SCOPE THIS TEST


                                                ;************************** TEST 56 ***************************
                                                ;*BITSTUFF EOM FUNCTION TEST
                                                ;*THIS TEST LOADS OUT SILO WITH: 2 FLAGS,4 CHAR MESSAGE,EOM
                                                ;*SOM,4 CHAR MESS,EOM. THE DATA STREAM IS CHECKED TO BE
                                                ;*4 CHAR,BCC,FLAG,4 CHAR,BCC,FLAG,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 56
                                                ;---------------
027126  012737  000056  001226          TST56:  MOV     #56,TSTNO
027134  012737  030606  001216                  MOV     #TST57,NEXT
                                                                            ;R1 CONTAINS BASE DMC11 ADDRESS
027142  104412                                  MSTCLR                      ;MASTER CLEAR DMC11
027144  005061  000004                          CLR     4(R1)               ;CLEAR PORT4
027150  104414                                  ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027152  122117                                  122117                      ;PUT LINE UNIT IN BITSTUFF MODE
027154  004737  033374                          JSR     PC,CLRIO            ;DO THIS AFTER MODE IS SET
027160  005037  033612                          CLR     BITCON              ;CONSECUTIVE 1'S COUNTER INIT TO 0

                                                ;LOAD OUT DATA SILO

027164  012711  004000                          MOV     #BIT11,(R1)         ;SET LINE UNIT LOOP
027170  012704  033614                          MOV     #MESDAT,R4          ;LOAD POINTER TO DATA
027174  005037  027330                          CLR     10$                 ;CLEAR SOFT BCC
027200  005137  027330                          COM     10$                 ;START AT -1
027204  012700  000004                          MOV     #4,R0               ;LOAD CHARACTER COUNT
027210  004737  033176                          JSR     PC,SYNLD            ;LOAD 2 FLAG CHARACTERS IN OUT SILO
027214  004737  032176                          JSR     PC,OUTRDY           ;WAIT FOR OUTRDY
027220  004537  033332                          JSR     R5,MESLD            ;LOAD SILO WITH 4 CHAR MESS
027224  033614                                  MESDAT                      ;ADDRESS OF MESSAGE
027226  000004                                  4                           ;NUMBER OF CHARACTERS
027230  004737  033306                          JSR     PC,EOM              ;LOAD GARBAGE CHARACTER, WITH EOM SET
027234  004737  033306                          JSR     PC,EOM
027240  004737  033256                          JSR     PC,SOM              ;LOAD GARBAGE CHAR WITH SOM SET
027244  004537  033332                          JSR     R5,MESLD            ;LOAD FOUR MORE CHARACTERS
027250  033614                                  MESDAT                      ;ADDRESS OF MESSAGE
027252  000004                                  4                           ;NUMBER OF CHACTERS
027254  004737  033306                          JSR     PC,EOM              ;SET EOM
027260  004737  033306                          JSR     PC,EOM              ;SET EOM
027264  004737  032044                          JSR     PC,OCOR             ;WAIT FOR OCOR
027270  005003                                  CLR     R3                  ;CLEAR BIT COUNTER
027272  104415  000022                          DATACLK,22                  ;CLOCK DATA
027276  112405                          12$:    MOVB    (R4)+,R5            ;LOAD R5 WITH CHAR
027300  010502                                  MOV     R5,R2               ;LOAD R2 WITH CHAR

```

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

027302  010537  027376                           MOV     R5,71$              ;LOAD FOR STUFF CHECK
027306  012737  102010  033030                   MOV     #CRC.CCITT,XPOLY'         ;LOAD POLYNOMIAL
027314  010537  027326                           MOV     R5,67$              ;LOAD SOFT CHAR FOR BCC
027320  004537  032706                           JSR     R5,SIMBCC           ;CALCULATE SOFT BCC
027324  000010                                   10                          ;SHIFT COUNT
027326  000000                           67$:    0                           ;CHARACTER
027330  000000                           10$:    0                           ;OLD BCC
027332  013737  033032  027330                   MOV     CALBCC,10$          ;LOAD SOFT BCC FOR NEXT SHIFT
027340  104415  000001                   64$:    DATACLK,        1           ;SHIFT DATA IN TO BIT WINDOW
027344  106002                                   RORB    R2                  ;SHIFT SOFT DATA
027346  103005                                   BCC     65$                 ;BR IF A SPACE
027350  004737  032012                           JSR     PC,GETSI            ;LOOK AT BIT WINDOW
027354  103406                                   BCS     66$                 ;BR IF OK (MARK)
027356  104006                                   HLT     6                   ;ERROR, BIT WINDOW WAS A SPACE
027360  000404                                   BR      66$                 ;CONTINUE
027362  004737  032012                   65$:    JSR     PC,GETSI            ;LOOK AT BIT WINDOW
027366  103001                                   BCC     66$                 ;BR IF OK (SPACE)
027370  104006                                   HLT     6                   ;ERROR, BIT WINDOW WAS A MARK
027372                                   66$:
027372  004537  033474                           JSR     R5,STFFCK
027376  000000                           71$:    0
027400  000001                                   1
027402  110237  027376                           MOVB    R2,71$              ;SHIFT FOR NEXT STUFF CHECK
027406  005203                                   INC     R3                  ;BUMP BIT COUNTER
027410  022703  000010                           CMP     #10,R3              ;DONE FULL 8 BITS YET
027414  001351                                   BNE     64$                 ;BR IF NO
027416  005003                                   CLR     R3                  ;CLEAR BIT COUNTER
027420  005300                                   DEC     R0                  ;DEC CHARACTER COUNT
027422  001325                                   BNE     12$                 ;BR IF NOT DONE YET

                                                 ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

027424  005137  033032                           COM     CALBCC              ;ADJUST BCC FOR SDLC
027430  013700  033032                           MOV     CALBCC,R0           ;PUT BCC IN R0
027434  010037  027476                           MOV     R0,72$              ;LOAD BCC FOR STUFF CHECK
027440  104415  000001                   68$:    DATACLK,1                   ;SHIFT HARDWARE BCC
027444  006000                                   ROR     R0                  ;SHIFT SOFT BCC
027446  103005                                   BCC     69$                 ;BR IF CARRY CLEAR
027450  004737  032012                           JSR     PC,GETSI            ;LOOK AT BIT WINDOW
027454  103406                                   BCS     70$                 ;BR IF OK (MARK)
027456  104014                                   HLT     14                  ;ERROR, CRC WRONG (SPACE)
027460  000404                                   BR      70$                 ;CONTINUE
027462  004737  032012                   69$:    JSR     PC,GETSI            ;LOOK AT BIT WINDOW
027466  103001                                   BCC     70$                 ;BR IF OK (SPACE)
027470  104014                                   HLT     14                  ;ERROR, CRC WRONG (MARK)
027472                                   70$:
027472  004537  033474                           JSR     R5,STFFCK           ;CHECK BCC CHAR FOR ZERO STUFFS
027476  000000                           72$:    0                           ;CHARACTER
027500  000001                                   1                           ;SHIFT COUNT
027502  010037  027476                           MOV     R0,72$              ;SHIFT SOFTBCC ONCE
027506  005203                                   INC     R3                  ;BUMP BIT COUNTER
027510  022703  000020                           CMP     #20,R3              ;FINISHED BCC YET?
027514  001351                                   BNE     68$                 ;BR IF NO
```

```
027516  005003                                     CLR     R3                  ;CLEAR BIT COUNTER

                                                   ;CHECK FOR FLAG TO FOLLOW BCC

027520  012737  000176  001252                     MOV     #↑B<01111110>,TEMP3 ;PUT FLAG CHARACTER IN TEMP3
027526  104415  000001                     73$:    DATACLK,     1              ;CLOCK FLAG ONCE
027532  106037  001252                             RORB    TEMP3               ;SHIFT SOFT FLAG
027536  103405                                     BCS     74$                 ;BR IF BIT IS MARK
027540  004737  032012                             JSR     PC,GETSI            ;LOOK AT BIT WINDOW
027544  103006                                     BCC     75$                 ;BR IF OK
027546  104026                                     HLT     26                  ;ERROR IN FLAG CHAR
027550  000404                                     BR      75$
027552  004737  032012                     74$:    JSR     PC,GETSI            ;LOOK AT BIT WINDOW
027556  103401                                     BCS     75$                 ;BR IF OK
027560  104026                                     HLT     26                  ;ERROR IN FLAG CHAR
027562  005203                             75$:    INC     R3                  ;INC BIT COUNT
027564  022703  000010                             CMP     #10,R3              ;FLAG DONE YET?
027570  001356                                     BNE     73$                 ;BR IF NO
027572  005003                                     CLR     R3                  ;CLEAR BIT COUNT

                                                   ;CHECK FOR ANOTHER FLAG CAUSED BY THE SOM

027574  012737  000176  001252                     MOV     #↑B<01111110>,TEMP3 ;PUT FLAG CHARACTER IN TEMP3
027602  104415  000001                     76$:    DATACLK,     1              ;CLOCK FLAG ONCE
027606  106037  001252                             RORB    TEMP3               ;SHIFT SOFT FLAG
027612  103405                                     BCS     77$                 ;BR IF BIT IS MARK
027614  004737  032012                             JSR     PC,GETSI            ;LOOK AT BIT WINDOW
027620  103006                                     BCC     78$                 ;BR IF OK
027622  104026                                     HLT     26                  ;ERROR IN FLAG CHAR
027624  000404                                     BR      78$
027626  004737  032012                     77$:    JSR     PC,GETSI            ;LOOK AT BIT WINDOW
027632  103401                                     BCS     78$                 ;BR IF OK
027634  104026                                     HLT     26                  ;ERROR IN FLAG CHAR
027636  005203                             78$:    INC     R3                  ;INC BIT COUNT
027640  022703  000010                             CMP     #10,R3              ;FLAG DONE YET?
027644  001356                                     BNE     76$                 ;BR IF NO
027646  005003                                     CLR     R3                  ;CLEAR BIT COUNT
027650  012700  000004                             MOV     #4,R0               ;RESET CHARACTER COUNTER
027654  012704  033614                             MOV     #MESDAT,R4          ;LOAD MESSAGE POINTER
027660  005037  027722                             CLR     11$                 ;CLR SOFT BCC
027664  005137  027722                             COM     11$                 ;ADJUST TO -1 FOR SDLC
027670  112405                             13$:    MOVB    (R4)+,R5            ;LOAD CHAR IN R5
027672  010502                                     MOV     R5,R2               ;LOAD CHAR IN R2

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

027674  010537  027770                             MOV     R5,86$              ;LOAD FOR STUFF CHECK
027700  012737  102010  033030                     MOV     #CRC.CCITT,XPOLY            ;LOAD POLYNOMIAL
027706  010537  027720                             MOV     R5,82$              ;LOAD SOFT CHAR FOR BCC
027712  004537  032706                             JSR     R5,SIMBCC           ;CALCULATE SOFT BCC
027716  000010                                     10                          ;SHIFT COUNT
027720  000000                             82$:    0                           ;CHARACTER
027722  000000                             11$:    0                           ;OLD BCC
027724  013737  033032  027722                     MOV     CALBCC,11$          ;LOAD SOFT BCC FOR NEXT SHIFT
027732  104415  000001                     79$:    DATACLK,     1              ;SHIFT DATA IN TO BIT WINDOW
027736  106002                                     RORB    R2                  ;SHIFT SOFT DATA
```

```
027740  103005                          BCC     80$             ;BR IF A SPACE
027742  004737  032012                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
027746  103406                          BCS     81$             ;BR IF OK (MARK)
027750  104006                          HLT     6               ;ERROR, BIT WINDOW WAS A SPACE
027752  000404                          BR      81$             ;CONTINUE
027754  004737  032012          80$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
027760  103001                          BCC     81$             ;BR IF OK (SPACE)
027762  104006                          HLT     6               ;ERROR, BIT WINDOW WAS A MARK
027764                          81$:
027764  004537  033474                  JSR     R5,STFFCK
027770  000000                  86$:    0
027772  000001                          1
027774  110237  027770                  MOVB    R2,86$          ;SHIFT FOR NEXT STUFF CHECK
030000  005203                          INC     R3              ;BUMP BIT COUNTER
030002  022703  000010                  CMP     #10,R3          ;DONE FULL 8 BITS YET
030006  001351                          BNE     79$             ;BR IF NO
030010  005003                          CLR     R3              ;CLEAR BIT COUNTER
030012  005300                          DEC     R0              ;DEC CHARACTER COUNT
030014  001325                          BNE     13$             ;BR IF NOT DONE YET

                                        ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

030016  005137  033032                  COM     CALBCC          ;ADJUST BCC FOR SDLC
030022  013700  033032                  MOV     CALBCC,R0       ;PUT BCC IN R0
030026  010037  030070                  MOV     R0,87$          ;LOAD BCC FOR STUFF CHECK
030032  104415  000001          93$:    DATACLK,1               ;SHIFT HARDWARE BCC
030036  006000                          ROR     R0              ;SHIFT SOFT BCC
030040  103005                          BCC     84$             ;BR IF CARRY CLEAR
030042  004737  032012                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
030046  103406                          BCS     85$             ;BR IF OK (MARK)
030050  104014                          HLT     14              ;ERROR, CRC WRONG (SPACE)
030052  000404                          BR      85$             ;CONTINUE
030054  004737  032012          84$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
030060  103001                          BCC     85$             ;BR IF OK (SPACE)
030062  104014                          HLT     14              ;ERROR, CRC WRONG (MARK)
030064                          85$:
030064  004537  033474                  JSR     R5,STFFCK       ;CHECK BCC CHAR FOR ZERO STUFFS
030070  000000                  87$:    0                       ;CHARACTER
030072  000001                          1                       ;SHIFT COUNT
030074  010037  030070                  MOV     R0,87$          ;SHIFT SOFTBCC ONCE
030100  005203                          INC     R3              ;BUMP BIT COUNTER
030102  022703  000020                  CMP     #20,R3          ;FINISHED BCC YET?
030106  001351                          BNE     83$             ;BR IF NO
030110  005003                          CLR     R3              ;CLEAR BIT COUNTER

                                        ;CHECK FOR FLAG TO FOLLOW BCC

030112  012737  000176  001252          MOV     #↑B<01111110>,TEMP3 ;PUT FLAG CHARACTER IN TEMP3
030120  104415  000001          88$:    DATACLK,        1       ;CLOCK FLAG ONCE
030124  106037  001252                  RORB    TEMP3           ;SHIFT SOFT FLAG
030130  103405                          BCS     89$             ;BR IF BIT IS MARK
030132  004737  032012                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
030136  103006                          BCC     90$             ;BR IF OK
030140  104026                          HLT     26              ;ERROR IN FLAG CHAR
030142  000404                          BR      90$
030144  004737  032012          89$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
```

```
030150  103401                            BCS     90$             ;BR IF OK
030152  104026                            HLT     26              ;ERROR IN FLAG CHAR
030154  005203                    90$:    INC     R3              ;INC BIT COUNT
030156  022703  000010                    CMP     #10,R3          ;FLAG DONE YET?
030162  001356                            BNE     88$             ;BR IF NO
030164  005003                            CLR     R3              ;CLEAR BIT COUNT

                                          ;CHECK TO SEE IF TRANSMITTER IS MARKING

030166  104415  000001            2$:     DATACLK,        1       ;CLOCK TRANSMITTER
030172  004737  032012                    JSR     PC,GETSI        ;LOOK AT WINDOW
030176  103401                            BCS     3$              ;IT SHOULD BE MARKING
030200  104024                            HLT     24              ;ERROR, BIT WAS A SPACE
030202  005203                    3$:     INC     R3              ;BUMP BIT COUNTER
030204  022703  000007                    CMP     #7,R3           ;DONE YET
030210  001366                            BNE     2$              ;BR IF NO
030212  104415  000010                    DATACLK,        10      ;GIVE ENOUGH TICKS TO CLEAR OUT ACTIVE
030216  005003                            CLR     R3              ;CLEAR BIT COUNTER
030220  104415  000001            4$:     DATACLK,        1       ;SHIFT OUT NEXT BIT
030224  004737  032012                    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
030230  103401                            BCS     .+4             ;BR IF IT IS A MARK
030232  104024                            HLT     24              ;ERROR, TRANSMITTER IS NOT MARKING
030234  005203                            INC     R3              ;INC BIT COUNT
030236  022703  000020                    CMP     #20,R3          ;DONE YET?
030242  001366                            BNE     4$              ;BR IF NO

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

030244  104415  000001                    DATACLK,        1       ;GET LAST BIT IN RECEIVER
030250  012703  000004                    MOV     #4,R3           ;R3=CHARACTER COUNT
030254  012702  033614                    MOV     #MESDAT,R2      ;LOAD MESSAGE POINTER IN R2
030260  004737  032652            40$:    JSR     PC,INRDY        ;WAIT FOR INRDY
030264  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030266  021204                            021204
030270  016104  000004                    MOV     4(R1),R4        ;PUT "FOUND" IN R4
030274  112205                            MOVB    (R2)+,R5        ;PUT "EXPECTED" IN R5
030276  120504                            CMPB    R5,R4           ;IS RECEIVED DATA CORRECT?
030300  001401                            BEQ     41$             ;BR IF YES
030302  104010                            HLT     10              ;RECEIVE DATA ERROR
030304  005303                    41$:    DEC     R3              ;DEC CHARACTER COUNT
030306  001364                            BNE     40$             ;BR IF NOT DONE YET

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

030310  004737  032652                    JSR     PC,INRDY        ;WAIT FOR INRDY
030314  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030316  021204                            021204                  ;GET FIRST HALF OF CRC
030320  116137  000004  001252            MOVB    4(R1),TEMP3     ;PUT IN TEMP3
030326  042737  177400  001252            BIC     #177400,TEMP3   ;CLEAR HI BYTE
030334  004737  032652                    JSR     PC,INRDY        ;WAIT FOR INRDY
030340  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030342  021244                            021244
030344  016104  000004                    MOV     4(R1),R4        ;PUT "FOUND" IN R4
030350  042704  000374                    BIC     #374,R4         ;CLEAR UNWANTED BITS
```

```
030354  012705  000003                   MOV     #3,R5           ;PUT "EXPECTED" IN R5
030360  120504                           CMPB    R5,R4           ;ARE IN BCC MATCH AND BLOCK END SET?
030362  001401                           BEQ     50$
030364  104042                           HLT     42              ;IN BCC MATCH ERROR
030366                           50$:
030366  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030370  021204                           021204                  ;GET LAST HALF
030372  116137  000004  001251           MOVB    4(R1),TEMP2+1   ;PUT IN TEMP2
030400  042737  000377  001250           BIC     #377,TEMP2      ;CLEAR LO BYTE
030406  053737  001250  001252           BIS     TEMP2,TEMP3     ;16 BIT BCC NOW IN TEMP3
030414  023737  033032  001252           CMP     CALBCC,TEMP3    ;IS IT CORRECT?
030422  001401                           BEQ     42$             ;BR IF OK
030424  104027                           HLT     27

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

030426  012703  000004           42$:    MOV     #4,R3           ;R3=CHARACTER COUNT
030432  012702  033614                   MOV     #MESDAT,R2      ;LOAD MESSAGE POINTER IN R2
030436  004737  032652           43$:    JSR     PC,INRDY        ;WAIT FOR INRDY
030442  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030444  021204                           021204
030446  016104  000004                   MOV     4(R1),R4        ;PUT "FOUND" IN R4
030452  112205                           MOVB    (R2)+,R5        ;PUT "EXPECTED" IN R5
030454  120504                           CMPB    R5,R4           ;IS RECEIVED DATA CORRECT?
030456  001401                           BEQ     44$             ;BR IF YES
030460  104010                           HLT     10              ;RECEIVE DATA ERROR
030462  005303                   44$:    DEC     R3              ;DEC CHARACTER COUNT
030464  001364                           BNE     43$             ;BR IF NOT DONE YET

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

030466  004737  032652                   JSR     PC,INRDY        ;WAIT FOR INRDY
030472  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030474  021204                           021204                  ;GET FIRST HALF OF CRC
030476  116137  000004  001252           MOVB    4(R1),TEMP3     ;PUT IN TEMP3
030504  042737  177400  001252           BIC     #177400,TEMP3   ;CLEAR HI BYTE
030512  004737  032652                   JSR     PC,INRDY        ;WAIT FOR INRDY
030516  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030520  021244                           021244
030522  016104  000004                   MOV     4(R1),R4        ;PUT "FOUND" IN R4
030526  042704  000374                   BIC     #374,R4         ;CLEAR UNWANTED BITS
030532  012705  000003                   MOV     #3,R5           ;PUT "EXPECTED" IN R5
030536  120504                           CMPB    R5,R4           ;ARE IN BCC MATCH AND BLOCK END SET?
030540  001401                           BEQ     51$
030542  104042                           HLT     42              ;IN BCC MATCH ERROR
030544                           51$:
030544  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030546  021204                           021204                  ;GET LAST HALF
030550  116137  000004  001251           MOVB    4(R1),TEMP2+1   ;PUT IN TEMP2
030556  042737  000377  001250           BIC     #377,TEMP2      ;CLEAR LO BYTE
030564  053737  001250  001252           BIS     TEMP2,TEMP3     ;16 BIT BCC NOW IN TEMP3
030572  023737  033032  001252           CMP     CALBCC,TEMP3    ;IS IT CORRECT?
030600  001401                           BEQ     5$              ;BR IF OK
030602  104027                           HLT     27
```

```
030604  104400                      5$:     SCOPE                           ;SCOPE THIS TEST



                                            ;************************** TEST 57 ***************************
                                            ;*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 CHARACTERS  AND A BLOCK END WERE RECEIVED, AND IN ACTIVE IS CLEAR
                                            ;:**************************************************************


                                            :  TEST 57
                                            ;--------------
030606  012737  000057  001226      TST57:  MOV     #57,TSTNO
030614  012737  031026  001216              MOV     #TST60,NEXT
                                                                            ;R1 CONTAINS BASE DMC11 ADDRESS
030622  104412                              MSTCLR                          ;MASTER CLEAR DMC11
030624  005061  000004                      CLR     4(R1)                   ;CLEAR PORT4
030630  104414                              ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030632  122117                              122117                          ;PUT LU IN BITSTUFF MODE
030634  004737  033374                      JSR     PC,CLRIO                ;DO THIS AFTER MODE IS SET
030640  012711  004000                      MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
030644  012702  033614                      MOV     #MESDAT,R2              ;R2 POINTS TO MESSAGE
030650  012700  000003                      MOV     #3,R0                   ;R0 = CHAR COUNT
030654  004737  033176                      JSR     PC,SYNLD                ;LOAD SILO WITH TWO FLAGS
030660  004737  032176                      JSR     PC,OUTRDY               ;WAIT FOR OUTRDY
030664  004537  033332                      JSR     R5,MESLD                ;LOAD MESSAGE IN SILO
030670  033614                              MESDAT                          ;START OF MESSAGE
030672  000004                              4                               ;CHARACTER COUNT
030674  004737  032044                      JSR     PC,OCOR                 ;WAIT FOR OCOR
030700  104415  000065                      DATACLK,        65              ;CLOCK DATA (EMPTY SILO)
030704  004537  033332                      JSR     R5,MESLD                ;PUT MORE CHARACTERS IN SILO
030710  033614                              MESDAT
030712  000004                              4
030714  004737  032044                      JSR     PC,OCOR
030720  104415  000006                      DATACLK,        6               ;CLOCK UNTIL RTS IS CLEARED
030724  104414                              ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030726  021264                              021264                          ;GET RTS
030730  032761  000040  000004              BIT     #BIT5,4(R1)             ;IS IT CLEAR?
030736  001401                              BEQ     5$                      ;BR IF YES
030740  104034                              HLT     34                      ;ERROR, RTS NOT CLEAR
030742  104415  000041              5$:     DATACLK,        41              ;CLOCK XMITTER SOME MORE
030746  004737  032652              1$:     JSR     PC,INRDY                ;OK LETS CHECK WHAT WAS RECEIVED
030752  104414                              ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030754  021204                              021204                          ;GET RECEIVE DATA
030756  016104  000004                      MOV     4(R1),R4                ;PUT IT IN R4
030762  112205                              MOVB    (R2)+,R5                ;R5 = "EXPECTED"
030764  120504                              CMPB    R5,R4                   ;IS DATA CORRECT?
030766  001401                              BEQ     2$                      ;BR IF OK
030770  104010                              HLT     10                      ;DATA ERROR
030772  005300                      2$:     DEC     R0                      ;DEC CHAR COUNT
030774  001364                              BNE     1$                      ;BR IF NOT DONE YET
030776  004737  032652                      JSR     PC,INRDY                ;WAIT FOR INRDY
031002  104414                              ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031004  021244                              021244                          ;READ LU-12
031006  016104  000004                      MOV     4(R1),R4                ;PUT "FOUND" IN R4
```

```
031012  012705  000022                  MOV     #22,R5          ;PUT "EXPECTED" IN R5
031016  120504                          CMPB    R5,R4           ;ARE BLOCK END AND IN RDY SET?
                                                                ;AND IN ACTIVE AND IN BCC MATCH CLEAR?
031020  001401                          BEQ     6$              ;BR IF YES
031022  104032                          HLT     32              ;ERROR, BLOCK END NOT SET
                                                                ;OR IN BCC MATCH NOT CLEAR
                                                                ;OR IN ACTIVE NOT CLEAR
031024                          6$:
031024  104400                          SCOPE                   ;SCOPE THIS TEST


                                        ;************************* TEST 60 *************************
                                        ;*BITSTUFF CABLE DATA TEST
                                        ;*THIS TEST LOADS OUT SILO WITH THE FOLLOWING:
                                        ;*2 FLAGS,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 60
                                        ;--------------
031026  012737  000060  001226  TST60:  MOV     #60,TSTNO
031034  012737  031450  001216          MOV     #TST61,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
031042  104412                          MSTCLR                  ;MASTER CLEAR DMC11
031044  032737  040000  001366          BIT     #BIT14,STAT1    ;SKIP TEST IF NO
031052  001575                          BEQ     3$              ;LOOPBACK CONNECTOR ON
031054  005061  000004                  CLR     4(R1)           ;CLEAR PORT4
031060  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031062  122117                          122117                  ;PUT LINE UNIT IN BITSTUFF MODE
031064  004737  033374                  JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
031070  012711  004000                  MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
031074  004737  033176                  JSR     PC,SYNLD        ;LOAD TWO FLAGS
031100  012737  102010  033030          MOV     #CRC.CCITT,XPOLY;LOAD POLYNOMIAL FOR SOFT CRC CALC
031106  005037  031142                  CLR     6$              ;CLEAR OLD BCC
031112  005137  031142                  COM     6$              ;ADJUST TO -1 FOR SDLC
031116  012703  000020                  MOV     #16.,R3         ;CHARACTER COUNT
031122  012702  033620                  MOV     #FLTDAT,R2      ;R2= POINTER
031126  112237  031140          7$:     MOVB    (R2)+,5$        ;LOAD CHAR FOR SOFT BCC CALC.
031132  004537  032706                  JSR     R5,SIMBCC       ;CALC SOFT BCC
031136  000010                          10                      ;SHIFT COUNT
031140  000000                  5$:     0                       ;CHARACTER
031142  000000                  6$:     0                       ;OLD BCC
031144  013737  033032  031142          MOV     CALBCC,6$       ;LOAD OLD BCC
031152  005303                          DEC     R3              ;DEC COUNT
031154  001364                          BNE     7$              ;BR IF NOT DONE YET
031156  005137  033032                  COM     CALBCC          ;ADJUST CALBCC FOR SDLC
031162  004537  033332                  JSR     R5,MESLD        ;LOAD SILO
031166  033620                          FLTDAT                  ;MESSAGE ADDRESS
031170  000020                          16.                     ;CHARACTER COUNT
031172  004737  033306                  JSR     PC,EOM          ;LOAD AN EOM
031176  004737  033306                  JSR     PC,EOM
031202  004537  033332                  JSR     R5,MESLD        ;LOAD SILO
031206  033620                          FLTDAT                  ;MESSAGE ADDRESS
```

```
031210  000020                            16.                         ;CHARACTER COUNT
031212  004737  033306                    JSR     PC,EOM              ;LOAD AN EOM
031216  004737  033306                    JSR     PC,EOM
031222  004537  033332                    JSR     R5,MESLD            ;LOAD SILO
031226  033620                            FLTDAT                      ;MESSAGE ADDRESS
031230  000020                            16.                         ;CHARACTER COUNT
031232  004737  033306                    JSR     PC,EOM              ;LOAD AN EOM
031236  004737  033306                    JSR     PC,EOM
031242  004737  032044                    JSR     PC,OCOR             ;WAIT FOR OCOR
031246  005011                            CLR     (R1)                ;CLEAR LINE UNIT LOOP
031250  012700  000003                    MOV     #3,R0               ;R0 = MESSAGE COUNT
031254  012703  000020                    MOV     #16.,R3             ;R3= CHARACTER COUNT
031260  012702  033620                    MOV     #FLTDAT,R2          ;LOAD MESSAGE POINTER IN R2
031264  004737  032652            1$:     JSR     PC,INRDY            ;WAIT FOR INRDY
031270  104414                            ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031272  021204                            021204                      ;GET DATA FROM IN SILO
031274  016104  000004                    MOV     4(R1),R4            ;PUT CHARACTER IN "FOUND"
031300  112205                            MOVB    (R2)+,R5            ;PUT "EXPECTED" IN R5
031302  120504                            CMPB    R5,R4               ;IS RECEIVED DATA CORRECT
031304  001401                            BEQ     2$                  ;BR IF OK
031306  104025                            HLT     25                  ;DATA ERROR
031310                            2$:
031310  005303                            DEC     R3                  ;DEC CHARACTER COUNT
031312  001364                            BNE     1$                  ;BR IF NOT DONE THIS MESSAGE
031314  012703  000020                    MOV     #16.,R3             ;RESET CHARACTER COUNT

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

031320  004737  032652                    JSR     PC,INRDY            ;WAIT FOR INRDY
031324  104414                            ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031326  021204                            021204                      ;GET FIRST HALF OF CRC
031330  116137  000004  001252            MOVB    4(R1),TEMP3         ;PUT IN TEMP3
031336  042737  177400  001252            BIC     #177400,TEMP3       ;CLEAR HI BYTE
031344  004737  032652                    JSR     PC,INRDY            ;WAIT FOR INRDY
031350  104414                            ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031352  021244                            021244
031354  016104  000004                    MOV     4(R1),R4            ;PUT "FOUND" IN R4
031360  042704  000374                    BIC     #374,R4             ;CLEAR UNWANTED BITS
031364  012705  000003                    MOV     #3,R5               ;PUT "EXPECTED" IN R5
031370  120504                            CMPB    R5,R4               ;ARE IN BCC MATCH AND BLOCK END SET?
031372  001401                            BEQ     25$
031374  104042                            HLT     42                  ;IN BCC MATCH ERROR
031376                            25$:
031376  104414                            ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031400  021204                            021204                      ;GET LAST HALF
031402  116137  000004  001251            MOVB    4(R1),TEMP2+1       ;PUT IN TEMP2
031410  042737  000377  001250            BIC     #377,TEMP2          ;CLEAR LO BYTE
031416  053737  001250  001252            BIS     TEMP2,TEMP3         ;16 BIT BCC NOW IN TEMP3
031424  023737  033032  001252            CMP     CALBCC,TEMP3        ;IS IT CORRECT?
031432  001401                            BEQ     4$                  ;BR IF OK
031434  104027                            HLT     27
031436  012702  033620            4$:     MOV     #FLTDAT,R2          ;RESET MESSAGE POINTER
031442  005300                            DEC     R0                  ;DECREMENT COUNTER
031444  001307                            BNE     1$                  ;BR IF NOT DONE
031446  104400                    3$:     SCOPE                       ;SCOPE THIS TEST
```

```
                                                 ;;*************************- TEST 61 ***************************
                                                 ;;*BITSTUFF CABLE DATA TEST
                                                 ;;*THIS TEST LOADS OUT SILO WITH THE FOLLOWING:
                                                 ;;*2 FLAGS,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 61
                                                 ;--------------
031450  012737  000061  001226          TST61:   MOV     #61,TSTNO
031456  012737  003364  001216                   MOV     #.EOP,NEXT
                                                                         ;R1 CONTAINS BASE DMC11 ADDRESS
031464  104412                                   MSTCLR                  ;MASTER CLEAR DMC11
031466  032737  040000  001366                   BIT     #BIT14,STAT1    ;SKIP TEST IF NO
031474  001545                                   BEQ     3$              ;LOOPBACK CONNECTOR ON
031476  005061  000004                           CLR     4(R1)           ;CLEAR PORT4
031502  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031504  122117                                   122117                  ;PUT LINE UNIT IN BITSTUFF MODE
031506  004737  033374                           JSR     PC,CLRIO        ;DO THIS AFTER MODE IS SET
031512  012711  004000                           MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
031516  004737  033176                           JSR     PC,SYNLD        ;LOAD TWO FLAGS
031522  012737  102010  033030                   MOV     #CRC.CCITT,XPOLY;LOAD POLYNOMIAL FOR SOFT CRC CALC
031530  005037  031564                           CLR     6$              ;CLEAR OLD BCC
031534  005137  031564                           COM     6$              ;ADJUST TO -1 FOR SDLC
031540  012703  000073                           MOV     #59.,R3         ;CHARACTER COUNT
031544  012702  033614                           MOV     #MESDAT,R2      ;R2= POINTER
031550  112237  031562                  7$:      MOVB    (R2)+,5$        ;LOAD CHAR FOR SOFT BCC CALC.
031554  004537  032706                           JSR     R5,SIMBCC       ;CALC SOFT BCC
031560  000010                                   10                      ;SHIFT COUNT
031562  000000                          5$:      0                       ;CHARACTER
031564  000000                          6$:      0                       ;OLD BCC
031566  013737  033032  031564                   MOV     CALBCC,6$       ;LOAD OLD BCC
031574  005303                                   DEC     R3              ;DEC COUNT
031576  001364                                   BNE     7$              ;BR IF NOT DONE YET
031600  005137  033032                           COM     CALBCC          ;ADJUST CALBCC FOR SDLC
031604  004537  033332                           JSR     R5,MESLD        ;LOAD SILO
031610  033614                                   MESDAT                  ;MESSAGE ADDRESS
031612  000073                                   59.                     ;CHARACTER COUNT
031614  004737  033306                           JSR     PC,EOM          ;LOAD AN EOM
031620  004737  033306                           JSR     PC,EOM
031624  004737  032044                           JSR     PC,OCOR          ;WAIT FOR OCOR
031630  005011                                   CLR     (R1)            ;CLEAR LINE UNIT LOOP
031632  012700  000073                           MOV     #59.,R0         ;R0= CHARACTER COUNT
031636  012702  033614                           MOV     #MESDAT,R2      ;LOAD MESSAGE POINTER IN R2
031642  004737  032652                  1$:      JSR     PC,INRDY        ;WAIT FOR INRDY
031646  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031650  021204                                   021204                  ;GET DATA FROM IN SILO
031652  016104  000004                           MOV     4(R1),R4        ;PUT CHARACTER IN "FOUND"
031656  112205                                   MOVB    (R2)+,R5        ;PUT "EXPECTED" IN R5
031660  120504                                   CMPB    R5,R4           ;IS RECEIVED DATA CORRECT
031662  001401                                   BEQ     2$              ;BR IF OK
031664  104025                                   HLT     25              ;DATA ERROR
```

```
 4967  031666                                        2$:
 4968  031666  005300                                         DEC     R0                ;DECREMENT COUNTER
 4969  031670  001364                                         BNE     1$                ;BR IF NOT DONE
 4970
 4971                                                         ;CHECK TO SEE THAT IN BCC MATCH IS SET
 4972                                                         ;AND THAT THE BCC WAS RECEIVED CORRECTLY
 4973
 4974  031672  004737  032652                                 JSR     PC,INRDY          ;WAIT FOR INRDY
 4975  031676  104414                                         ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4976  031700  021204                                         021204                    ;GET FIRST HALF OF CRC
 4977  031702  116137  000004  001252                         MOVB    4(R1),TEMP3       ;PUT IN TEMP3
 4978  031710  042737  177400  001252                         BIC     #177400,TEMP3     ;CLEAR HI BYTE
 4979  031716  004737  032652                                 JSR     PC,INRDY          ;WAIT FOR INRDY
 4980  031722  104414                                         ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4981  031724  021244                                         021244
 4982  031726  016104  000004                                 MOV     4(R1),R4          ;PUT "FOUND" IN R4
 4983  031732  042704  000374                                 BIC     #374,R4           ;CLEAR UNWANTED BITS
 4984  031736  012705  000003                                 MOV     #3,R5             ;PUT "EXPECTED" IN R5
 4985  031742  120504                                         CMPB    R5,R4             ;ARE IN BCC MATCH AND BLCCK END SET?
 4986  031744  001401                                         BEQ     25$
 4987  031746  104042                                         HLT     42                ;IN BCC MATCH ERROR
 4988  031750                                        25$:
 4989  031750  104414                                         ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4930  031752  021204                                         021204                    ;GET LAST HALF
 4991  031754  116137  000004  001251                         MOVB    4(R1),TEMP2+1     ;PUT IN TEMP2
 4992  031762  042737  000377  001250                         BIC     #377,TEMP2        ;CLEAR LO BYTE
 4993  031770  053737  001250  001252                         BIS     TEMP2,TEMP3       ;16 BIT BCC NOW IN TEMP3
 4994  031776  023737  033032  001252                         CMP     CALBCC,TEMP3      ;IS IT CORRECT?
 4995  032004  001401                                         BEQ     3$                ;BR IF OK
 4996  032006  104027                                         HLT     27
 4997  032010  104400                               3$:      SCOPE                     ;SCOPE THIS TEST
 4998                                        00300
 4999                                        00400
 5000                                        00500   ;SUBROUTINES
 5001                                        00600   ;-----------
 5002                                        00700
 5003  032012                                00800   GETSI:
 5004                                        00900            ;THIS SUBROUTINE READS LU 17, AND PUTS IT INTO NITCH.
 5005                                        01000            ;NITCH IS ROTATED LEFT UNTILL THE SI BIT IS IN CARRY
 5006                                        01100
 5007  032012  104414                        01200            ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5008  032014  021364                        01300            021364                    ;PORT4←LU 17
 5009  032016  017737  147370  032042        01400            MOV     @DMP04,NITCH      ;STORE LU 17
 5010  032024  106137  032042                01500            ROLB    NITCH
 5011  032030  106137  032042                01600            ROLB    NITCH
 5012  032034  106137  032042                01700            ROLB    NITCH             ;PUT SI IN THE CARRY BIT
 5013  032040  000207                        01800            RTS     PC
 5014  032042  000000                        01900   NITCH:   0
 5015                                        02000
 5016                                        02100
 5017  032044                                02200   OCOR:
 5018                                        02300            ;THIS SUBROUTINE SPINS ON OCOR
 5019                                        02400
 5020  032044  104414                                         ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5021  032046  021364                        02600            021364                    ;PORT4←LU 17
 5022  032050  032777  000020  147334        02700            BIT     #BIT4,@DMP04      ;IS OCOR SET?
```

```
5023 032056 001772                 02800           BEQ     OCOR            ;BR IF NO
5024 032060 000207                 02900           RTS     PC              ;OK OCOR IS SET, GO BACK
5025                               03000
5026                               03100
5027 032062                        03200   SYNC:
5028                               03300           ;THIS SUBROUNTINE LOADS THE SILO WITH THE NUMBER OF SYNC
5029                               03400           ;CHARACTERS PASSED TO IT IN THE WORD AFTER THE JSR CALL
5030                               03500           ;AND A NON-SYNC CHARACTER (301)
5031                               03600
5032 032062 013637 001246          03700           MOV     @(SP)+,TEMP1    ;GET COUNT
5033 032066 062746 000002          03800           ADD     #2,-(SP)        ;ADJUST STACK
5034 032072 012761 000026 000004   03900           MOV     #26,4(R1)       ;LOAD PORT4
5035 032100 104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5036 032102 122114                 04100           122114                  ;LOAD SYNC REGISTER
5037 032104 004737 032176          04200   1$:     JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
5038 032110 012761 000001 000004   04300           MOV     #1,4(R1)        ;LOAD PORT4
5039 032116 104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5040 032120 122111                 04500           122111                  ;SET SOM
5041 032122 012761 000026 000004   04600           MOV     #26,4(R1)       ;LOAD PORT4
5042 032130 104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5043 032132 122110                 04800           122110                  ;LOAD OUT DATA
5044 032134 005337 001246          04900           DEC     TEMP1           ;ALL DONE?
5045 032140 001361                 05000           BNE     1$              ;BR IF NOT
5046 032142 004737 032176          05100           JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
5047 032146 005061 000004          05200           CLR     4(R1)           ;LOAD PORT4
5048 032152 104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5049 032154 122111                 05400           122111                  ;SET SOM
5050 032156 012761 000301 000004   05500           MOV     #301,4(R1)      ;LOAD PORT4
5051 032164 104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5052 032166 122110                 05700           122110                  ;LOAD OUT DATA
5053 032170 004737 032044          05800           JSR     PC,OCOR         ;WAIT FOR OCOR
5054 032174 000207                 05900           RTS     PC
5055                               06000
5056                               06100
5057 032176                        06200   OUTRDY:
5058                               06300           ;THIS SUBROUTINE SPINS ON OUT READY
5059                               06400
5060 032176 005037 001256          06500           CLR     TEMP5           ;CLEAR TIMER
5061 032202                        06600   1$:
5062 032202 104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5063 032204 021224                 06800           021224                  ;PORT4+LU11
5064 032206 032777 000020 147176   06900           BIT     #BIT4,@DMF04    ;IS OUT RDY SET?
5065 032214 001004                 07000           BNE     2$              ;BR IF YES
5066 032216 005237 001256          07100           INC     TEMP5           ;INC TIMER
5067 032222 001367                 07200           BNE     1$              ;KEEP CHECKING IF NOT DONE
5068 032224 104036                 07300           HLT     36              ;ERROR, OUT READY NOT SET
5069 032226 000207                 07400   2$:     RTS     PC
5070                               07500
5071                               07600
5072 032230                        07700   CHAR:
5073                               07800           ;THIS SUBROUTINE LOADS THE SILO WITH 3 SYNCS
5074                               07900           ;AND THE CHARACTER PASSED TO IT.
5075                               08000
5076 032230 013637 001250          08100           MOV     @(SP)+,TEMP2    ;GET CHARACTER
5077 032234 062746 000002          08200           ADD     #2,-(SP)        ;ADJUST STACK
5078 032240 012737 000003 001246   08300           MOV     #3,TEMP1        ;SET FOR 3 SYNCS
```

```
 5079  032246  012761  000026  000004  08400                MOV     #26,4(R1)          ;LOAD PORT4
 5080  032254  104414                                       ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5081  032256  122114                          08600        122114                     ;LOAD SYNC REGISTER
 5082  032260  004737  032176                  08700  1$:   JSR     PC,OUTRDY          ;WAIT FOR OUTRDY
 5083  032264  012761  000001  000004          08800        MOV     #1,4(R1)           ;LOAD PORT4
 5084  032272  104414                                       ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5085  032274  122111                          09000        122111                     ;SET SOM
 5086  032276  012761  000026  000004          09100        MOV     #26,4(R1)          ;LOAD PORT4
 5087  032304  104414                                       ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5088  032306  122110                          09300        122110                     ;LOAD OUT DATA
 5089  032310  005337  001246                  09400        DEC     TEMP1              ;ALL DONE?
 5090  032314  001361                          09500        BNE     1$                 ;BR IF NOT
 5091  032316  004737  032176                  09600        JSR     PC,OUTRDY          ;WAIT FOR OUTRDY
 5092  032322  013761  001250  000004          09700        MOV     TEMP2,4(R1)        ;LOAD PORT4
 5093  032330  104414                                       ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5094  032332  122110                          09900        122110                     ;LOAD OUT DATA
 5095  032334  004737  032044                  10000        JSR     PC,OCOR            ;WAIT FOR OCOR
 5096  032340  000207                          10100        RTS     PC
 5097                                          10200
 5098                                          10300
 5099  032342                                  10400  CHARSD:
 5100                                          10500        ;THIS SUBROUTINE LOADS THE SILO WITH THE CHARACTER PASSED TO IT.
 5101                                          10600
 5102  032342  013637  001250                  10700        MOV     @(SP)+,TEMP2       ;GET CHARACTER
 5103  032346  062746  000002                  10800        ADD     #2,-(SP)           ;ADJUST STACK
 5104  032352  004737  032176                  10900        JSR     PC,OUTRDY          ;WAIT FOR OUTRDY
 5105  032356  013761  001250  000004          11000        MOV     TEMP2,4(R1)        ;LOAD PORT4
 5106  032364  104414                                       ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5107  032366  122110                          11200        122110                     ;LOAD OUT DATA
 5108  032370  004737  032176                  11300        JSR     PC,OUTRDY          ;WAIT FOR OUTRDY
 5109  032374  104414                                       ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5110  032376  122110                          11500        122110                     ;LOAD GARBAGE CHAR
 5111  032400  004737  032044                  11600        JSR     PC,OCOR            ;WAIT FOR OCOR
 5112  032404  000207                          11700        RTS     PC
 5113                                          11800
 5114                                          11900
 5115  032406                                  12000  SILOLD:
 5116                                          12100        ;THIS SUBROUTINE FILLS THE OUT SILO
 5117                                          12200        ; WITH A BINARY COUNT PATTERN
 5118                                          12300
 5119  032406  012737  000073  001250          12400        MOV     #73,TEMP2          ;LOAD COUNT
 5120  032414  005737  032646                  12500        TST     SCHAR              ;FIRST TIME HERE?
 5121  032420  100470                          12600        BMI     4$                 ;BR IF BITSTUFF
 5122  032422  001032                          12700        BNE     2$                 ;BR IF NO
 5123  032424  062737  000002  001250          12800        ADD     #2,TEMP2           ;ADD 2 TO CHARACTER COUNT
 5124  032432  012737  000003  001246          12900        MOV     #3,TEMP1           ;SET FOR 3 SYNCS
 5125  032440  012761  000026  000004          13000        MOV     #26,4(R1)          ;LOAD PORT4
 5126  032446  104414                                       ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5127  032450  122114                          13200        122114                     ;LOAD SYNC REGISTER
 5128  032452  004737  032176                  13300  1$:   JSR     PC,OUTRDY          ;WAIT FOR OUTRDY
 5129  032456  012761  000001  000004          13400        MOV     #1,4(R1)           ;LOAD PORT4
 5130  032464  104414                                       ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5131  032466  122111                          13600        122111                     ;SET SOM
 5132  032470  012761  000026  000004          13700        MOV     #26,4(R1)          ;LOAD PORT4
 5133  032476  104414                                       ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5134  032500  122110                          13900        122110                     ;LOAD OUT DATA
```

```
5135  032502  005337  001246          14000          DEC     TEMP1            ;ALL DONE?
5136  032506  001361                  14100          BNE     1$               ;BR IF NOT
5137  032510  004737  032176          14200   2$:    JSR     PC,OUTRDY        ;WAIT FOR OUTRDY
5138  032514  013761  032646  000004  14300          MOV     SCHAR,4(R.)      ;LOAD PORT4
5139  032522  104414                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5140  032524  122110                  14500          122110                   ;LOAD OUT DATA
5141  032526  005737  032650          14600          TST     STUFLG           ;BITSTUFF???
5142  032532  001407                  14700          BEQ     6$               ;BR IF NO
5143  032534  013737  032646  032546  14800          MOV     SCHAR,5$         ;IT IS SDLD SO CHECK BITSTUFFING
5144  032542  004537  033414          14900          JSR     RS,STFFCL        ;ADD ANY BIT STUFF CLOCK TICKS
5145  032546  000000                  15000   5$:    0                        ;CHARACTER
5146  032550  000010                  15100          10                       ;CHIFT COUNT
5147  032552  005237  032646          15200   6$:    INC     SCHAR            ;NEXT CHARACTER
5148  032556  022737  000400  032646  15300          CMP     #400,SCHAR       ;ALL DONE?
5149  032564  001403                  15400          BEQ     3$
5150  032566  005337  001250          15500          DEC     TEMP2            ;DECREMENT COUNT
5151  032572  001346                  15600          BNE     2$               ;BR IF NOT DONE
5152  032574  004737  032044          15700   3$:    JSR     PC,OCOR          ;WAIT FOR OCOR
5153  032600  000207                  15800          RTS     PC
5154  032602  005037  032646          15900   4$:    CLR     SCHAR            ;START PATTERN AT ZERO
5155  032606  012737  177777  032650  16000          MOV     #-1,STUFLG       ;SET BITSTUFF FLAG
5156  032614  005037  033612          16100          CLR     BITCON           ;CLEAR STUFF COUNT
5157  032620  062737  000002  001250  16200          ADD     #2,TEMP2         ;ADD 2 TO CHARACTER COUNT
5158  032626  012761  000001  000004  16300          MOV     #1,4(R1)         ;SET BIT0 IN PORT4
5159  032634  104414                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5160  032636  122111                  16500          122111                   ;SET SOM!
5161  032640  104414                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5162  032642  122110                  16700          122110                   ;LOAD GARBAGE CHAR
5163  032644  000721                  16800          BR      2$               ;GO LOAD SILO
5164  032646  000000                  16900   SCHAR: 0
5165  032650  000000                  17000   STUFLG: 0
5166                                  17100
5167                                  17200
5168  032652                          17300   INRDY:
5169                                  17400          ;THIS SUBROUTINE SPINS ON INRDY
5170                                  17500          ;IF INRDY FAILS TO SET THE DELAY TIMES OUT AND AN
5171                                  17600          ;ERROR IS REPORTED. FOR BETTER SCOPE LOOPS THIS
5172                                  17700          ;DELAY CAN BE MADE SHORTER BY ALTERING THE NUMBER
5173                                  17800          ;INITIALLY LOADED INTO TEMP1, THE SMALLER THE NUMBER
5174                                  17900          ;THE SHORTER THE DELAY. 0 IS THE LONGEST DELAY.
5175                                  18000
5176  032652  012737  000000  001246  18100          MOV     #0,TEMP1         ;SET UP DELAY COUNTER
5177  032660                          18200   1$:
5178  032660  104414                                 ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5179  032662  021244                  18400          021244                   ;PORT4+LU12
5180  032664  032777  000020  146520  18500          BIT     #BIT4,@DMP04     ;IS INRDY SET?
5181  032672  001004                  18600          BNE     2$               ;BR IF YES
5182  032674  005237  001246          18700          INC     TEMP1            ;INC DELAY
5183  032700  001367                  18800          BNE     1$               ;TRY AGAIN
5184  032702  104037                  18900          HLT     37               ;ERROR,NO INRDY
5185  032704  000207                  19000   2$:    RTS     PC               ;RETURN
5186                                  19100
5187                                  19200
5188  032706                                  SIMBCC:
5189                                                 ;THIS SUBROUTINE CALCULATES THE CRC USING POLYNOMIAL GIVEN
5190                                                 ;IN XPOLY. THE CORRECT CRC IS RETURNED IN CALBCC, AND THE
```

```
                                                ;STATE OF THE LSB OF THE BCC IS RETURNED IN THE C BIT.

032706  010046                                  MOV     R0,-(SP)         ;SAVE R0 ON STACK
032710  012537  001246                          MOV     (R5)+,TEMP1      ;TEMP1 = SHIFT COUNT
032714  012537  001250                          MOV     (R5)+,TEMP2      ;TEMP2 = CHARACTER
032720  012537  033032                          MOV     (R5)+,CALBCC     ;CALBCC = OLD BCC
032724  013700  033032                  1$:     MOV     CALBCC,R0        ;PUT OLD BCC IN R0
032730  000241                                  CLC
032732  006037  033032                          ROR     CALBCC           ;SHIFT OLD BCC
032736  006037  001250                          ROR     TEMP2            ;SHIFT CHARACTER
032742  005500                                  ADC     R0               ;ADD CHAR CARRY TO OLD BCC
032744  006000                                  ROR     R0               ;PUT BIT0 TO CARRY BIT
032746  103011                                  BCC     2$               ;CARRY IS FEEDBACK BIT
032750  013700  033030                          MOV     XPOLY,R0         ;IF FEEDBACK = 1
032754  043700  033032                          BIC     CALBCC,R0        ;EXCLUSIVLY OR XPOLY TO CALBCC
032760  043737  033030  033032                  BIC     XPOLY,CALBCC
032766  050037  033032                          BIS     R0,CALBCC
032772  005337  001246                  2$:     DEC     TEMP1            ;DEC SHIFT COUNT
032776  001352                                  BNE     1$               ;BR IF NOT DONE
033000  012737  000001  001246                  MOV     #1,TEMP1         ;GET SET TO INVERT BIT0
033006  013700  033032                          MOV     CALBCC,R0        ;PUT RESULT IN R0
033012  006000                                  ROR     R0               ;SHIFT BIT0 TO CARRY
033014  005537  001246                          ADC     TEMP1            ;INVERT CARRY TO BIT0 OF TEMP1
033020  006037  001246                          ROR     TEMP1            ;PUT INVERTED BIT IN CARRY
033024  012600                                  MOV     (SP)+,R0         ;RESTORE R0
033026  000205                                  RTS     R5               ;RETURN
033030  000000                          XPOLY:  0
033032  000000                          CALBCC: 0
        000200                          LRC8=200
        120001                          CRC16=120001
        102010                          CRC.CCITT=102010


033034                          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
033034  013637  001250          20300           MOV     @(SP)+,TEMP2     ;GET CHARACTER
033040  062746  000002          20400           ADD     #2,-(SP)         ;ADJUST STACK
033044  012737  000002  001246  20500           MOV     #2,TEMP1         ;SET FOR 2 SYNCS
033052  012761  000026  000004  20600           MOV     #26,4(R1)        ;LOAD PORT4
033060  104414                                  ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033062  122114                  20800           122114                   ;LOAD SYNC REGISTER
033064  004737  032176          20900   1$:     JSR     PC,OUTPDY        ;WAIT FOR OUTRDY
033070  012761  000001  000004  21000           MOV     #1,4(R1)         ;LOAD PORT4
033076  104414                                  ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033100  122111                  21200           122111                   ;SET SOM
033102  012761  000026  000004  21300           MOV     #26,4(R1)        ;LOAD PORT4
033110  104414                                  ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033112  122110                  21500           122110                   ;LOAD OUT DATA
033114  005337  001246          21600           DEC     TEMP1            ;ALL DONE?
033120  001361                  21700           BNE     1$               ;BR IF NOT
033122  004737  032176          21800           JSR     PC,OUTRDY        ;WAIT FOR OUTRDY
033126  013761  001250  000004  21900           MOV     TEMP2,4(R1)      ;LOAD PORT4
033134  104414                                  ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
033136  122110                           22100            122110                     ;LOAD OUT DATA
033140  004737  032044                   22200            JSR      PC,DCOR           ;WAIT FOR DCOR
033144  000207                           22300            RTS      PC
                                         22400
                                         22500
033146                                                GETQ0:
                                                          ;THIS SUBROUTINE READS THE STATE OF THE TRANSMIT
                                                          ;BCC LSB AND PUTS IT IN THE CARRY BIT

033146  104414                                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033150  021364                                            021364                     ;PORT4+LU-17
033152  106177  146234                                    ROLB     @DMP04            ;PUT Q0 IN CARRY
033156  000207                                            RTS      PC                ;RETURN


033160                                                GETQ1:
                                                          ;THIS SUBROUTINE READS THE STATE OF THE RECEIVE
                                                          ;BCC LSB AND PUTS IT IN THE CARRY BIT

033160  104414                                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033162  021364                                            021364                     ;PORT4+LU-17
033164  106177  146222                                    ROLB     @DMP04            ;PUT Q0 IN CARRY
033170  106177  146216                                    ROLB     @DMP04            ;PUT Q1 IN CARRY
033174  000207                                            RTS      PC                ;RETURN


033176                                   22800  SYNLD:
                                         22900            ;THIS SUBROUTINE LOADS OUT SILO WITH
                                         23000            ;2 SYNC CHARACTERS WITH SOM SET
                                         23100
033176  012737  000002  001246           23200            MOV      #2,TEMP1          ;LOAD COUNTER FOR 2 SYNCS
033204  012761  000026  000004           23300            MOV      #26,4(R1)         ;PORT4+26
033212  104414                                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033214  122114                           23500            122114                     ;LOAD SYNC REG
033216  004737  032176                   23600  1$:       JSR      PC,OUTRDY         ;WAIT FOR OUTRDY
033222  012761  000001  000004           23700            MOV      #1,4(R1)          ;LOAD PORT4
033230  104414                                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033232  122111                           23900            122111                     ;SET SOM
033234  012761  000026  000004           24000            MOV      #26,4(R1)         ;PORT+26
033242  104414                                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033244  122110                           24200            122110                     ;LOAD OUT DATA WITH SYNC
033246  005337  001246                   24300            DEC      TEMP1             ;DECREMENT COUNTER
033252  001361                           24400            BNE      1$                ;BR IF NOT DONE
033254  000207                           24500            RTS      PC                ;RETURN
                                         24600
                                         24700
033256                                   24800  SOM:
                                         24900            ;THIS SUBROUTINE LOADS SOM AND OUT DATA WITH A
                                         25000            ;GARBAGE CHARACTER (0)
                                         25100
033256  004737  032176                   25200            JSR      PC,OUTRDY         ;WAIT FOR OUTRDY
033262  012761  000001  000004           25300            MOV      #1,4(R1)          ;PORT4+1
033270  104414                                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033272  122111                           25500            122111                     ;SET SOM
033274  005061  000004                   25600            CLR      4(R1)             ;CLEAR DATA CHAR
033300  104414                                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
 5303  033302  122110                    25800            122110                      ;LOAD GARBAGE CHARACTER
 5304  033304  000207                    25900            RTS     PC                  ;RETURN
 5305                                    26000
 5306                                    26100
 5307  033306                            26200   EOM:
 5308                                    26300            ;THIS SUBROUTINE LOADS EOM AND OUT DATA WITH A
 5309                                    26400            ;GARBAGE CHARACTER (2) TO ENABLE TRANSMISSION OF BCC
 5310                                    26500
 5311  033306  004737  032176            26600            JSR     PC,OUTRDY           ;WAIT FOR OUTRDY
 5312  033312  012761  000002  000004    26700            MOV     #2,4(R1)            ;PORT4+2
 5313  033320  104414                                     ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5314  033322  122111                    26900            122111                      ;SET EOM
 5315  033324  104414                                     ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5316  033326  122110                    27100            122110                      ;LOAD GARBAGE CHARACTER
 5317  033330  000207                    27200            RTS     PC                  ;RETURN
 5318                                    27300
 5319                                    27400
 5320  033332                            27500   MESLD:
 5321                                    27600            ;THIS SUBROUTINE LOADS SILO WITH MESSAGE
 5322                                    27700            ;THE FIRST ARUGUMENT IS THE ADDRESS OF THE MESSAGE
 5323                                    27800            ;THE SECOND ARGUMENT IS THE NUMBER OF CHARACTERS IN THE MESSAGE
 5324                                    27900
 5325  033332  010046                    28000            MOV     R0,-(SP)            ;SAVE R0
 5326  033334  012500                    28100            MOV     (R5)+,R0            ;R0=MESSAGE POINTER
 5327  033336  012537  001246            28200            MOV     (R5)+,TEMP1         ;TEMP1=CHARACTER COUNT
 5328  033342  004737  032176            28300   1$:      JSR     PC,OUTRDY           ;WAIT FOR OUT RDY
 5329  033346  112061  000004            28400            MOVB    (R0)+,4(R1)         ;LOAD PORT4 WITH CHARACTER
 5330  033352  104414                                     ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5331  033354  122110                    28600            122110                      ;LOAD OUT DATA SILO
 5332  033356  005337  001246            28700            DEC     TEMP1               ;DEC CHAR COUNT
 5333  033362  001367                    28800            BNE     1$                  ;BR IF NOT DONE
 5334  033364  004737  032044            28900            JSR     PC,OCOR             ;WAIT FOR OCOR
 5335  033370  012600                    29000            MOV     (SP)+,R0            ;RESTORE R0
 5336  033372  000205                    29100            RTS     R5                  ;RETURN
 5337                                    29200
 5338                                    29300
 5339  033374                            29400   CLRIC:
 5340                                    29500            ;THIS SUBROUTINE SETS IN CLR AND OUT CLR TO
 5341                                    29600            ;CLEAR THE TRANSMIT AND RECEIVE BCC REGISTERS
 5342                                    29700
 5343  033374  012761  000200  000004    29800            MOV     #BIT7,4(R1)         ;LOAD PORT4
 5344  033402  104414                                     ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5345  033404  122112                    30000            122112                      ;SET IN CLR!
 5346  033406  104414                                     ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 5347  033410  122111                    30200            122111                      ;SET OUT CLR!
 5348  033412  000207                    30300            RTS     PC                  ;RETURN
 5349                                    30400
 5350                                    30500
 5351  033414                            30600   STFFCL:
 5352                                    30700            ;THIS SUBROUTINE ADDS ANY NECESSSARY BIT STUFF CLOCK TICKS
 5353                                    30800            ;FIRST ARGUMENT IS CHAR, SECOND ARGUMENT IS SHIFT COUNT.
 5354                                    30900
 5355  033414  010046                    31000            MOV     R0,-(SP)            ;SAVE R0
 5356  033416  012500                    31100            MOV     (R5)+,R0            ;PUT CHAR IN R0
 5357  033420  012537  001252            31200            MOV     (R5)+,TEMP3         ;PUT SHIFT COUNT IN TEMP3
 5358  033424  106000                    31300   1$:      RORB    R0                  ;LOOK AT NEXT BIT
 5359  033426  103403                    31400          BCS     2$                :BR IF A MARK
 5360  033430  005037  033612            31500          CLR     BITCON            :IT WAS A SPACE, CLEAR 1'S COUNTER
 5361  033434  000412                    31600          BR      3$                :CONTINUE
 5362  033436  005237  033612            31700   2$:    INC     BITCON            :INC CONSECUTIVE 1'S COUNTER
 5363  033442  022737  000005  033612    31800          CMP     #5,BITCON         :IS IT 5 YET?
 5364  033450  001004                    31900          BNE     3$                :BR IF NO
 5365  033452  005037  033612            32000          CLR     BITCON            :YES!  SO START AGAIN
 5366  033456  104415  000001            32100          DATACLK,        1         :GIVE EXTRA TICK TO STUFF ZERO
 5367  033462  005337  001252            32200   3$:    DEC     TEMP3             :DEC SHIFT COUNT
 5368  033466  001356                    32300          BNE     1$                :BR IF NOT DONE
 5369  033470  012600                    32400          MOV     (SP)+,R0          :RESTORE R0
 5370  033472  000205                    32500          RTS     R5                :RETURN
 5371                                    32600
 5372                                    32700
 5373  033474                            32800   STFFCK:
 5374                                    32900          ;THIS SUBROUTINE CHECKS TO SEE IF TRANSMITTER
 5375                                    33000          ;IS STUFFING ZEROS WHEN IT SHOULD. FIRST ARGUMENT
 5376                                    33100          ;IS THE CHARACTER, SECOND ARGUMENT IS SHIFT COUNT.
 5377                                    33200
 5378  033474  010046                    33300          MOV     R0,-(SP)          :SAVE R0
 5379  033476  012500                    33400          MOV     (R5)+,R0          :PUT CHAR IN R0
 5380  033500  012537  001252            33500          MOV     (R5)+,TEMP3       :PUT SHIFT COUNT IN TEMP3
 5381  033504  106000                    33600   1$:    RORB    R0                :SHIFT OUT NEXT BIT
 5382  033506  103403                    33700          BCS     2$                :BR IF IT IS A MARK
 5383  033510  005037  033612            33800          CLR     BITCON            :IT WAS A SPACE, CLEAR 1'S COUNTER
 5384  033514  000416                    33900          BR      3$                :CONTINUE
 5385  033516  005237  033612            34000   2$:    INC     BITCON            :INC CONSECUTIVE 1'S COUNTER
 5386  033522  022737  000005  033612    34100          CMP     #5,BITCON         :5 IN A ROW YET?
 5387  033530  001010                    34200          BNE     3$                :BR IF NO
 5388  033532  005037  033612            34300          CLR     BITCON            :YES, SO START OVER
 5389  033536  104415  000001            34400          DATACLK,        1         :EXTRA TICK TO STUFF ZERO
 5390  033542  004737  032012            34500          JSR     PC,GETSI          :LOOK AT WINDOW
 5391  033546  103001                    34600          BCC     3$                :IS IT A ZERO, BR IF YES
 5392  033550  104030                    34700          HLT     30                :NO, ERROR ZERO WAS NOT STUFFED
 5393  033552  005337  001252            34800   3$:    DEC     TEMP3             :DEC SHIFT COUNT
 5394  033556  001352                    34900          BNE     1$                :BR IF NOT DONE
 5395  033560  012600                    35000          MOV     (SP)+,R0          :RESTORE R0
 5396  033562  000205                    35100          RTS     R5                :RETURN
 5397                                    35200
 5398                                    35300
 5399  033564                            35400   CTSDLY:
 5400                                    35500          ;THIS SUBROUTINE WASTES TIME UNTIL CTS SETS,
 5401                                    35600          ;BUT HOPEFULLY NOT SO LONG THAT THE SILO RUNS OUT
 5402                                    35700
 5403  033564  010046                    35800          MOV     R0,-(SP)          :SAVE R0
 5404  033566  012700  000032            35900          MOV     #32,R0            :LOAD R0 WITH COUNT
 5405  033572  027777  145406  145404    36000   1$:    CMP     @TKCSR,@TKCSR     :WASTE TIME
 5406  033600  005300                    36100          DEC     R0                :DECREMENT COUNTER
 5407  033602  001373                    36200          BNE     1$                :DO IT AGAIN IF NOT = 0
 5408  033604  012600                    36300          MOV     (SP)+,R0          :RESTORE R0
 5409  033606  000207                    36400          RTS     PC                :RETURN
 5410                                    36500
 5411                                    36600
 5412  033610  000176                    36700   FLAG:   ↑B<01111110>              :FLAG CHARACTER
 5413  033612  000000                    36800   BITCON: 0
 5414  033614     000     125     252    36900   MESDAT: .BYTE   0,125,252,377
```

```
033617     377
033620     001      002      004  37000   FLTDAT: .BYTE   1,2,4,10,20,40,100,200,376,375,373,367,357,337,277,177
033623     010      020      040
033626     100      200      376
033631     375      373      367
033634     357      337      277
033637     177
033640     100      140      160  37100   STUFDT: .BYTE   100,140,160,170,3,300,174,176,177,1
033643     170      003      300
033646     174      176      177
033651     001
033652     363      347      317  37200           .BYTE   363,347,317,200,0,377,377,377,200,37
033655     200      000      377
033660     377      377      200
033663     037
                                  37300   .EVEN
033664  046377   047111   020105  00300   EM1:    .ASCIZ  <377>/LINE UNIT INITALIZATION TEST/
033722  046377   047111   020105  00400   EM2:    .ASCIZ  <377>↑LINE UNIT REGISTER READ/ONLY TEST↑
033765     377   044514   042516  00500   EM3:    .ASCIZ  <377>↑LINE UNIT REGISTER WRITE/READ TEST↑
034031     377   044514   042516  00600   EM4:    .ASCIZ  <377>/LINE UNIT INTERNAL CLOCK FAILURE
034073     377   051124   047101  00700   EM5:    .ASCIZ  <377>/TRANSMITTER DATA ERROR/
034123     377   042522   042503  00800   EM6:    .ASCIZ  <377>/RECEIVER TEST/
034142  051377   041505   044505  00900   EM7:    .ASCIZ  <377>/RECEIVER DATA ERROR/
034167     377   047515   042504  01000   EM10:   .ASCIZ  <377>/MODEM SIGNAL ERROR/
034213     377   051124   047101  01100   EM11:   .ASCIZ  <377>/TRANSMITTER CRC ERROR/
034242  051377   041505   044505  01200   EM12:   .ASCIZ  <377>/RECEIVER CRC ERROR/
034266  044777   020116   041502  01300   EM13:   .ASCIZ  <377>/IN BCC MATCH ERROR (LU REG 12)/
034326  052377   040522   051516  01400   EM14:   .ASCIZ  <377>/TRANSMITTER FAILED TO GO TO MARK STATE
034376  041777   041101   042514  01500   EM15:   .ASCIZ  <377>/CABLE DATA TEST/
034417     377   046106   043501  01600   EM16:   .ASCIZ  <377>/FLAG ERROR/
034433     377   051124   047101  01700   EM17:   .ASCIZ  <377>/TRANSMITTER FAILED TO STUFF A ZERO
034477     377   053523   052111  01800   EM20:   .ASCIZ  <377>/SWITCH PAC TEST/
034520  040777   047502   052122  01900   EM21:   .ASCIZ  <377>/ABORT ERROR/
034535     377   051124   047101  02000   EM22:   .ASCIZ  <377>/TRANSMITTER ERROR/
034560  044377   046101   020106  02100   EM23:   .ASCIZ  <377>/HALF DUPLEX TEST/
034602  047777   052125   051040  02200   EM24:   .ASCIZ  <377>/OUT READY NOT SET/
034625     377   047111   051040  02300   EM25:   .ASCIZ  <377>/IN READY NOT SET/
                                  02400
034647     377   054105   042520  02500   DH1:    .ASCIZ  <377>/EXPECTED  FOUND/
034670  042777   050130   041505  02600   DH2:    .ASCIZ  <377>/EXPECTED  FOUND  LU-REGISTER/
034726  041777   040510   040522  02700   DH3:    .ASCIZ  <377>/CHARACTER    BIT THAT FAILED/
034764  041777   051117   042522  02800   DH4:    .ASCIZ  <377>/CORRECT CRC    BIT THAT FAILED
035024  042777   050130   041505  02900   DH5:    .ASCIZ  <377>/EXPECTED   FOUND   SHIFT/
035056  042777   050130   041505  03000   DH6:    .ASCIZ  <377>/EXPECTED   FOUND   CHARACTER   SHIFT
035124  041377   047514   045503  03100   DH7:    .ASCIZ  <377>/BLOCK END NOT SET/
035147     377   052122   020123  03200   DH10:   .ASCIZ  <377>/RTS DID NOT CLEAR/
                                  03300   .EVEN
                                  03400
035172  000002                    03500   DT1:    2
035174     003      007           03600           .BYTE   3,7
035176  001272                    03700           SAVR5
035200     003      002           03800           .BYTE   3,2
035202  001270                    03900           SAVR4
035204  000003                    04000   DT2:    3
035206     003      007           04100           .BYTE   3,7
035210  001272                    04200           SAVR5
```

```
035212     003     010         04300                 .BYTE    3,10
035214  001270                 04400                 SAVR4
035216     003     002         04500                 .BYTE    3,2
035220  001264                 04600                 SAVR2
035222  000002                 04700       DT3:      2
035224     003     017         04800                 .BYTE    3,17
035226  001272                 04900                 SAVR5
035230     002     002         05000                 .BYTE    2,2
035232  001266                 05100                 SAVR3
035234  000002                 05200       DT4:      2
035236     006     021         05300                 .BYTE    6,21
035240  033032                 05400                 CALBCC
035242     002     002         05500                 .BYTE    2,2
035244  001266                 05600                 SAVR3
035246  000003                 05700       DT5:      3
035250     001     011         05800                 .BYTE    1,11
035252  001300                 05900                 ZERO
035254     001     011         06000                 .BYTE    1,11
035256  001302                 06100                 ONE
035260     002     002         06200                 .BYTE    2,2
035262  001260                 06300                 SAVR0
035264  000003                 06400       DT6:      3
035266     001     011         06500                 .BYTE    1,11
035270  001302                 06600                 ONE
035272     001     011         06700                 .BYTE    1,11
035274  001300                 06800                 ZERO
035276     002     002         06900                 .BYTE    2,2
035300  001260                                       SAVR0
035302  000004                             DT7:      4
035304     001     011                               .BYTE    1,11
035306  001300                 07300                 ZERO
035310     001     011         07400                 .BYTE    1,11
035312  001302                 07500                 ONE
035314     003     007         07600                 .BYTE    3,7
035316  001272                 07700                 SAVR5
035320     002     001         07800                 .BYTE    2,1
035322  001266                 07900                 SAVR3
035324  000004                 08000       DT10:     4
035326     001     011         08100                 .BYTE    1,11
035330  001302                 08200                 ONE
035332     001     011         08300                 .BYTE    1,11
035334  001300                 08400                 ZERO
035336     003     007         08500                 .BYTE    3,7
035340  001272                 08600                 SAVR5
035342     002     001         08700                 .BYTE    2,1
035344  001266                 08800                 SAVR3
035346  000002                 08900       DT11:     2
035350     003     007         09000                 .BYTE    3,7
035352  033610                 09100                 FLAG
035354     002     002         09200                 .BYTE    2,2
035356  001266                 09300                 SAVR3
035360  000002                 09400       DT12:     2
035362     006     004         09500                 .BYTE    6,4
035364  033032                 09600                 CALBCC
035366     006     002         09700                 .BYTE    6,2
035370  001252                 09800                 TEMP3
```

```
                                    09900
     035372                         10000   .ERRTAB:
     035372  000000                 10100           0
     035374  000000                 10200           0
     035376  000000                 10300           0
     035400  033564                 10400   EM1
     035402  034670                 10500   DH2     ;HLT    1
     035404  035204                 10600   DT2
     035406  033722                 10700   EM2
     035410  034670                 10800   DH2     ;HLT    2
     035412  035204                 10900   DT2
     035414  033765                 11000   EM3
     035416  034670                 11100   DH2     ;HLT    3
     035420  035204                 11200   DT2
     035422  034031                 11300   EM4
     035424  000000                 11400   0       ;HLT    4
     035426  000000                 11500   0
     035430  034073                 11600   EM5
     035432  034670                 11700   DH2     ;HLT    5
     035434  035204                 11800   DT2
     035436  034073                 11900   EM5
     035440  034726                 12000   DH3     ;HLT    6
     035442  035222                 12100   DT3
     035444  034123                 12200   EM6
     035446  034647                 12300   DH1     ;HLT    7
     035450  035172                 12400   DT1
     035452  034142                 12500   EM7
     035454  034647                 12600   DH1     ;HLT    10
     035456  035172                 12700   DT1
     035460  034167                 12800   EM10
     035462  034647                 12900   DH1     ;HLT    11
     035464  035172                 13000   DT1
     035466  034213                 13100   EM11
     035470  035024                 13200   DH5     ;HLT    12
     035472  035246                 13300   DT5
     035474  034242                 13400   EM12
     035476  035024                 13500   DH5     ;HLT    13
     035500  035246                 13600   DT5
     035502  034213                 13700   EM11
     035504  034764                 13800   DH4     ;HLT    14
     035506  035234                 13900   DT4
     035510  034266                 14000   EM13
     035512  000000                 14100   0       ;HLT    15
     035514  000000                 14200   0
     035516  034213                 14300   EM11
     035520  035024                 14400   DH5     ;HLT    16
     035522  035264                 14500   DT6
     035524  034242                 14600   EM12
     035526  035024                 14700   DH5     ;HLT    17
     035530  035264                 14800   DT6
     035532  034213                 14900   EM11
     035534  035056                 15000   DH6     ;HLT    20
     035536  035302                 15100   DT7
     035540  034213                 15200   EM11
     035542  035056                 15300   DH6     ;HLT    21
     035544  035324                 15400   DT10
```

```
        035546  034242                     15500   EM12
        035550  035056                     15600   DH6        ;HLT      22
        035552  035302                     15700   DT7
        035554  034242                     15800   EM12
        035556  035056                     15900   DH6        ;HLT      23
        035560  035324                     16000   DT10
        035562  034326                     16100   EM14
        035564  000000                     16200   0          ;HLT      24
        035566  000000                     16300   0
        035570  034376                     16400   EM15
        035572  034647                     16500   DH1        ;HLT      25
        035574  035172                     16600   DT1
        035576  034417                     16700   EM16
        035600  034726                     16800   DH3        ;HLT      26
        035602  035346                     16900   DT11
        035604  034242                     17000   EM12
        035606  034647                     17100   DH1        ;HLT      27
        035610  035360                     17200   DT12
        035612  034433                     17300   EM17
        035614  000000                     17400   0          ;HLT      30
        035616  000000                     17500   0
        035620  034477                     17600   EM20
        035622  034647                     17700   DH1        ;HLT      31
        035624  035172                     17800   DT1
        035626  034520                     17900   EM21
        035630  035124                     18000   DH7        ;HLT      32
        035632  000000                     18100   0
        035634  034520                     18200   EM21
        035636  034726                     18300   DH3        ;HLT      33
        035640  035222                     18400   DT3
        035642  034535                     18500   EM22
        035644  035147                     18600   DH10       ;HLT      34
        035646  000000                     18700   0
        035650  034560                     18800   EM23
        035652  034670                     18900   DH2        ;HLT      35
        035654  035204                     19000   DT2
        035656  034602                     19100   EM24
        035660  000000                     19200   0          ;HLT      36
        035662  000000                     19300   0
        035664  034625                     19400   EM25
        035666  000000                     19500   0          ;HLT      37
        035670  000000                     19600   0
        035672  034123                     19700   EM6
        035674  034670                     19800   DH2        ;HLT      40
        035676  035204                     19900   DT2
        035700  034073                     20000   EM5
        035702  035024                     20100   DH5        ;HLT      41
        035704  035246                     20200   DT5
        035706  034266                     20300   EM13
        035710  034647                     20400   DH1        ;HLT      42
        035712  035172                     20500   DT1
                                           20600
                                           20700
        035714                             20800   CORMAX:
                000001                     21300   .END
```

```
DZDMF   MACY11 30(1046)  11-JUL-77  11:59  PAGE 106
DZDME.P11    12-MAY-77 14:18           CROSS REFERENCE TABLE -- USER SYMBOLS

ADRCNT= 004373             879*    915*    924#
AJDONE  003024             569     608     613     659#
ALSTRT  002446             568#    663
AUTO.S  010512             526    1368#
BCOLD   033034            3424    3454    3506    3542    3594    3624    3676    3706    5224#
BINWRD  004714             965*    968*    969    1006#
BITCON  033612            2205*   2279*   2353*   2454*   2460*   2465*   2469    2471*   2504*   2537*   2554*   2621*   3195*
                          3232*   3416*   3498*   3509*   3586*   3668*   3748*   3831*   3919*   4121*   4435*   5156*   5360*
                          5362*   5363    5365*   5383*   5385*   5386    5388*   5413#
BIT0  = 000001              95#   1155    1156
BIT1  = 000002              94#    531    1149    1155    1156    1449    1532    1545
BIT10 = 002000              85#   1513    1524
BIT11 = 004000              84#   2057    2094    2142    2206    2280    2354    2424    2505    2585    2672    2746    2805
                          2837    2869    2909    2943    3000    3046    3092    3138    3189    3280    3371    3417    3499
                          3587    3669    3749    3832    3923    4055    4125    4439    4764    4832    4934
BIT12 = 010000              53#   1464    1543
BIT13 = 020000              82#   1467    1516    1547    1756    3320    3346    3367
BIT14 = 040000              81#    781    1478    1480    1547    1558    1758    3322    3369    4826    4928
BIT15 = 100000              80#    485     572     575     596     599    1451    1519
BIT2  = 000004              93#    531     712    1156
BIT3  = 000010              92#   1549    1556    3348
BIT4  = 000020              91#   1139    1162    1164    2178    2957    3014    3060    3106    3152    5022    5064    5180
BIT5  = 000040              90#   1661    1761    2117    2254    2328    2402    2559    3237    4781
BIT6  = 000100              89#   1144    1145    1551    2846    2886    2918
BIT7  = 000200              88#   1145    1265    1661    2150    2962    5343
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             719     804#
BRX     003732             719     805#
CALBCC  033032            3443    3473    3531    3561    3613    3643    3695    3725    3798    3884    3951    3976*   3977
                          4158    4183*   4184    4239    4264*   4265    4364    4407    4473    4498*   4499    4572    4597*
                          4598    4697    4740    4844    4847*   4904    4946    4949*   4994    5196*   5197    5199*   5205
                          5206*   5207*   5211    5218#   5431
CHAR    032230            3281    5072#
CHARSD  032342            2949    3006    3052    3098    3144    5099#
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#
CLRIC   033374            2056    2093    2141    2204    2278    2352    2423    2503    2583    2671    2745    2904    2936
                          2868    2908    2941    2998    3044    3090    3136    3182    3229    3279    3415    3419    3448
                          3497    3500    3536    3585    3588    3618    3667    3670    3700    3747    3830    3918    4054
                          4120    4434    4763    4831    4933    5339#
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  035714         5431#
CRAM    006606         1187#   1430
CRC.CC= 102010         3420    3450    3502    3538    3590    3620    3672    3702    3755    3838    3945    4152    4233
                       4467    4566    4839    4936    5221#
CRC16 = 120001         5220#
CREAM   001320          195#    483*   1289*   1290    1292*   1296
CSR     006510         1187#   1395
CSRMAP  010514         1370#
CTSDLY  033564         5399#
CYCLE   010060          720     761     762    1280#
DATABP  005216         1049*   1052    1074    1077#
DATACL= 104415          241#   2101    2149    2153    2221    2223    2237    2251    2295    2297    2311    2325    2369
                       2371    2385    2399    2437    2439    2457    2472    2521    2523    2538    2556    2605    2607
                       2625    2689    2692    2710    2719    2762    2763    2772    2809    2841    2881    2913    2951
                       3008    3054    3100    3146    3191    3192    3241    3242    3286    3426    3427    3456    3457
                       3508    3511    3544    3545    3596    3597    3626    3627    3678    3679    3708    3709    3769
                       3773    3852    3857    3937    3952    3979    4002    4019    4026    4028    4066    4144    4159
                       4186    4209    4240    4267    4290    4307    4314    4316    4327    4459    4474    4501    4524
                       4542    4573    4600    4623    4640    4647    4649    4660    4773    4778    4784    5366    5389
DATAHD  005204         1048*   1070    1073#
DELAY = 104413          237#
DEVADR  004370          877*    912     922#
DEVTAB  003010          583     648#
DH1     034647         5431#
DH10    035147         5431#
DH2     034670         5431#
DH3     034726         5431#
DH4     034764         5431#
DH5     035024         5431#
DH6     035056         5431#
DH7     035124         5431#
DISPLA  001200          142#    498*    504*    735*
DISPRE  000174          128#    504
DMACTV  001306          189#    521     676*    677    1280    1294    1376*   1577*   1533*   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#
DMCSP   001404          262#    568*    602     607*    612*    647     765     802    1094    1117    1163    1298*   1307
                       1356    1678*
DMCSPH  001406          263#   1144*   1145*   1149*   1155*   1156*   1162*   1164*   1307*   1309*   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     5009     5022     5064     5180     5258*
                            5268*    5269*
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     035172          5431*
DT10    035324          5431*
DT11    035346          5431*
DT12    035360          5431*
DT2     035204          5431*
DT3     035222          5431*
DT4     035234          5431*
DT5     035246          5431*
DT6     035264          5431*
DT7     035302          5431*
EM1     033664          5431*
EM10    034167          5431*
EM11    034213          5431*
EM12    034242          5431*
EM13    034266          5431*
EM14    034326          5431*
EM15    034376          5431*
EM16    034417          5431*
EM17    034433          5431*
EM2     033722          5431*
EM20    034477          5431*
EM21    034520          5431*
EM22    034535          5431*
EM23    034560          5431*
EM24    034602          5431*
EM25    034625          5431*
EM3     033765          5431*
EM4     034031          5431*
EM5     034073          5431*
EM6     034123          5431*
EM7     034142          5431*
EOM     033306          3933    3934    4063    4064    4135    4136    4140    4141    4449    4450    4455    4456    4851
                        4852    4856    4857    4861    4862    4953    4954    5307*
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*
```

```
DZDMF   MACY11 30 1046)  11-JUL-77  11:59  PAGE 110
DZDME.P11     12-MAY-77 14:18            CROSS REFERENCE TABLE -- USER SYMBOLS

ERTABO  005322           1062    1097*
EXIT  = 000205             96*
EXITER  005252           1082    1087*
FLAG    033610           5412*   5431
FLOAT   002536            585*    591
FLTDAT  033620           4838    4849    4854    4859    4867    4907    5416*
FY      002566            594*    605     609     615
GETQI   033160           3464    3468    3552    3556    3634    3638    3716    3720    3866    3870    5262*
GETQO   033146           3434    3438    3521    3525    3604    3608    3686    3690    3782    3786    5252*
GETSI   03[illegible]12  2226    2230    2240    2244    2300    2304    2314    2318    2374    2378    2388    2392    2442
                         2446    2461    2466    2473    2526    2530    2541    2545    2610    2614    2628    2632    2695
                         2699    2711    2720    2764    2775    2779    3955    3959    3982    3986    4005    4009    4020
                         4029    4162    4166    4189    4193    4212    4216    4243    4247    4270    4274    4293    4297
                         4308    4317    4477    4481    4504    4508    4527    4531    4545    4549    4576    4580    4603
                         4607    4626    4630    4641    4650    5003*   5390
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   032652           3193    3243    3287    3295    4067    4080    4085    4330    4344    4349    4373    4387    4392
                         4663    4677    4682    4706    4720    4725    4785    4795    4868    4884    4889    4959    4974
                         4979    5168*
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*
                         3409*   3447*   3491*   3535*   3579*   3617*   3661*   3699*
LOKFLG  001326            203*
LOLIM   004364            875*    905     920*
LPCNT   001224            160*    792*    793     796*
LRC8  = 000200           5219*
LSTERR  001234            164*    490*    732*   1034    1036*   1124*
LUTYPE= 000000              1    5188
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  033614           3380    3924    3931    4056    4061    4126    4133    4138    4224    4329    4372    4440    4447
                         4453    4557    4662    4705    4765    4770    4775    4940    4951    4958    5414*
MESLD   033332           2431    3379    3930    4060    4132    4137    4446    4452    4769    4774    4848    4853    4858
                         4950    5320*
MI[illegible]  001322     196*    484*    745    1288*   1293*   1297
```

```
MLOCK   006043         714    1187#
MNEW    006144         669    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    2089
                      2137    2200    2274    2348    2419    2499    2579    2667    2741    2800    2832    2864    2904
                      2937    2994    3040    3086    3132    3178    3225    3275    3319    3366    3411    3493    3581
                      3663    3743    3826    3914    4050    4116    4430    4759    4825    4927
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*   2087*   2135*   2198*   2272*   2346*   2417*   2497*   2577*   2663*   2739*   2798*
                      2830*   2862*   2902*   2935*   2992*   3038*   3084*   3130*   3176*   3223*   3273*   3317*   3364*
                      3408*   3490*   3578*   3660*   3741*   3824*   3912*   4048*   4114*   4428*   4757*   4823*   4925*
NITCH   032042        5009*   5010*   5011*   5012*   5014#
NOACT   007154         523    1187#   1282
NODEV   002674         577     616#
NUM     006450        1187#   1382
OCOR    032044        2100    2148    2218    2292    2366    2518    2604    2687    2760    2880    3768    3851    3935
                      4065    4142    4457    4772    4777    4863    4955    5017#   5023    5053    5095    5111    5152
                      5248    5334
OK      002646         603     610#    639
ONE     001302         187#   5431
OUTRDY  032176        2207    2214    2281    2288    2355    2362    2425    2506    2514    2589    2595    2600    2644
                      2675    2683    2747    2871    3283    3929    4059    4131    4445    4768    5037    5046    5057#
                      5082    5091    5104    5108    5128    5137    5235    5244    5281    5297    5311    5329
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
PRIC    006547            1187#   1413
PS    = 177776              63#    476*    711*   1617*   1627*
PUSHR0= 010046              71#   1083
PUSH1S= 005746              69#
PUSH2S= 024646              73#
QV.FLG  001327             204#    482*    750*    790
RESREG  005220            1075    1078#
RESTAR  005350            1108    1114#
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    2054    2059    2061    2065    2091    2096    2098    2103    2113    2139
                          2144    2145    2151    2155    2165    2175    2202    2209    2211    2216    2252    2276    2283
                          2285    2290    2326    2350    2357    2359    2364    2400    2421    2427    2429    2501    2508
                          2510    2516    2557    2581    2591    2593    2597    2602    2646    2669    2677    2679    2685
                          2743    2749    2751    2754    2756    2758    2802    2810    2834    2642    2866    2873    2875
                          2878    2882    2906    2914    2939    2945    2947    2953    2963    2965    2996    3002    3004
                          3010    3019    3042    3048    3050    3056    3065    3088    3094    3096    3102    3111    3134
                          3140    3142    3148    3157    3180    3194    3227    3238    3244    3277    3284    3288    3296
                          3326    3329    3338    3341    3373    3377    3385    3413    3495    3583    3665    3745    3758
                          3762    3766    3807    3828    3841    3845    3849    3893    3916    4052    4068    4081    4086
                          4095    4118    4331    4345    4350    4359    4374    4388    4393    4402    4432    4664    4678
                          4683    4692    4707    4721    4726    4735    4761    4779    4786    4796    4829    4869    4885
                          4890    4899    4931    4960    4975    4980    4989    5007    5020    5035    5039    5042    5048
                          5051    5062    5080    5084    5087    5093    5106    5109    5126    5130    5133    5139    5159
                          5161    5178    5233    5237    5240    5246    5256    5266    5279    5283    5296    5299    5302
                          5313    5315    5330    5344    5346
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    5431
SAVR1   001262             179#    628*    937*    944
SAVR2   001264             180#    936*    945    5431
SAVR3   001266             181#    935*    946    5431
SAVR4   001270             182#    934*    947    5431
SAVR5   001272             183#    933*    948    5431
SAVSP   001274             184#
SAVOS = 104406             227#   1038
SCHAR   032646            3183*   3184*   3230*   3231*   5120    5138    5143    5147*   5148    5154*   5164#
SCOPE = 104400             215#   1688    1711    1735    1766    1808    1850    1908    1960    1982    2004    2037    2074
                          2122    2183    2257    2331    2405    2482    2562    2651    2726    2786    2918    2950    2890
                          2922    2980    3026    3072    3118    3164    3209    3259    3304    3352    3395    3477    3565
                          3647    3729    3812    3898    4035    4097    4410    4743    4807    4910    4997
SCOP1 = 104401             217#   1794    1807    1836    1849    1882    1903    1937    1955    3446    3476    3534    3564
                          3616    3646    3698    3728
SIMOLD  032406            3190    3206    3240    3256    5115#
SIMBCC  032706            3429    3459    3516    3547    3599    3629    3681    3711    3777    3951    3947    4154    4235
                          4469    4568    4840    4942    5188#
SKIP    002632             573     576     597     600     606#
SOFTSW  010052            1229    1268#
SOM     033256            4451    5293#
```

```
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    3320    3322    3346    3367    3369    4826    4928
STAT2   001370           252#   1302#   1978     2000
STAT3   001372           253#   1303#
STFFCK  033474          2549    2636    3512     3774    3858    3963    3990    4170    4197    4251    4278    4485    4512
                        4584    4611    5373#
STFFCL  033414          5144    5351#
STRTSW  001236           169#    513#    516#    517     519     529     531     666     712     721     1323    1347#   1377
                        1601
STUFDT  033640          2432    2434    5422#
STUFLG  032650          3186#   3233#   5141     5155#   5165#
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    032062          2807    2839    2911     5027#
SYNLD   033176          3756    3839    3928     4058    4130    4444    4767    4833    4935    5273#
TEMP    001416           271#    971    1118#    1119#   1160#   1165#   1171#   1183#   2019#   2025#   2029#   2034#
TEMP1   001246           173#    537#   1189     1613#   1614#   5032#   5044#   5078#   5089#   5124#   5135#   5176#   5182#
                        5194#   5208#   5210#    5213#   5214#   5231#   5242#   5277#   5288#   5327#   5332#
TEMP2   001250           174#    538#   1191     4361#   4362#   4363    4404#   4405#   4406    4694#   4695#   4696    4737#
                        4738#   4739    4901#    4902#   4903    4991#   4992#   4993    5076#   5092    5102#   5105    5119#
                        5123#   5150#   5157#    5195#   5200#   5229#   5245
TEMP3   001252           175#    540#    566#    618     627#    1193    1386    1389    1501#   2222#   2224#   2296#   2298#
                        2370#   2372#   2438#    2440#   2522#   2524#   2606#   2608#   2691#   2693#   2771#   2773#   4001#
                        4003#   4083#   4084#    4208#   4210#   4289#   4291#   4347#   4348#   4363#   4364    4390#   4391#
                        4406#   4407    4523#    4525#   4541#   4543#   4622#   4624#   4680#   4681#   4696#   4697    4723#
                        4724#   4739#   4740     4887#   4888#   4903#   4904    4977#   4978#   4993#   4994    5357#   5367#
                        5380#   5393#   5431
TEMP4   001254           176#    541#   1195     1399    1402    1408    1411    1487    1490    1496    1499
TEMP5   001256           177#    542#   1197     1380#   1393#   1599    5060#   5066#
TIMER = 104416           243#   2063    2952     3009    3055    3101    3147    3328    3340    3375
TKCSR   001204           148#    848    1214     1259    1655    5405
TKDBR   001206           149#    350     856     1217    1219    1261    1657
```

```
TLAST = 031450          1350     4998#
TPCSR   001210           150#     832      854      1029     1262     1658
TPDBR   001212           151#     834#     856#     1031#    1264#    1660#
TRPJ#   004730          1017#
TSTNC   001226           161#     491#    1102     1130     1334     1341     1343     1675#    1699#    1722#    1746#    1777#    1819#
                        1861#    1919#    1971#    1993#    2015#    2049#    2086#    2134#    2197#    2271#    2345#    2416#    2496#
                        2576#    2664#    2738#    2797#    2829#    2861#    2901#    2934#    2991#    3037#    3083#    3129#    3175#
                        3222#    3272#    3316#    3363#    3407#    3489#    3577#    3659#    3740#    3823#    3911#    4047#    4113#
                        4427#    4756#    4822#    4924#
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   014022          2050     2086#
TST16   014174          2087     2134#
TST17   014406          2135     2197#
TST2    012374          1676     1699#
TST20   014670          2198     2271#
TST21   015152          2272     2345#
TST22   015434          2346     2416#
TST23   015744          2417     2496#
TST24   016252          2497     2576#
TST25   016604          2577     2664#
TST26   017072          2665     2738#
TST27   017312          2739     2797#
TST3    012442          1700     1722#
TST30   017414          2798     2829#
TST31   017520          2830     2861#
TST32   017656          2862     2901#
TST33   017762          2902     2934#
TST34   020166          2935     2991#
TST35   020334          2992     3037#
TST36   020504          3038     3083#
TST37   020654          3084     3129#
TST4    012514          1723     1746#
TST40   021024          3130     3175#
TST41   021200          3176     3222#
TST42   021364          3223     3272#
TST43   021526          3273     3316#
TST44   021710          3317     3363#
TST45   022054          3364     3407#
TST46   022420          3408     3489#
TST47   023012          3490     3577#
TST5    012616          1747     1777#
TST50   023356          3578     3659#
TST51   023722          3660     3740#
TST52   024244          3741     3823#
TST53   024602          3824     3911#
TST54   025304          3912     4047#
TST55   025526          4048     4113#
TST56   027126          4114     4427#
TST57   030606          4428     4756#
TST6    012756          1778     1819#
TST60   031026          4757     4822#
```

```
TST61   031450            4823    4924#   4998
TST62 = ****** U          4925
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    1328    1346    1352    1390    1412
                          1426    1429    1436    1439    1446    1452    1461    1468    1475    1590
TYPMSG  005106            1051    1054#
VEC     006526            1187#   1404
VECMAP  012010            1589    1601#
WHICH   012002            1392    1597#
WRDCNT  004710             962*    995*   1003#
WRKO.F  005174            1066    1069#
XBX     005000            1028    1030    1032#
XCSR    003546             738     763#
XERR    003570             744     772#
XHEAD   006224             535    1187#
XLOC    003022             593*    611*    614     645     658#
XPASS   003562             742     769#
XPOLY   033030            3420*   3450*   3502*   3538*   3590*   3620*   3672*   3702*   3755*   3838*   3945*   4152*   4233*
                          4467*   4566*   4834*   4936*   5204    5206    5217#
XSTATQ  007454             544    1187#
XTSTN   005330            1060    1100#
XVEC    003554             740     766#
ZERO    001300             186#   5431
$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#   2075#   2079
                          2082#   2123#   2126    2130#   2184#   2187    2193#   2258#   2261    2267#   2332#   2335    2341#
                          2406#   2409    2412#   2483#   2486    2492#   2563#   2566    2572#   2652#   2655    2660#   2727#
                          2730    2734#   2787#   2790    2793#   2819#   2822    2825#   2851#   2854    2857#   2891#   2894
                          2897#   2923#   2926    2930#   2981#   2984    2987#   3027#   3030    3033#   3073#   3076    3079#
                          3119#   3122    3125#   3165#   3168    3171#   3210#   3213    3218#   3260#   3263    3268#   3305#
                          3308    3312#   3353#   3356    3359#   3396#   3399    3403#   3478#   3481    3485#   3566#   3569
                          3573#   3648#   3651    3655#   3730#   3733    3736#   3813#   3816    3819#   3899#   3902    3907#
                          4036#   4039    4043#   4098#   4101    4109#   4411#   4414    4423#   4744#   4747    4752#   4808#
                          4811    4818#   4911#   4914    4920#
$ENDAD  003522             123     509     755#   1079
$N    = 000061               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#   2075    2082    2084    2089    2090#   2123
                          2130    2132    2137    2138#   2184    2193    2195    2200    2201#   2258    2267    2269    2274
                          2275#   2332    2341    2343    2348    2349#   2406    2412    2414    2419    2420#   2483    2492
                          2494    2499    2500#   2563    2572    2574    2579    2580#   2652    2660    2662    2667    2668#
                          2727    2734    2736    2741    2742#   2787    2793    2795    2800    2801#   2819    2825    2827
                          2832    2833#   2851    2857    2859    2864    2865#   2891    2897    2899    2904    2905#   2923
                          2930    2932    2937    2938#   2981    2987    2989    2994    2995#   3027    3033    3035    3040
                          3041#   3073    3079    3081    3086    3087#   3119    3125    3127    3132    3133#   3165    3171
                          3173    3178    3179#   3210    3218    3220    3225    3226#   3260    3268    3270    3275    3276#
                          3305    3312    3314    3319    3320#   3353    3359    3361    3366    3367#   3396    3403    3405
```

```
                                3411     3412#    3478     3485     3487     3493     3494#    3566     3573     3575     3581     3582#    3648
                                3655     3657     3663     3664#    3730     3736     3738     3743     3744#    3813     3919     3821     3826
                                3827#    3899     3907     3909     3914     3915#    4036     4043     4045     4050     4051#    4098     4109
                                4111     4116     4117#    4411     4423     4425     4430     4431#    4744     4752     4754     4759     4760#
                                4808     4818     4820     4825     4826#    4911     4920     4922     4927     4928#    4998#
$S      = 000063                   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#    2087
                                2090#    2135     2138#    2198     2201#    2272     2275#    2346     2349#    2417     2420#    2497     2500#
                                2577     2580#    2665     2668#    2739     2742#    2798     2801#    2830     2833#    2862     2865#    2902
                                2905#    2935     2938#    2992     2995#    3038     3041#    3084     3087#    3130     3133#    3176     3179#
                                3223     3226#    3273     3276#    3317     3320#    3364     3367#    3408     3412#    3490     3494#    3578
                                3582#    3660     3664#    3741     3744#    3824     3827#    3912     3915#    4048     4051#    4114     4117#
                                4428     4431#    4757     4760#    4823     4826#    4925     4928#
$.      = 000017                   1#     207#     215      217#     219#     221#     223#     225#     227#     229#     231#     233#     235#
                                 237#     239#     241#     243#     245#
.       = 035714                 108#     109      112#     119#     124#     127#     131#     135#     137#     189#     190#     191#     192#
                                 272#     277#     279#     280#     281#     282#     284#     295#     286#     287#     289#     290#     291#
                                 292#     294#     295#     296#     297#     299#     300#     301#     302#     304#     305#     306#     307#
                                 309#     310#     311#     312#     314#     315#     316#     317#     319#     320#     321#     322#     324#
                                 325#     326#     327#     329#     330#     331#     332#     334#     335#     336#     337#     339#     340#
                                 341#     342#     344#     345#     346#     347#     349#     350#     351#     352#     354#     355#     356#
                                 357#     515      525      657#     675      1110     1120     1168#    1203#    1205#    1260     1263     1294
                                1378     1422     1542     1546     1593     1624     1656     1659     2765     3321     3347     3368     4030
                                4318     4651
.BEGIN   003152                  690#
.CNVRT   004472                  234      955#
.CONVR   004466                  232      954#
.DATAC   005552                  242     1159#
.DELAY   005436                  238     1132#
.EOP     003364                  731#    4925
.ERRTA   035372                 1046     5431#
.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   005236                  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#
```

```
DZDMF   MACY11 30(1046)  11-JUL-77  11:59  PAGE 118                                        PAGE: 0136
DZDME.P11    12-MAY-77 14:18          CROSS REFERENCE TABLE -- MACRO NAMES

DMEND       1#     725
DMFRNT      1#
HLT        75#    1687    1710    1734    1765    1793    1806    1835    1848    1881    1902    1936    1954    1981    2003
          2027    2036    2072    2110    2120    2162    2172    2181    2228    2232    2242    2246    2256    2302    2306
          2316    2320    2330    2376    2380    2390    2394    2404    2444    2448    2463    2468    2475    2528    2532
          2543    2547    2561    2612    2616    2630    2634    2697    2701    2713    2722    2766    2777    2781    2817
          2849    2889    2921    2960    2972    2979    3017    3025    3063    3071    3109    3117    3155    3163    3200
          3250    3294    3303    3336    3351    3394    3436    3440    3466    3470    3523    3527    3554    3558    3606
          3610    3636    3640    3688    3692    3718    3722    3784    3788    3868    3872    3957    3961    3984    3988
          4007    4011    4022    4031    4074    4093    4164    4168    4191    4195    4214    4218    4245    4249    4272
          4276    4295    4299    4310    4319    4337    4357    4366    4380    4400    4409    4479    4483    4506    4510
          4529    4533    4547    4551    4578    4582    4605    4609    4628    4632    4643    4652    4670    4690    4699
          4713    4733    4742    4783    4792    4803    4875    4897    4906    4966    4987    4996    5068    5184    5392
$ABORT      1#    3260
$ALTC       1#     547
$BCC        1#    3396    3478    3566    3648
$BINCR      1#    3730    3813
$BINWI      1#    2563
$BUFFE      1#    1199
$CDATA      1#    4808    4911
$CLOCK      1#    2005
$COMP       1#    2884    2955    2967    2974    3012    3021    3058    3067    3104    3113    3150    3159    3290    3331
          3343    4088    4352    4395    4685    4728    4892    4982
$CRC        1#    3418    3448    3500    3536    3588    3619    3670    3700
$CRCSH      1#    3749    3832
$CYCLE      1#    1271
$EMPTY      1#    4744
$EOP        1#     725
$FINI       1#    4998
$FLAG       1#    2222    2296    2370    2438    2522    2606    2691    2771    4001    4209    4289    4523    4541    4622
$FLOAT      1#    1867    1887    1925    1942
$GETPA      1#
$HALF       1#
$HEADE      1#
$INACT      1#    2787    2819    2891
$INIT       1#
$LINE1      1#    1851    1909
$LU1        1#    1665    1689    1712    1736
$LU12       1#    1767
$LU17       1#    1809
$MARHI      1#
$MARK       1#
$MATCH      1#    4077    4340    4383    4673    4716    4880    4970
$MOCK       1#
$MODEM      1#    3305
$MSG        1#    1187
$MULT       1#    2652
$PATTE      1#    3165    3210
$PFAIL      1#    1103
$QCQI       1#    5252    5262
$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    1929
          1839    1841    1871    1873    1892    1894    1928    1930    1946    1948    1975    1997    2021    2030    2054
          2059    2061    2065    2091    2096    2098    2103    2113    2139    2144    2146    2151    2155    2165    2175
          2202    2209    2211    2216    2252    2276    2283    2285    2290    2326    2350    2357    2359    2364    2400
```

```
            2421    2427    2429    2501    2508    2510    2516    2557    2581    2591    2593    2597    2602    2646    2669
            2677    2679    2685    2743    2749    2751    2754    2756    2758    2802    2810    2834    2842    2866    2873
            2875    2878    2882    2906    2914    2939    2945    2947    2953    2963    2965    2996    3002    3004    3010
            3019    3042    3048    3050    3056    3065    3088    3094    3096    3102    3111    3134    3140    3142    3148
            3157    3180    3194    3227    3238    3244    3277    3284    3268    3296    3326    3329    3338    3341    3373
            3377    3384    3413    3495    3583    3665    3745    3758    3762    3766    3807    3828    3841    3845    3849
            3893    3916    4052    4068    4081    4086    4095    4118    4331    4345    4350    4359    4374    4382    4393
            4402    4432    4664    4678    4683    4692    4707    4721    4726    4735    4761    4779    4786    4796    4829
            4869    4885    4890    4899    4931    4960    4975    4980    4989    5007    5020    5035    5039    5042    5049
            5051    5062    5080    5084    5087    5093    5106    5109    5126    5130    5133    5139    5159    5161    5178
            5233    5237    5240    5246    5256    5266    5279    5283    5286    5299    5302    5313    5315    5330    5344
            5346
$RCRC         1#    4036
$REC          1#    2923    2981    3027    3073    3119
$SCOPE        1#     775
$SIMBC        1#    5198
$SINAC        1#    2851
$SOFTC        1#    1207
$STUFF        1#    2406
$SWPAC        1#    1961    1983
$TCHAR        1#    3923    4055    4125    4439
$TCRC         1#    3899    4098    4411
$TRANW        1#    3944    4151    4232    4466    4565
$TRAN1        1#    2038    2075    2123
$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    2084    2132
            2195    2269    2343    2414    2494    2574    2662    2736    2755    2827    2859    2899    2932    2989    3035
            3081    3127    3173    3220    3270    3314    3361    3405    3487    3575    3657    3738    3821    3909    4045
            4111    4425    4754    4820    4922
$VARIA        1#     134
$WINDO        1#    2184    2258    2332    2483
$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    2075    2082    2123    2130    2184
            2193    2258    2267    2332    2341    2406    2412    2483    2492    2563    2572    2652    2660    2727    2734
            2787    2793    2819    2825    2851    2857    2891    2897    2923    2930    2981    2987    3027    3033    3073
            3079    3119    3125    3165    3171    3210    3218    3260    3268    3305    3312    3353    3359    3396    3403
            3478    3485    3566    3573    3648    3655    3730    3736    3813    3819    3899    3907    4036    4043    4098
            4109    4411    4423    4744    4752    4808    4818    4911    4920
$ZEROS        1#    2727


. ABS.  035714     000


 ERRORS DETECTED:  0

 DZDMF,DZDMF/SOL/CRF←IPLUTL,DZDME/EQ:LUTYPE
 RUN-TIME: 17 23 1 SECONDS
 RUN-TIME RATIO: 257/42=6.0
 CORE USED:  32K  (63 PAGES)
```