PDP11

DDCMP MODE LINE UNIT

MD-11-DZKCE-A

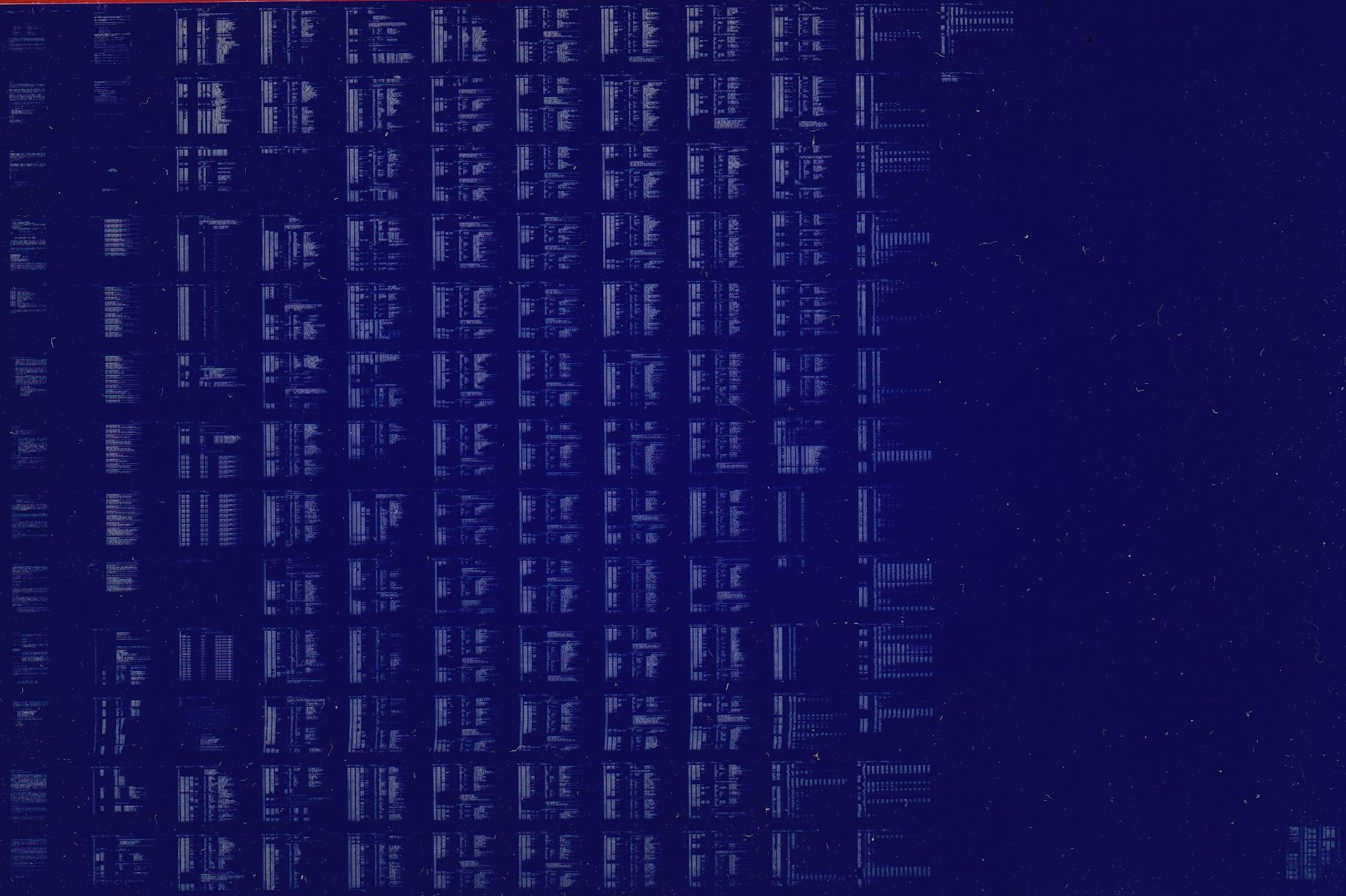
EP-DZKCE-A-DL-A

FICHE 1 OF 1

AUG 1977

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZKCE-A-D
PRODUCT NAME:	DDCMP MODE LINE UNIT TESTS
DATE:	MAY 1977
MAINTAINER:	DIAGNOSTICS
AUTHOR:	DINESH GORADIA

1. ABSTRACT

The function of the KMC11 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 KMC11 are correct in its environment.

Parameters must be set up to alert the diagnostics to the KMC11 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.

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

Currently there are four 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 four diagnostics are:

1. DZKCC [REV] Basic W/R and Micro-processor tests
2. DZKCD [REV] Jump and main memory tests
3. DZKCE [REV] DDCMP Line unit tests
4. DZKCF [REV] BITSTUFF Line unit tests
5. DZKCA [REV] KMC11 CPU MICRO-DIAGNOSTICS

2. REQUIREMENTS

2.1 EQUIPMENT

Any PDP11 family CPU (except an LSI-11) with minimum 8K memory
ASR 33 (or equilivalent)
KMC11-AN IOP (M8204)
KMC11-DA OR KMC11-MD OR KMC11-MA

2.2 STORAGE

Program will use all 8K of memory except where ABL and BOOTSTRAP LOADER reside. Locations 2100 thru 2300; 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 KMC11 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 KMC11 STATUS
                 -------------------

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

        002100 160010 045310  177777  000000
        002110 160020 045320  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. 2100 in the program. In this example the table contains the information and status of two KMC11'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 KMC11'S TO BE TESTED?1

01
CSR ADDRESS?160010
VECTOR ADDRESS?310
BR PRIORITY LEVEL?  (4,5,6,7)?5
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 KMC11'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 KMC11'S DESIRED ACTIVE. Please note that a message is typed out for setting the switch register equal to KMC11's active. this means if the system has four KMC11s; bits 00,01,02,03 will be set in loc 'KMACTV' from the switch register. Using this switch(SW00) alters that location;therefore if four KMC11s 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 KMC11s 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 KMC desired active. EXAMPLE: If you have 4 KMC's but only want to run the first and the last set SWR bits 0 and 3 = 1. PRESS CONTINUE
- E: Number (IF VALID) will be in data lights (excluding 11/05)
- F: Set with any other switch settings desired. PRESS CONTINUE.

4.1.3 DYNAMIC SWITCHES

ERROR SWITCHES

1.	SW 12	Delete print out/bell on error.
2.	SW 13	Delete error printout.
3.	SW 15	Halt on the error.
4.	SW 08	Goto beginning of the test(on error).
5.	SW 10	Goto next test(on error).

SCOPE SWITCHES

1. SW06 Halt in ROMCLK routine before clocking micro-processor instruction. This allows the operator to scope a micro-processor instruction in the static state before it is clocked. Hit continue to resume running.
2. SW09 (if enabled by 'SCOP1') on an error; If an '*' is printed in front of the test no. (ex. *TEST NO. 10) SW09 is incorporated in that test and therefore SW09 is usually the best switch for the scope loop (SW14=0, SW10=0, SW09=1, SW08=0). If SW09 is not enabeled; and there is a HARD error (constant): SW08 is best. (SW14=1,0, SW10=0, SW09=0, SW08=1). for intermittemt errors; SW14=1 will loop on test reguardless of error or not error. (SW14=1, SW10=0, SW09=0, SW08=1,0)
3. SW11 Inhibit interations.
4. SW14 Loop on current test.

4.2 STARTING ADDRESS

Starting address is at 000200 there are no other starting addresses for the KMC11 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 KMC11'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 KMC11 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 '$TSTNM' (address 1202)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 KMC11 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 KMC11 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 KMC 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

KMC11 IOP(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 KMC11 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 KMC11'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 DZKCE CSR:  175000 VEC:  0300 PASSES:  000001
ERRORS:  000000
```

NOTE: The pass count and error counts are cummulitive for each KMC11 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 KMC11 since the diagnostic was started are reflected in PASSES: and ERRORS:.

8.4 KEY LOCATIONS

$lpadr (1206)	Contains the address where program will return when iteration count is reached or if loop on test is asserted.
NEXT (1442)	Contains the address of the next test to be peformed.
$TSTNM (1202)	Containg the number of the test now being peformed.
RUN (1500)	The bit in 'RUN' always points to the KMC11 currently being tested. EXAMPLE: (RUN) 1500/0000000001000000 Means that KMC11 no.06 is the KMC11 now running.
KMCR00-KMCR17 KMST00-KMST17 (2100)-(2300)	These locations contain the information needed to test up to 16 (decimal) KMC11s sequentialy. they contain the CSR,VECTOR and STATUS concerning the configuration of each KMC11.
KMACTV (1470)	Each bit set in this location indicates that the associated KMC11 will be tested in turn. EXAMPLE: (KMACTV) 1470/0000000000011111 means that KMC11 no. 00,01,02,03,04 will be tested. EXAMPLE: (KMACTV) 1470/0000000000010001 Means that KMC11 no. 00,04 will be tested.
KMCSR (2066)	Contains the CSR of the current KMC11 under test.

8.4A 'STATUS TABLE' (2100-2300)

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 KMC11'S. the table can contain up to 16 KMC11'S. Following the map is a description of the bits for each map entry

```
              MAP OF KMC11 STATUS
              -------------------

      PC     CSR   STAT1   STAT2   STAT3
      --     ---   -----   -----   -----
    002100 160010 045310  177777  000000
    002110 160020 016320  000000  000000
```

Each map entry contains 4 words which contain the status information for 1 KMC11. The PC shows where in core memory the first of the 4 words is. In the example above the first KMC'S status is in locations, 2100, 2102, 2104, and 2106. The second KMC status is located at 2110, 2112, 2114, and 2116. The information contained in each 4 word entry is defined as follows:

```
CSR:     Contains KMC11 CSR address

STAT1:   BITS 00-08 IS KMC11 VECTOR ADDRESS
         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 KMC11 BR PRIORITY LEVEL

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

STAT3:   NOT USED
```

8.5 METHOD OF AUTO SIZING

8.5.1 FINDING THE CONTROL STATUS REGISTER.

The auto-sizing routine finds a KMC11 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 CRAM address is written to a 125252 then it is read back. If it contains a -1 or 125252 , if not, the address is updated by 10 and the search continues. A -1 indicates a KMC11 with no CRAM, and a 125252 indicates a KMC11 with CRAM. 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 KMC11's in the system will be found by the auto-sizer. If it does not find a KMC11 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 KMC is programmed to interrupt. The PS is lowered by 1 until the KMC interrupts, a delay is made and if no interupt occures at PS level 3 (because of a bad KMC11) 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 KMC11 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.

****

Note: for ipg's line unit m8202-ye users.

CABLE DATA TEST:[TEST 56 TEST 57]

THESE TESTS WON'T RUN RELIABLY ON LINE UNITS WITHOUT TERMINATING RESISTENCE.

APT/ACT/XXDP/SLIDE

+++++++++++

THIS DIAGNOSTIC IS APT/ACT/XXDP/SLIDE COMPATIBLE USER WOULD BE ABLE TO RUN IT UNDER APT/ACT/XXDP ENVIRONMENT.

NOTE: FOR MANUFACTURING PURPOSE ONLY ITS DESCRIBED HOW TO RUN UNDER APT ENVIRONMENT.

****

ETABLE SETTING FOR APT TO RUN UNDER APT

++++++++++++++++++++

FIRST PASS TIME:

LONGEST TEST TIME:

ADDITIONAL TEST TIME:

ALL THE ABOVE PARAMETERS ARE DEPENDENT ON PARTICULAR DIAGNOSTICS AND SHOULD BE LOADED AT THE TIME OF SETTING ETABLE.THERE IS NO DEFAULT TIME SET UP.

SOFTWARE ENVIRONMENT:001   ENVIRONMENT MODE:200

SWITCH 1:-SHOULD BE USED AS NORMAL SWITCH REGISTER.

SWITCH 2:-NOT USED.

CPU OPTIONS:-NOT USED.

MEMORY TYPE 1:-BITS<2:4>:=BITS <12:14> OF STAT1 OF DEV:0.

```
MAXIMUM ADDRESS:-BITS<17:19>:=BITS<12:14> OF STAT1 OF DEV:1

                 BITS<2:4>:=BITS <12:14> OF STAT1 OF DEV:2

                 BITS<10:12>:=BITS<12:14> OF STAT1 OF DEV:3
```

IN THE SAME MANNER

MEMORY TYPE 2 MAXIMUM ADDRESS:-GETS STAT1<12:14> OF DEVICE 4,5,6,7.

MEMORY TYPE 3 MAXIMUM ADDRESS:-GETS STAT1<12:14> OF DEVICE 8,9,10,11.

MEMORY TYPE 4 MAXIMUM ADDRESS:-GETS STAT1<12:14> OF DEVICE 12,13,14,15.

INTERRUPT VECTOR 1:FIRST DEVICE RECEIVE VECTOR.

```
REST OF THE DEVICE(KMC'S) VECTOR SHOULD BE SET UP SEQUENTIALLY
IN INCREMENTS OF 10.

BUS PRIORITY:KMC'S PRIORITY(SHOULD BE SAME FOR ALL KMC'S UNDER
TEST).

INTERRUPT VECTOR 2:NOT USED.

BUS PRIORITY:NOT USED.

BASE ADDRESS:FIRST DEVICE CSR ADDRESS.

        REST SHOULD FOLLOW SEQUENTIALLY

        IN INCREMENTS OF 10.

DEVICE MAP:AS DESCRIBED IN APT MANUAL.

CONTROLLER SPECIFIC CODE 1:-NO.  OF DEVICES UNDER TEST.

CONTROLLER SPECIFIC CODE 2:-NOT USED.

DEVICE DESCRIPTOR WORD 0:STAT2 OF FIRST DEVICE.

.  .

.  .

TO

.  .

.  .

DEVICE DESCRIPTOR WORD 15:STAT2 OF 16TH DEVICE.(KMC)
```

DOCUMENT
****
MAINDEC-11-DZKCE-A
****

```
2265     ************************** TEST 1 **************************
         OUT CONTROL REGISTER READ/ONLY TEST
         DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
         BITS ARE IN THE CORRECT STATE

2291     ************************** TEST 2 **************************
         IN CONTROL REGISTER READ/ONLY TEST
         DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
         BITS ARE IN THE CORRECT STATE

2316     ************************** TEST 3 **************************
         MODEM CONTROL REGISTER READ/ONLY TEST
         DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
         BITS ARE IN THE CORRECT STATE

2342     ************************** TEST 4 **************************
         MAINTENANCE REGISTER READ/ONLY TEST
         DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
         BITS ARE IN THE CORRECT STATE

2372     ************************** 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

2415     ************************** 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

2458     ************************** TEST 7 **************************
         LINE UNIT REGISTER WRITE/READ TEST
         FLOAT A 1 THROUGH LINE UNIT REGISTER 13
         FLOAT A 0 THROUGH LINE UNIT REGISTER 13

2517     ************************** TEST 10 **************************
         LINE UNIT REGISTER WRITE/READ TEST
         FLOAT A 1 THROUGH LINE UNIT REGISTER 14
         FLOAT A 0 THROUGH LINE UNIT REGISTER 14
```

```
2570       ************************** TEST 11 **************************
           SWITCH PAC TEST
           THIS TEST READS SWITCH PAC#1
           THIS SWITCH PAC CONTAINS THE DDCMP LINE #

2594       ************************** TEST 12 **************************
           SWITCH PAC TEST
           THIS TEST READS SWITCH PAC#2
           THIS SWITCH PAC CONTAINS THE BM873 BOOT ADD

2618       ************************** TEST 13 **************************
           LINE UNIT CLOCK TEST
           THIS TEST VERIFYS THAT THE LU INTERNAL CLOCK
           (BIT 1 IN LU-17) IS WORKING

2653       ************************** 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

2687       ************************** TEST 15 **************************
           DDCMP TEST OF RTS AND OUT ACTIVE
           SET SOM AND LOAD OUT DATA SILO
           SINGLE STEP 2 DATA CLOCKS, VERIFY
           THAT RTS AND ACTIVE ARE SET

2732       ************************** 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

2790       ************************** TEST 17 **************************
           DDCMP TRANSMITTER TEST
           SINGLE CLOCK THE CHARACTER 0
           VERIFY EACH BIT POSITION AS IT
           PASSES THE BIT WINDOW (SI BIT)
           ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE

2843       ************************** TEST 20 **************************
           DDCMP TRANSMITTER TEST
           SINGLE CLOCK THE CHARACTER 125
           VERIFY EACH BIT POSITION AS IT
           PASSES THE BIT WINDOW (SI BIT)
           ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE

2896       ************************** TEST 21 **************************
           DDCMP TRANSMITTER TEST
           SINGLE CLOCK THE CHARACTER 252
           VERIFY EACH BIT POSITION AS IT
           PASSES THE BIT WINDOW (SI BIT)
           ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
```

```
2949    **************************** TEST 22 ****************************
        DDCMP TRANSMITTER TEST
        SINGLE CLOCK THE CHARACTER 377
        VERIFY EACH BIT POSITION AS IT
        PASSES THE BIT WINDOW (SI BIT)
        ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE

3002    **************************** TEST 23 ****************************
        DDCMP TRANSMITTER TEST
        SINGLE CLOCK A BINARY COUNT PATTERN
        VERIFY EACH BIT POSITION AS IT
        PASSES THE BIT WINDOW (SI BIT)
        ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
        AND R5 CONTAINS THE CHARACTER THAT FAILED

3065    **************************** TEST 24 ****************************
        DDCMP STRIP SYNC TEST
        SET LU LOOP, SINGLE STEP 5 SYNCS,
        VERIFY THAT IN ACTIVE DOES NOT SET

3095    **************************** TEST 25 ****************************
        DDCMP IN ACTIVE TEST
        SET LU LOOP, SINGLE STEP 5 SYNCS AND A NON-SYNC (301)
        VERIFY THAT IN ACTIVE IS SET

3125    **************************** TEST 26 ****************************
        DDCMP IN ACTIVE TEST
        SET LU LOOP, SINGLE STEP 1 SYNC AND A NON-SYNC (301)
        VERIFY THAT IN ACTIVE DOES NOT SET

3155    **************************** TEST 27 ****************************
        DDCMP IN ACTIVE TEST
        SET LU LOOP, SINGLE STEP 2 SYNCS AND A NON-SYNC (301)
        VERIFY THAT IN ACTIVE IS SET

3185    **************************** TEST 30 ****************************
        IN CLEAR TEST
        SYNC UP RECEIVER AND TRANSMIT A CHARACTER
        WAIT FOR IN RDY, THEN SET IN CLEAR
        VERIFY THAT IN ACTIVE AND IN RDY ARE CLEARED

3236    **************************** TEST 31 ****************************
        DDCMP BASIC RECEICER TEST
        SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 0
        VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED

3275    **************************** TEST 32 ****************************
        DDCMP BASIC RECEICER TEST
        SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 125
        VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED
```

```
3314    ************************** TEST 33 **************************
        DDCMP BASIC RECEICER TEST
        SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 252
        VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED

3353    ************************** TEST 34 **************************
        DDCMP BASIC RECEICER TEST
        SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 377
        VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED

3392    ************************** TEST 35 **************************
        DDCMP DATA TEST
        THIS TEST SINGLE STEPS A BINARY COUNT PATTERN
        CHECKING EACH CHARACTER AS IT IS RECEIVED

3433    ************************** TEST 36 **************************
        DDCMP DATA TEST
        THIS TEST SINGLE STEPS A BINARY COUNT PATTERN
        CHECKING EACH CHARACTER AS IT IS RECEIVED
        THIS TEST IS EXACTLY THE SAME AS THE LAST TEST,
        EXCEPT LINE UNIT LOOP IS SET IN LU REGISTER 12

3479    ************************** TEST 37 **************************
        TRANSMITTER MARK TEST
        SINGLE CLOCK 3 SYNCS AND A 301 AND 20 EXTRA
        CLOCK TICKS, VERIFY THAT A 301, A 377 AND A 377
        WERE RECEIVED INDICATING THAT THE TRANSMITTER WENT
        TO A MARK STATE FOR 16 BITS WHEN OUT SILO WAS EMPTY

3526    ************************** TEST 40 **************************
        CABLE TURNAROUND TEST
        CLEAR LINE UNIT LOOP, SET DTR
        VERIFY THAT RING AND MODEM READY ARE SET
        CLEAR DTR, VERIFY THAT RING AND MRDY ARE CLEARED

3579    ************************** TEST 41 **************************
        CABLE TURNAROUND TEST
        CLEAR LINE UNIT LOOP, LOAD OUT DATA SILO
        VERIFY THAT ALL MODEM SIGNALS ARE SET

3627    ************************** TEST 42 **************************
        TEST OF CRC OPERATION
        USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
        0, VERIFY THE LSB OF THE BCC ON EACH SHIFT
        TEST TRANSMITTER FIRST THEN THE RECEIVER BCC

3704    ************************** TEST 43 **************************
        TEST OF CRC OPERATION
        USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
        377, VERIFY THE LSB OF THE BCC ON EACH SHIFT
        TEST TRANSMITTER FIRST THEN THE RECEIVER BCC
```

```
3781    ************************** TEST 44 **************************
        TEST OF CRC OPERATION
        USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
        125, VERIFY THE LSB OF THE BCC ON EACH SHIFT
        TEST TRANSMITTER FIRST THEN THE RECEIVER BCC

3858    ************************** TEST 45 **************************
        TEST OF CRC OPERATION
        USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
        252, VERIFY THE LSB OF THE BCC ON EACH SHIFT
        TEST TRANSMITTER FIRST THEN THE RECEIVER BCC

3935    ************************** TEST 46 **************************
        TRANSMITTER CRC TEST
        USING THE CRC16 POLYNOMINAL, SINGLE CLOCK A BINARY
        COUNT PATTERN, VERIFY THE LSB OF THE TRANSMITTER BCC ON EACH SHIFT

4003    ************************** TEST 47 **************************
        RECEIVER CRC TEST
        USING THE CRC16 POLYNOMINAL, SINGLE CLOCK A BINARY
        COUNT PATTERN, VERIFY THE LSB OF THE RECEIVER BCC ON EACH SHIFT

4071    ************************** TEST 50 **************************
        TRANSMITTER DDCMP CRC TEST
        THIS TEST TRANSMITS A FOUR CHARACTER MESSAGE WITH CRC
        BOTH DATA AND THE BCC ARE VERIFIED IN THE BIT
        WINDOW. THE FOUR CHARACTERS ARE 0,125,252,377
        THE TRANSMITTER IS CHECKED FOR GOING TO A MARK STATE AFTER THE BCC

4174    ************************** TEST 51 **************************
        RECEIVER DDCMP CRC TEST
        THIS TEST CLOCKS A FOUR CHARACTER MESSAGE WITH BCC
        AND VERIFYS CORRECT DATA RECEPTION AND BCC MATCH
        THE FOUR CHARACTER MESSAGE IS 0,125,252,377

4233    ************************** TEST 52 **************************
        DDCMP EOM FUNCTION TEST
        THIS TEST LOADS OUT SILO WITH: 2 SYNCS,4 CHAR MESSAGE,EOM
        4 CHARACTER MESS,EOM. THE DATA STREAM IS CHECKED TO BE
        4 CHAR,BCC,4 CHAR,BCC,MARKS. THIS TEST VERIFYS THAT
        THE CHARCTERS LOADED WITH EOM SET ARE LOST
        ALL DATA AND BCC'S ARE CHECKED IN THE BIT WINDOW
        THE FOUR CHARACTER MESSAGE IS 0,125,252,377
        RECEIVED DATA IS VERIFIED, AND IN BCC MATCH IS CHECKED

4481    ************************** TEST 53 **************************
        DDCMP EOM FUNCTION TEST
        THIS TEST LOADS OUT SILO WITH: 2 SYNCS,4 CHAR MESSAGE,EOM
        SOM,4 CHAR MESS,EOM. THE DATA STREAM IS CHECKED TO BE
        4 CHAR,BCC,4 CHAR,BCC,MARKS. THIS TEST VERIFYS THAT
        THE CHARCTERS LOADED WITH EOM SET ARE LOST
        ALSO THAT THE CHAR LOADED WITH SOM IS NOT IN THE BCC
        ALL DATA AND BCC'S ARE CHECKED IN THE BIT WINDOW
        THE FOUR CHARACTER MESSAGE IS 0,125,252,377
```

```
               RECEIVED DATA IS VERIFIED, AND IN BCC MATCH IS CHECKED

       4761    *************************** TEST 54 ***************************
               EMPTY SILO TEST
               LOAD SILO WITH 2 SYNCS, 4 CHAR MESSAGE, SINGLE CLOCK
               UNTIL THE SILO IS EMPTY, LOAD 4 MORE CHARACTERS IN THE
               SILO. GIVE MORE TICKS, AND VERIFY THAT ONLY THE FIRST
               4 CHARACTER MESSAGE WAS RECEIVED AND THAT RTS IS CLEAR

       4827    *************************** TEST 55 ***************************
               HALF DUPLEX TEST
               SET LINE UNIT LOOP AND HALF DUPLEX, SEND SYNCS AND A

       4830    MESSAGE. VERIFY THAT IN-ACTIVE AND IN-READY ARE CLEAR

       4864    *************************** TEST 56 ***************************
               DDCMP CABLE DATA TEST
               THIS TEST LOADS OUT SILO WITH THE FOLLOWING:
               4 SYNCS,16 CHAR,EOM,16 CHAR,EOM,16 CHAR,EOM
               THE 16 CHARACTERS INCLUDE A FLOATING ONE AND ZERO
               THE DATA IS TRANSMITTED OVER THE CABLE USING THE INTERNAL CLOCK
               RECEIVED DATA IS VERIFIED AS IS IN BCC MATCH
               LOOP-BACK CONNECTOR MUST BE ON TO RUN THIS TEST

       4961    *************************** TEST 57 ***************************
               DDCMP CABLE DATA TEST
               THIS TEST LOADS OUT SILO WITH THE FOLLOWING:
               4 SYNCS,59 DATA CHARACTERS,EOM WITH GARBAGE CHARACTER
               THE DATA IS TRANSMITTED OVER THE CABLE USING THE INTERNAL CLOCK
               RECEIVED DATA IS VERIFIED AS IS IN BCC MATCH
               LOOP-BACK CONNECTOR MUST BE ON TO RUN THIS TEST
```

```
                        .TITLE  MAINDEC-11-DZKCE-A
                        ;*COPYRIGHT (C) 1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY DINESH GORADIA
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*




                        ;*MAINDEC-11-DZKCE-A   KMC11 DDCMP LINE UNIT TESTS
                        ;*COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                        ;*----------------------------------------------------------------

                        ;STARTING PROCEDURE
                        ;LOAD PROGRAM
                        ;LOAD ADDRESS 000200
                        ;SWR=0  AUTOSIZE KMC11
                        ;SW07=1  USE CURRENT KMC11 PARAMETERS
                        ;SW00=1  INPUT NEW KMC11 PARAMETERS
                        ;PRESS START
                        ;PROGRAM WILL TYPE "MAINDEC-11-DZKCE-A   KMC11 DDCMP LINE UNIT TESTS"
                        ;PROGRAM WILL TYPE STATUS MAP
                        ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                        ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
                        ;AND THEN RESUME TESTING
                        ;SUBSEQUENT RESTARTS WILL NOT TYPE PROGRAM TITLE

                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1200 ***
        001200          STACK=  1200
                        .EQUIV  EMT,ERROR               ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE               ;;BASIC DEFINITION OF SCOPE CALL

                        ;*MISCELLANEOUS DEFINITIONS
        000011          HT=     11                      ;;CODE FOR HORIZONTAL TAB
        000012          LF=     12                      ;;CODE FOR LINE FEED
        000015          CR=     15                      ;;CODE FOR CARRIAGE RETURN
        000200          CRLF=   200                     ;;CODE FOR CARRIAGE RETURN-LINE FEED
        177776          PS=     177776                  ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
        177774          STKLMT= 177774                  ;;STACK LIMIT REGISTER
        177772          PIRQ=   177772                  ;;PROGRAM INTERRUPT REQUEST REGISTER
        177570          DSWR=   177570                  ;;HARDWARE SWITCH REGISTER
        177570          DDISP=  177570                  ;;HARDWARE DISPLAY REGISTER

                        ;*GENERAL PURPOSE 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          R6=     %6              ;;GENERAL REGISTER
        000007          R7=     %7              ;;GENERAL REGISTER
        000006          SP=     %6              ;;STACK POINTER
        000007          PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
        000000          PR0=    0               ;;PRIORITY LEVEL 0
        000040          PR1=    40              ;;PRIORITY LEVEL 1
        000100          PR2=    100             ;;PRIORITY LEVEL 2
        000140          PR3=    140             ;;PRIORITY LEVEL 3
        000200          PR4=    200             ;;PRIORITY LEVEL 4
        000240          PR5=    240             ;;PRIORITY LEVEL 5
        000300          PR6=    300             ;;PRIORITY LEVEL 6
        000340          PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
        100000          SW15=   100000
        040000          SW14=   40000
        020000          SW13=   20000
        010000          SW12=   10000
        004000          SW11=   4000
        002000          SW10=   2000
        001000          SW09=   1000
        000400          SW08=   400
        000200          SW07=   200
        000100          SW06=   100
        000040          SW05=   40
        000020          SW04=   20
        000010          SW03=   10
        000004          SW02=   4
        000002          SW01=   2
        000001          SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
        100000          BIT15=  100000
        040000          BIT14=  40000
        020000          BIT13=  20000
        010000          BIT12=  10000
        004000          BIT11=  4000
        002000          BIT10=  2000
        001000          BIT09=  1000
        000400          BIT08=  400
        000200          BIT07=  200
        000100          BIT06=  100
        000040          BIT05=  40
        000020          BIT04=  20
        000010          BIT03=  10
        000004          BIT02=  4
        000002          BIT01=  2
        000001          BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
        000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
        000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
        000014          TBITVEC=14              ;;"T" BIT
        000014          TRTVEC= 14              ;;TRACE TRAP
        000014          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
        000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
        000024          PWRVEC= 24              ;;POWER FAIL
        000030          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
        000034          TRAPVEC=34              ;;"TRAP" TRAP
        000060          TKVEC=  60              ;;TTY KEYBOARD VECTOR
        000064          TPVEC=  64              ;;TTY PRINTER VECTOR
        000240          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR




                        ;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
```

```

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

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

        000020                  .=20
000020  004134                          $SCOPE          ; SCOPE LOOP HANDLER.
000022  000340                          PR7             ; SERVICE AT LEVEL 7.
000024  007126                          $PWRDN                  ;POWER FAIL HANDLER
000026  000340                          PR7                     ;SERVICE AT LEVEL 7
000030  006512                          $ERROR                  ;ERROR HANDLER
000032  000340                          PR7                     ;SERVICE AT LEVEL 7
000034  006414                          $TRAP                   ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                          PR7                     ;SERVICE AT LEVEL 7
                                .SBTTL  ACT11 HOOKS

                                ;;*************************************************************
                                ;HOOKS REQUIRED BY ACT11
        000040                          $SVPC=.                 ;SAVE PC
        000046                          .=46
000046  004070                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
        000052                          .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
        000040                          .=$SVPC                 ;; RESTORE PC

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

        000200                  .=200
000200  000137  002402                  JMP     .START          ;GO TO START OF PROGRAM



        001000                  .=1000
001000  005200  040515  047111  MTITLE: .ASCII  <200><12>/MAINDEC-11-DZKCE-A/<200>
001025     113  041515  030461          .ASCIZ  /KMC11 DDCMP LINE UNIT TESTS/<200>

        177570                  DSWR    =       177570
        177570                  DDISP   =       177570
```

```
                                .SBTTL  COMMON TAGS

                                ;;*********************************************************************
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

                001200                   .=1200
        001200                  $CMTAG:                                  ;;START OF COMMON TAGS
        001200  000000                   .WORD   0
        001202     000          $TSTNM:  .BYTE   0                       ;;CONTAINS THE TEST NUMBER
        001203     000          $ERFLG:  .BYTE   0                       ;;CONTAINS ERROR FLAG
        001204  000000          $ICNT:   .WORD   0                       ;;CONTAINS SUBTEST ITERATION COUNT
        001206  000000          $LPADR:  .WORD   0                       ;;CONTAINS SCOPE LOOP ADDRESS
        001210  000000          $LPERR:  .WORD   0                       ;;CONTAINS SCOPE RETURN FOR ERRORS
        001212  000000          $ERTTL:  .WORD   0                       ;;CONTAINS TOTAL ERRORS DETECTED
        001214     000          $ITEMB:  .BYTE   0                       ;;CONTAINS ITEM CONTROL BYTE
        001215     001          $ERMAX:  .BYTE   1                       ;;CONTAINS MAX. ERRORS PER TEST
        001216  000000          $ERRPC:  .WORD   0                       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
        001220  000000          $GDADR:  .WORD   0                       ;;CONTAINS ADDRESS OF 'GOOD' DATA
        001222  000000          $BDADR:  .WORD   0                       ;;CONTAINS ADDRESS OF 'BAD' DATA
        001224  000000          $GDDAT:  .WORD   0                       ;;CONTAINS 'GOOD' DATA
        001226  000000          $BDDAT:  .WORD   0                       ;;CONTAINS 'BAD' DATA
        001230  000000                   .WORD   0                       ;;RESERVED--NOT TO BE USED
        001232  000000                   .WORD   0
        001234     000          $AUTOB:  .BYTE   0                       ;;AUTOMATIC MODE INDICATOR
        001235     000          $INTAG:  .BYTE   0                       ;;INTERRUPT MODE INDICATOR
        001236  000000                   .WORD   0
        001240  177570          SWR:     .WORD   DSWR                    ;;ADDRESS OF SWITCH REGISTER
        001242  177570          DISPLAY: .WORD   DDISP                   ;;ADDRESS OF DISPLAY REGISTER
        001244  177560          $TKS:    177560                          ;;TTY KBD STATUS
        001246  177562          $TKB:    177562                          ;;TTY KBD BUFFER
        001250  177564          $TPS:    177564                          ;;TTY PRINTER STATUS REG. ADDRESS
        001252  177566          $TPB:    177566                          ;;TTY PRINTER BUFFER REG. ADDRESS
        001254     000          $NULL:   .BYTE   0                       ;;CONTAINS NULL CHARACTER FOR FILLS
        001255     002          $FILLS:  .BYTE   2                       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
        001256     012          $FILLC:  .BYTE   12                      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
        001257     000          $TPFLG:  .BYTE   0                       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
        001260  000000          $REGAD:  .WORD   0                       ;;CONTAINS THE ADDRESS FROM
                                                                         ;;WHICH ($REG0) WAS OBTAINED
        001262  000000          $REG0:   .WORD   0                       ;;CONTAINS (($REGAD)+0)
        001264  000000          $REG1:   .WORD   0                       ;;CONTAINS (($REGAD)+2)
        001266  000000          $REG2:   .WORD   0                       ;;CONTAINS (($REGAD)+4)
        001270  000000          $REG3:   .WORD   0                       ;;CONTAINS (($REGAD)+6)
        001272  000000          $REG4:   .WORD   0                       ;;CONTAINS (($REGAD)+10)
        001274  000000          $REG5:   .WORD   0                       ;;CONTAINS (($REGAD)+12)
        001276  000000          $TMP0:   .WORD   0                       ;;USER DEFINED
        001300  000000          $TMP1:   .WORD   0                       ;;USER DEFINED
        001302  000000          $TMP2:   .WORD   0                       ;;USER DEFINED
        001304  000000          $TMP3:   .WORD   0                       ;;USER DEFINED
        001306  000000          $TMP4:   .WORD   0                       ;;USER DEFINED
        001310  000000          $TIMES:  0                               ;;MAX. NUMBER OF ITERATIONS
        001312     077          $QUES:   .ASCII  /?/                     ;;QUESTION MARK
        001313     015          $CRLF:   .ASCII  <15>                    ;;CARRIAGE RETURN
        001314  000012          $LF:     .ASCIZ  <12>                    ;;LINE FEED
                                ;;*********************************************************************
                                .SBTTL  APT MAILBOX-ETABLE
```

```


                           ;;****************************************************************
                           .EVEN
001316                     $MAIL:                            ;;APT MAILBOX
001316  000000             $MSGTY: .WORD    AMSGTY           ;;MESSAGE TYPE CODE
001320  000000             $FATAL: .WORD    AFATAL           ;;FATAL ERROR NUMBER
001322  000000             $TESTN: .WORD    ATESTN           ;;TEST NUMBER
001324  000000             $PASS:  .WORD    APASS            ;;PASS COUNT
001326  000000             $DEVCT: .WORD    ADEVCT           ;;DEVICE COUNT
001330  000000             $UNIT:  .WORD    AUNIT            ;;I/O UNIT NUMBER
001332  000000             $MSGAD: .WORD    AMSGAD           ;;MESSAGE ADDRESS
001334  000000             $MSGLG: .WORD    AMSGLG           ;;MESSAGE LENGTH
001336                     $ETABLE:                          ;;APT ENVIRONMENT TABLE
001336     002             $ENV:   .BYTE    AENV             ;;ENVIRONMENT BYTE
001337     000             $ENVM:  .BYTE    AENVM            ;;ENVIRONMENT MODE BITS
001340  000000             $SWREG: .WORD    ASWREG           ;;APT SWITCH REGISTER
001342  000000             $USWR:  .WORD    AUSWR            ;;USER SWITCHES
001344  000000             $CPUOP: .WORD    ACPUOP           ;;CPU TYPE,OPTIONS
                           ;*                               BITS 15-11=CPU TYPE
                           ;*                                         11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                           ;*                                         11/70=06,PDQ=07,Q=10
                           ;*                               BIT 10=REAL TIME CLOCK
                           ;*                               BIT  9=FLOATING POINT PROCESSOR
                           ;*                               BIT  8=MEMORY MANAGEMENT
001346     000             $MAMS1: .BYTE    AMAMS1           ;;HIGH ADDRESS,M.S. BYTE
001347     000             $MTYP1: .BYTE    AMTYP1           ;;MEM. TYPE,BLK#1
                           ;*                               MEM.TYPE BYTE  --  (HIGH BYTE)
                           ;*                                         900 NSEC CORE=001
                           ;*                                         300 NSEC BIPOLAR=002
                           ;*                                         500 NSEC MOS=003
001350  000000             $MADR1: .WORD    AMADR1           ;;HIGH ADDRESS,BLK#1
                           ;*                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001352     000             $MAMS2: .BYTE    AMAMS2           ;;HIGH ADDRESS,M.S. BYTE
001353     000             $MTYP2: .BYTE    AMTYP2           ;;MEM.TYPE,BLK#2
001354  000000             $MADR2: .WORD    AMADR2           ;;MEM.LAST ADDRESS,BLK#2
001356     000             $MAMS3: .BYTE    AMAMS3           ;;HIGH ADDRESS,M.S.BYTE
001357     000             $MTYP3: .BYTE    AMTYP3           ;;MEM.TYPE,BLK#3
001360  000000             $MADR3: .WORD    AMADR3           ;;MEM.LAST ADDRESS,BLK#3
001362     000             $MAMS4: .BYTE    AMAMS4           ;;HIGH ADDRESS,M.S.BYTE
001363     000             $MTYP4: .BYTE    AMTYP4           ;;MEM.TYPE,BLK#4
001364  000000             $MADR4: .WORD    AMADR4           ;;MEM.LAST ADDRESS,BLK#4
001366  000000             $VECT1: .WORD    AVECT1           ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001370  000000             $VECT2: .WORD    AVECT2           ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001372  000000             $BASE:  .WORD    ABASE            ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001374  000000             $DEVM:  .WORD    ADEVM            ;;DEVICE MAP
001376  000000             $CDW1:  .WORD    ACDW1            ;;CONTROLLER DESCRIPTION WORD#1
001400  000000             $CDW2:  .WORD    ACDW2            ;;CONTROLLER DESCRIPTION WORD#2
001402  000000             $DDW0:  .WORD    ADDW0            ;;DEVICE DESCRIPTOR WORD#0
001404  000000             $DDW1:  .WORD    ADDW1            ;;DEVICE DESCRIPTOR WORD#1
001406  000000             $DDW2:  .WORD    ADDW2            ;;DEVICE DESCRIPTOR WORD#2
001410  000000             $DDW3:  .WORD    ADDW3            ;;DEVICE DESCRIPTOR WORD#3
001412  000000             $DDW4:  .WORD    ADDW4            ;;DEVICE DESCRIPTOR WORD#4
001414  000000             $DDW5:  .WORD    ADDW5            ;;DEVICE DESCRIPTOR WORD#5
001416  000000             $DDW6:  .WORD    ADDW6            ;;DEVICE DESCRIPTOR WORD#6
001420  000000             $DDW7:  .WORD    ADDW7            ;;DEVICE DESCRIPTOR WORD#7
001422  000000             $DDW8:  .WORD    ADDW8            ;;DEVICE DESCRIPTOR WORD#8
```

```
 001424  000000            $DDW9:  .WORD   ADDW9    ;;DEVICE DESCRIPTOR WORD#9
 001426  000000            $DDW10: .WORD   ADDW10   ;;DEVICE DESCRIPTOR WORD#10
 001430  000000            $DDW11: .WORD   ADDW11   ;;DEVICE DESCRIPTOR WORD#11
 001432  000000            $DDW12: .WORD   ADDW12   ;;DEVICE DESCRIPTOR WORD#12
 001434  000000            $DDW13: .WORD   ADDW13   ;;DEVICE DESCRIPTOR WORD#13
 001436  000000            $DDW14: .WORD   ADDW14   ;;DEVICE DESCRIPTOR WORD#14
 001440  000000            $DDW15: .WORD   ADDW15   ;;DEVICE DESCRIPTOR WORD#15


 001442                    $ETEND:

                           ;
                           ;  PROGRAM CONTROL PARAMETERS
                           ;-------------------------------------
 001442  000000            NEXT:   .WORD   0                 ; ADDRSS OF NEXT TEST TO BE EXECUTED
 001444  000000            LOCK:   .WORD   0                 ; ADDRESS FOR LOCK CURRENT DATA
                           ;
                           ;   PROGRAM VARIABLES
                           ;------------------------------
 001446  000000            STRTSW: .WORD   0                 ; SWITCHES AT START OF PROGRAM
 001450  000000            STAT:   .WORD   0                 ; KM STATUS WORD STORAGE
 001452  000000            CLKX:   .WORD   0                 ;
 001454  000000            MASKX:  .WORD   0                 ;
 001456  000000            SAVSP:  .WORD   0                 ; STACK POINTER STORAGE
 001460  000000            SAVPC:  .WORD   0                 ; PROGRAM COUNTER STORAGE
 001462  000000            ZERO:   .WORD   0                 ;
 001464  000001            ONE:    .WORD   1                 ;
 001466  000000            MEMLIM: .WORD   0                 ; HIGHEST LOCATION FOR NPR'S
 001470  000001            KMACTV: .BLKW   1                 ; KMC11 SELECTED ACTIVE
 001472  000001            KMNUM:  .BLKW   1                 ; OCTAL NUMBER OF KMC11'S
 001474  000001            SAVACT: .BLKW   1                 ; ORIGINAL ACTIVE DEVICES.
 001476  000001            SAVNUM: .BLKW   1                 ; WORKABLE NUMBER.
 001500  000000            RUN:    .WORD   0                 ; POINTER TO RUNNING DEVICES
                                   .EVEN
 001502  002072            CREAM:  .WORD   KM.MAP-6          ; TABLE POINTER
 001504  002276            MILK:   .WORD   CNT.MAP-4         ; TABLE POINTER
                           ;
                           ;  PROGRAM CONTROL FLAGS
                           ;-------------------------------
 001506     000            INIFLG: .BYTE   0                 ; PROGRAM INITIALIZING FLAG
         001510                    .EVEN
 001510     000            LOKFLG: .BYTE   0                 ; LOCK ON CURRENT TEST FLAG
 001511     000            QV.FLG: .BYTE   0                 ; QUICK VERIFY FLAG
                                                             ; ON FIRST PASS OF EACH KMC11 ITERATIONS WILL BE SUPPRES
                                   .EVEN
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT


 001512                         $ERRTB:
                                .EVEN
                                ;*      DF              ;; DOES NOT APPLY IN THIS DIAGNOSTIC.
 001512 000000                          0
 001514 000000                          0
 001516 000000                          0
 001520 031610                          EM1
 001522 032614                          DH2             ; ERROR 1
 001524 033130                          DT2
 001526 031646                          EM2
 001530 032614                          DH2             ; ERROR 2
 001532 033130                          DT2
 001534 031711                          EM3
 001536 032614                          DH2             ; ERROR 3
 001540 033130                          DT2
 001542 031755                          EM4
 001544 000000                          0               ; ERROR 4
 001546 000000                          0
 001550 032017                          EM5
 001552 032614                          DH2             ; ERROR 5
 001554 033130                          DT2
 001556 032017                          EM5
 001560 032652                          DH3             ; ERROR 6
 001562 033146                          DT3
 001564 032047                          EM6
 001566 032573                          DH1             ; ERROR 7
 001570 033116                          DT1
 001572 032066                          EM7
 001574 032573                          DH1             ; ERROR 10
 001576 033116                          DT1
 001600 032113                          EM10
 001602 032573                          DH1             ; ERROR 11
 001604 033116                          DT1
 001606 032137                          EM11
 001610 032750                          DH5             ; ERROR 12
 001612 033172                          DT5
 001614 032166                          EM12
 001616 032750                          DH5             ; ERROR 13
 001620 033172                          DT5
 001622 032137                          EM11
 001624 032710                          DH4             ; ERROR 14
 001626 033160                          DT4
```

```
DZKCE    MACY11 27(1006)  01-JUN-77  10:03  PAGE 10
DZKCE.P11     12-MAY-77 12:23               ERROR POINTER TABLE

   421  001630  032212                        EM13
   422  001632  000000                        0                   ; ERROR 15
   423  001634  000000                        0
   424  001636  032137                        EM11
   425  001640  032750                        DH5                 ; ERROR 16
   426  001642  033210                        DT6
   427  001644  032166                        EM12
   428  001646  032750                        DH5                 ; ERROR 17
   429  001650  033210                        DT6
   430  001652  032137                        EM11
   431  001654  033002                        DH6                 ; ERROR 20
   432  001656  033226                        DT7
   433  001660  032137                        EM11
   434  001662  033002                        DH6                 ; ERROR 21
   435  001664  033250                        DT10
   436  001666  032166                        EM12
   437  001670  033002                        DH6                 ; ERROR 22
   438  001672  033226                        DT7
   439  001674  032166                        EM12
   440  001676  033002                        DH6                 ; ERROR 23
   441  001700  033250                        DT10
   442  001702  032252                        EM14
   443  001704  000000                        0                   ; ERROR 24
   444  001706  000000                        0
   445  001710  032322                        EM15
   446  001712  032573                        DH1                 ; ERROR 25
   447  001714  033116                        DT1
   448  001716  032343                        EM16
   449  001720  032652                        DH3                 ; ERROR 16
   450  001722  033272                        DT11
   451  001724  032166                        EM12
   452  001726  032573                        DH1                 ; ERROR 27
   453  001730  033304                        DT12
   454  001732  032357                        EM17
   455  001734  000000                        0                   ; ERROR 30
   456  001736  000000                        0
   457  001740  032423                        EM20
   458  001742  032573                        DH1                 ; ERROR 31
   459  001744  033116                        DT1
   460  001746  032444                        EM21
   461  001750  033050                        DH7                 ; ERROR 32
   462  001752  000000                        0
   463  001754  032444                        EM21
   464  001756  032652                        DH3                 ; ERROR 33
   465  001760  033146                        DT3
   466  001762  032461                        EM22
   467  001764  033073                        DH10                ; ERROR 34
   468  001766  000000                        0
   469  001770  032504                        EM23
   470  001772  032614                        DH2                 ; ERROR 35
   471  001774  033130                        DT2
   472  001776  032526                        EM24
   473  002000  000000                        0                   ; ERROR 36
   474  002002  000000                        0
   475  002004  032551                        EM25
   476  002006  000000                        0                   ; ERROR 37
```

```
002010  000000                  0
002012  032047                  EM6
002014  032614                  DH2             ; ERROR 40
002016  033130                  DT2
002020  032017                  EM5
002022  032750                  DH5             ; ERROR 41
002024  033172                  DT5
002026  032212                  EM13
002030  032573                  DH1             ; ERROR 42
002032  033116                  DT1
        002034          .=2034
                        .SBTTL  APT PARAMETER BLOCK

                        ;;*****************************************************
                        ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                        ;;*****************************************************
        002034                  .$X=.   ;;SAVE CURRENT LOCATION
        000024                  .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                  200     ;;FOR APT START UP
        000044                  .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  002034                  $APTHDR ;;POINT TO APT HEADER BLOCK
        002034                  .=.$X   ;;RESET LOCATION COUNTER
                        ;;*****************************************************
                        ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                        ;INTERFACE SPEC.

002034                  $APTHD:
002034  000000          $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
002036  001316          $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
002040  000132          $TSTM:  .WORD   90.     ;;RUN TIM OF LONGEST TEST
002042  000137          $PASTM: .WORD   95.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
002044  000137          $UNITM: .WORD   95.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
002046  000052                  .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```

                                ;KMC11 CONTROL INDICATORS FOR CURRENT KMC11 UNDER TEST
                                ;-------------------------------------------------------------

002050  000000                  STAT1:  0
002052  000000                  STAT2:  0
002054  000000                  STAT3:  0

                                ;KMC11 VECTOR AND REGISTER INDIRECT POINTERS
                                ;------------------------------------------------

002056  000000                  KMRVEC: 0               ;POINTER TO KMC11 RECEIVER INTERRUPT VECTOR
002060  000000                  KMRLVL: 0               ;POINTER TO KMC11 RECEIVER INTERRUPT SERVICE PS
002062  000000                  KMTVEC: 0               ;POINTER TO KMC11 TRANSMITTER INTERRUPT VECTOR
002064  000000                  KMTLVL: 0               ;POINTER TO KMC11 TRANSMITTER INTERRUPT SERVICE PS
002066  000000                  KMCSR:  0               ;POINTER TO KMC11 CONTROL STATUS REGISTER
002070  000000                  KMCSRH: 0               ;POINTER TO KMC11 CONTROL STATUS REGISTER HIGH BYTE.
002072  000000                  KMCTL:  0               ;POINTER TO KMC11 CONTOL OUT REGISTER
002074  000000                  KMP04:  0               ;POINTER TO KMC11 PORT REGISTER(SEL 4)
002076  000000                  KMP06:  0               ;POINTER TO KMC11 PORT REGISTER(SEL 6)

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

                                ;TEMP:  0
                                ;.=.+40

                                ;KMC11 STATUS TABLE AND ADDRESS ASSIGNMENTS
                                ;----------------------------------------------

        002100                  .=2100
002100                          KM.MAP:
002100  000001                  KMCR00: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 00
002102  000001                  KMS100: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 00
002104  000001                  KMS200: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 00
002106  000001                  KMS300: .BLKW   1               ;3RD STATUS WORD

002110  000001                  KMCR01: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 01
002112  000001                  KMS101: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 01
002114  000001                  KMS201: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 01
002116  000001                  KMS301: .BLKW   1               ;3RD STATUS WORD

002120  000001                  KMCR02: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 02
002122  000001                  KMS102: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 02
002124  000001                  KMS202: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 02
002126  000001                  KMS302: .BLKW   1               ;3RD STATUS WORD

002130  000001                  KMCR03: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 03
002132  000001                  KMS103: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 03
002134  000001                  KMS203: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 03
002136  000001                  KMS303: .BLKW   1               ;3RD STATUS WORD

002140  000001                  KMCR04: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 04
002142  000001                  KMS104: .BLKW   1               ;VECTOR FOR KMC11 NUMBER C4
002144  000001                  KMS204: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 04
002146  000001                  KMS304: .BLKW   1               ;3RD STATUS WORD
```

```

002150	000001		KMCR05:	.BLKW	1		;CONTROL STATUS REGISTER FOR KMC11 NUMBER 05
002152	000001		KMS105:	.BLKW	1		;VECTOR FOR KMC11 NUMBER 05
002154	000001		KMS205:	.BLKW	1		;DDCMP LINE# FOR KMC11 NUMBER 05
002156	000001		KMS305:	.BLKW	1		;3RD STATUS WORD

002160	000001		KMCR06:	.BLKW	1		;CONTROL STATUS REGISTER FOR KMC11 NUMBER 06
002162	000001		KMS106:	.BLKW	1		;VECTOR FOR KMC11 NUMBER 06
002164	000001		KMS206:	.BLKW	1		;DDCMP LINE# FOR KMC11 NUMBER 06
002166	000001		KMS306:	.BLKW	1		;3RD STATUS WORD

002170	000001		KMCR07:	.BLKW	1		;CONTROL STATUS REGISTER FOR KMC11 NUMBER 07
002172	000001		KMS107:	.BLKW	1		;VECTOR FOR KMC11 NUMBER 07
002174	000001		KMS207:	.BLKW	1		;DDCMP LINE# FOR KMC11 NUMBER 07
002176	000001		KMS307:	.BLKW	1		;3RD STATUS WORD

002200	000001		KMCR10:	.BLKW	1		;CONTROL STATUS REGISTER FOR KMC11 NUMBER 10
002202	000001		KMS110:	.BLKW	1		;VECTOR FOR KMC11 NUMBER 10
002204	000001		KMS210:	.BLKW	1		;DDCMP LINE# FOR KMC11 NUMBER 10
002206	000001		KMS310:	.BLKW	1		;3RD STATUS WORD

002210	000001		KMCR11:	.BLKW	1		;CONTROL STATUS REGISTER FOR KMC11 NUMBER 11
002212	000001		KMS111:	.BLKW	1		;VECTOR FOR KMC11 NUMBER 11
002214	000001		KMS211:	.BLKW	1		;DDCMP LINE# FOR KMC11 NUMBER 11
002216	000001		KMS311:	.BLKW	1		;3RD STATUS WORD

002220	000001		KMCR12:	.BLKW	1		;CONTROL STATUS REGISTER FOR KMC11 NUMBER 12
002222	000001		KMS112:	.BLKW	1		;VECTOR FOR KMC11 NUMBER 12
002224	000001		KMS212:	.BLKW	1		;DDCMP LINE# FOR KMC11 NUMBER 12
002226	000001		KMS312:	.BLKW	1		;3RD STATUS WORD

002230	000001		KMCR13:	.BLKW	1		;CONTROL STATUS REGISTER FOR KMC11 NUMBER 13
002232	000001		KMS113:	.BLKW	1		;VECTOR FOR KMC11 NUMBER 13
002234	000001		KMS213:	.BLKW	1		;DDCMP LINE# FOR KMC11 NUMBER 13
002236	000001		KMS313:	.BLKW	1		;3RD STATUS WORD

002240	000001		KMCR14:	.BLKW	1		;CONTROL STATUS REGISTER FOR KMC11 NUMBER 14
002242	000001		KMS114:	.BLKW	1		;VECTOR FOR KMC11 NUMBER 14
002244	000001		KMS214:	.BLKW	1		;DDCMP LINE# FOR KMC11 NUMBER 14
002246	000001		KMS314:	.BLKW	1		;3RD STATUS WORD

002250	000001		KMCR15:	.BLKW	1		;CONTROL STATUS REGISTER FOR KMC11 NUMBER 15
002252	000001		KMS115:	.BLKW	1		;VECTOR FOR KMC11 NUMBER 15
002254	000001		KMS215:	.BLKW	1		;DDCMP LINE# FOR KMC11 NUMBER 15
002256	000001		KMS315:	.BLKW	1		;3RD STATUS WORD

002260	000001		KMCR16:	.BLKW	1		;CONTROL STATUS REGISTER FOR KMC11 NUMBER 16
002262	000001		KMS116:	.BLKW	1		;VECTOR FOR KMC11 NUMBER 16
002264	000001		KMS216:	.BLKW	1		;DDCMP LINE# FOR KMC11 NUMBER 16
002266	000001		KMS316:	.BLKW	1		;3RD STATUS WORD

002270	000001		KMCR17:	.BLKW	1		;CONTROL STATUS REGISTER FOR KMC11 NUMBER 17
002272	000001		KMS117:	.BLKW	1		;VECTOR FOR KMC11 NUMBER 17
002274	000001		KMS217:	.BLKW	1		;DDCMP LINE# FOR KMC11 NUMBER 17
002276	000001		KMS317:	.BLKW	1		;3RD STATUS WORD

```

```
    623  002300  000000                 KM.END: 000000
```

```

                              ;KMC11 PASS COUNT AND ERROR COUNT TABLE
                              ;---------------------------------------

002302                        CNT.MAP:
002302  000000                PACT00: 0                 ;PASS COUNT FOR KMC11 NUMBER 00
002304  000000                ERCT00: 0                 ;ERROR COUNT FOR KMC11 NUMBER 00

002306  000000                PACT01: 0                 ;PASS COUNT FOR KMC11 NUMBER 01
002310  000000                ERCT01: 0                 ;ERROR COUNT FOR KMC11 NUMBER 01

002312  000000                PACT02: 0                 ;PASS COUNT FOR KMC11 NUMBER 02
002314  000000                ERCT02: 0                 ;ERROR COUNT FOR KMC11 NUMBER 02

002316  000000                PACT03: 0                 ;PASS COUNT FOR KMC11 NUMBER 03
002320  000000                ERCT03: 0                 ;ERROR COUNT FOR KMC11 NUMBER 03

002322  000000                PACT04: 0                 ;PASS COUNT FOR KMC11 NUMBER 04
002324  000000                ERCT04: 0                 ;ERROR COUNT FOR KMC11 NUMBER 04

002326  000000                PACT05: 0                 ;PASS COUNT FOR KMC11 NUMBER 05
002330  000000                ERCT05: 0                 ;ERROR COUNT FOR KMC11 NUMBER 05

002332  000000                PACT06: 0                 ;PASS COUNT FOR KMC11 NUMBER 06
002334  000000                ERCT06: 0                 ;ERROR COUNT FOR KMC11 NUMBER 06

002336  000000                PACT07: 0                 ;PASS COUNT FOR KMC11 NUMBER 07
002340  000000                ERCT07: 0                 ;ERROR COUNT FOR KMC11 NUMBER 07

002342  000000                PACT10: 0                 ;PASS COUNT FOR KMC11 NUMBER 10
002344  000000                ERCT10: 0                 ;ERROR COUNT FOR KMC11 NUMBER 10

002346  000000                PACT11: 0                 ;PASS COUNT FOR KMC11 NUMBER 11
002350  000000                ERCT11: 0                 ;ERROR COUNT FOR KMC11 NUMBER 11

002352  000000                PACT12: 0                 ;PASS COUNT FOR KMC11 NUMBER 12
002354  000000                ERCT12: 0                 ;ERROR COUNT FOR KMC11 NUMBER 12

002356  000000                PACT13: 0                 ;PASS COUNT FOR KMC11 NUMBER 13
002360  000000                ERCT13: 0                 ;ERROR COUNT FOR KMC11 NUMBER 13

002362  000000                PACT14: 0                 ;PASS COUNT FOR KMC11 NUMBER 14
002364  000000                ERCT14: 0                 ;ERROR COUNT FOR KMC11 NUMBER 14

002366  000000                PACT15: 0                 ;PASS COUNT FOR KMC11 NUMBER 15
002370  000000                ERCT15: 0                 ;ERROR COUNT FOR KMC11 NUMBER 15

002372  000000                PACT16: 0                 ;PASS COUNT FOR KMC11 NUMBER 16
002374  000000                ERCT16: 0                 ;ERROR COUNT FOR KMC11 NUMBER 16

002376  000000                PACT17: 0                 ;PASS COUNT FOR KMC11 NUMBER 17
002400  000000                ERCT17: 0                 ;ERROR COUNT FOR KMC11 NUMBER 17

```

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

 15  14  13  12  11  10  09  08  07  06  05  04  03  02  01  00
-----------------------------------------------------------------
I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
I C I O I N I T I R I O I L I   I R I E I G I I I S I T I E I 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 * I * I V I E I C I T I O I R I * 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 * I B I M I   I A I D I D I * I * I L I I I N I E I   I # I * 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 KMC11 CSR ADDRESS

STAT1:  BITS 00-08 IS KMC11 VECTOR ADDRESS
        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 KMC11 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 [PROGRAMS G AND H ONLY])
```

```

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

002402  012737  000340  177776  .START: MOV     #340,PS         ;LOCK OUT INTERRUPTS
002410  012706  001200                  MOV     #STACK,SP       ;SET UP STACK
002414  012737  007126  000024          MOV     #$PWRDN,@#24    ;SET UP POWER FAIL VECTOR
002422  013737  001472  001476          MOV     KMNUM,SAVNUM    ;SAVE NUMBER OF DEVICES IN SYSTEM.
002430  005037  011416                  CLR     SWFLG           ;CLEAR SOFT TYPEOUT FLAG
002434  105037  001203                  CLRB    $ERFLG          ;CLEAR ERROR FLAG
002440  105037  001511                  CLRB    QV.FLG          ;ZERO QUICK VERIFY FLAG
002444  012737  002070  001502          MOV     #KM.MAP-10,CREAM;GET MAP POINTER.
002452  012737  002276  001504          MOV     #CNT.MAP-4,MILK ;GET PASS COUNT MAP POINTER
002460  012737  100000  001500          MOV     #BIT15,RUN      ;POINT POINTER TO FIRST DEVICE.
002466  012700  002302                  MOV     #CNT.MAP,R0     ;PASS COUNT POINTER TO R0
002472  005020                  23$:    CLR     (R0)+           ;CLEAR TABLE
002474  022700  002402                  CMP     #CNT.MAP+100,R0 ;DONE YET?
002500  001374                          BNE     23$             ;KEEP GOING
002502  005037  001216                  CLR     $ERRPC          ;CLEAR LAST ERROR POINTER
002506  012737  000001  001202          MOV     #1,$TSTNM       ;SET UP FOR TEST 1
002514  012737  002402  001206          MOV     #.START,$LPADR  ;SET UP FOR POWER FAIL BEFORE
                                                                ;TESTING STARTS
002522  132737  000001  001336          BITB    #1,$ENV         ; IS IT RUNNING UNDER APT?
002530  001404                          BEQ     3$              ; IF NOT CHECK FOR TYPE OF SWITCH REGISTER.
002532  013737  001340  000176          MOV     $SWREG,SWREG    ; LOAD SOFTWARE SWITCH REG.
002540  000423                          BR      6$+2            ; GO SET UP SOFTWARE SWITCH REG.
002542  013746  000006          3$:     MOV     @#6,-(SP)       ;SAVE CURRENT VECTORS
002546  013746  000004                  MOV     @#4,-(SP)
002552  012737  002606  000004          MOV     #6$,@#4         ;SET UP FOR TIMEOUT
002560  012737  177570  001240          MOV     #177570,SWR     ;SET SWR TO HARD SWR ADDRESS
002566  012737  177570  001242          MOV     #177570,DISPLAY ;SET DISPLAY TO HARD SWR ADDRESS
002574  022777  177777  176436          CMP     #-1,@SWR        ;REFERENCE HARDWARE SWITCH REGISTER
002602  001402                          BEQ     6$+2            ;IF = -1 USE SOFT SWR ANYWAY
002604  000407                          BR      7$              ;IF IT EXISTS AND NOT = -1 USE HARD SWR
002606  022626                  6$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
002610  012737  000176  001240          MOV     #SWREG,SWR      ;POINTER TO SOFT SWR
002616  012737  000174  001242          MOV     #DISPREG,DISPLAY;POINTER TO SOFT DISPLAY REG
002624  012637  000004          7$:     MOV     (SP)+,@#4       ;RESTORE VECTORS
002630  012637  000006                  MOV     (SP)+,@#6       ;
002634  105737  001506                  TSTB    INIFLG          ;HAS INITIALIZATION BEEN PERFORMED
002640  001006                          BNE     20$             ;BR IF YES
002642  022737  004070  000042          CMP     #$ENDAD,@#42    ;IF ACT-11 AUTOMATIC MODE, DON'T TYPE ID
002650  001402                          BEQ     20$
002652  104401  001000                  TYPE    ,MTITLE         ;TYPE TITLE MESSAGE
002656  004737  011212          20$:    JSR     PC,CKSWR                ;CHECK FOR SOFT SWR
002662  017737  176352  001446          MOV     @SWR,STRTSW     ;STORE STARTING SWITCHES
002670  005737  000042                  TST     @#42            ;IS IT RUNNING IN AUTO MODE?
002674  001402                          BEQ     .+6             ;BR IF NO
002676  005037  001446                  CLR     STRTSW          ;IF YES, CLEAR SWITCHES
002702  032737  000001  001446          BIT     #SW00,STRTSW    ;IF SW00=1, QUESTIONS ARE ASKED.
002710  001012                          BNE     17$             ;BR IF SW00=1
002712  105737  001446                  TSTB    STRTSW          ;BIT7=1??
```

N03

```
002716  100007                          BPL     17$             ;BR IF SW07=0
002720  005737  001470                  TST     KMACTV          ;ARE ANY DEVICES SELECTED?
002724  001027                          BNE     16$             ;BR IF YES
002726  104401  010731                  TYPE,   NOACT           ;NO DEVICES SELECTED.
002732  000000                          HALT                    ;STOP THE SHOW
002734  000776                          BR      .-2             ;DISQUALIFY CONTINUE SWITCH
002736  105737  001336          17$:    TSTB    $ENV            ; IS IT UNDER APT DUMP MODE?
002742  001405                          BEQ     27$             ; YES, CHECK IF APT SIZED IT?
002744  132737  000001  001336          BITB    #1,$ENV         ; IS IT UNDER Q,V OR RUN MODE?
002752  001012                          BNE     30$             ; YES, NEEDS ONLY APT SIZING.
002754  000406                          BR      33$             ; NO, NEEDS REGULAR AUTO.SIZE.
002756  105737  001337          27$:    TSTB    $ENVM           ; IS IT SIZED BY APT?
002762  100406                          BMI     30$             ; YES, NEEDS ONLY APT SIZING.
002764  042737  000001  001446          BIC     #SW00,STRTSW    ; SIZE ONLY IN AUTO MODE.
002772  004737  012110          33$:    JSR     PC,AUTO.SIZE    ; GO DO THE AUTO.SIZE.
002776  000402                          BR      16$             ; GO PRINT THE MAP.
003000  004737  013510          30$:    JSR     PC,APT.SIZE     ; GO DO THE APT SIZING.
003004  105737  001506          16$:    TSTB    INIFLG          ;FIRST TIME?
003010  001410                          BEQ     21$             ;BR IF YES
003012  105737  001446                  TSTB    STRTSW          ;IF USING SAME PARAMETERS DONT TYPE MAP
003016  100431                          BMI     1$
003020  032737  000006  001446          BIT     #BIT1!BIT2,STRTSW;IS TEST NO. OR LOCK SELECTED
003026  001403                          BEQ     24$             ;IF NO THEN TYPE STATUS
003030  000424                          BR      1$              ;IF YES DO NOT TYPE STATUS
003032  105137  001506          21$:    COMB    INIFLG          ;SET FLAG
003036  104401  010077          24$:    TYPE    ,XHEAD          ;TYPE HEADER
003042  012704  002100                  MOV     #KM.MAP,R4      ;SET POINTER
003046  010437  001276          5$:     MOV     R4,$TMP0        ;SET ADDRESS
003052  012437  001300                  MOV     (R4)+,$TMP1     ;SET CSR
003056  001411                          BEQ     1$              ;ALL DONE IF ZERO
003060  012437  001302                  MOV     (R4)+,$TMP2     ;SET STAT1
003064  012437  001304                  MOV     (R4)+,$TMP3     ;SET STAT2
003070  012437  001306                  MOV     (R4)+,$TMP4     ;SET STAT3
003074  104416                          CONVRT                  ;TYPE OUT STATUS MAP
003076  011060                          XSTATQ                  ;
003100  000762                          BR      5$
003102  012700  002100          1$:     MOV     #KM.MAP,R0      ;R0 POINTS TO STATUS TABLE

                                ;;***************************************************************
                                ;;*AUTO SIZE TEST
                                ;;*THIS TEST VERIFYS THAT THE KMC11S 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 KMC11, THE FIRST
                                ;;* KMC11 IS 760110. NO DEVICE SHOULD EVER BE AT
                                ;;*ADDRESS 760000.
                                ;;***************************************************************

003106  013746  000004                  MOV     @#4,-(SP)       ;SAVE LOC 4
003112  013746  000006                  MOV     @#6,-(SP)       ;SAVE LOC 6
003116  005037  000006                  CLR     @#6             ;CLEAR VEC+2
003122  005037  001302                  CLR     $TMP2           ;CLEAR FLAG
003126  011037  002066          AUSTRT: MOV     (R0),KMCSR      ;GET NEXT KMC CSR
003132  001510                          BEQ     AUDONE          ;BR IF DONE
003134  012737  003240  000004  2$:     MOV     #NODEV,@#4      ;SET UP FOR TIMEOUT
003142  012703  000010          3$:     MOV     #10,R3          ;R3 IS COUNT OF DEVICES BEFORE KMC
```

```
003146  012702  003342          4$:      MOV     #DEVTAB,R2        ;R2 IS DEVICE TABLE PONTER
003152  012701  160010                   MOV     #160010,R1        ;START WITH ADDRESS 160010
003156  005711                  FLOAT:   TST     (R1)              ;CHECK ADDRESS IN R1
003160  111204                           MOVB    (R2),R4           ;IF NO TIMEOUT, GET NEXT ADDRESS
003162  060401                           ADD     R4,R1             ;IN R1
003164  005201                           INC     R1                ;
003166  040401                           BIC     R4,R1             ;
003170  005703                           TST     R3                ;ANY MORE DEVICES TO CHECK FOR?
003172  001371                           BNE     FLOAT             ;BR IF YES
003174  012737  003244  000004           MOV     #ERR,@#4          ;OK ONLY KMC'S ARE LEFT, SET UP FOR TIMEOUT
003202  005711                  FY:      TST     (R1)              ;CHECK KMC ADDRESS
003204  020137  002066                   CMP     R1,KMCSR          ;DOES IT MATCH
003210  001403                           BEQ     OK                ;BR IF YES
003212  062701  000010                   ADD     #10,R1            ;GET NEXT KMC ADDRESS
003216  000771                           BR      FY                ;DO IT AGAIN
003220  062700  000010          OK:      ADD     #10,R0            ;SKIP TO NEXT KMC CSR
003224  062701  000010                   ADD     #10,R1            ; GET NEXT KMC ADDRESS
003230  011037  002066                   MOV     (R0),KMCSR        ; GET NEXT KMC CSR
003234  001447                           BEQ     AUDONE            ; BRANCH IF ALL DONE.
003236  000761                           BR      FY                ; DO IT AGAIN.
003240  122243                  NODEV:   CMPB    (R2)+,-(R3)       ;ON TIMEOUT, INC R2, DEC R3
003242  000002                           RTI                       ;$LPADR
003244  005737  001302          ERR:     TST     $TMP2             ;CHECK FLAG IF = 0 TYPE HEADER
003250  001014                           BNE     1$                ;SKIP HEADER
003252  104401                           TYPE                      ;TYPEOUT HEADER MESSAGE
003254  010762                           CONERR                    ;CONFIGURATION ERROR!!!!
003256  012737  003244  001460           MOV     #ERR,SAVPC        ;SAVE PC FOR TYPEOUT
003264  104417                           CNVRT                     ;TYPE OUT ERROR PC
003266  003322                           ERRPC                     ;
003270  104401                           TYPE                      ;TYPE REST OF HEADER
003272  011027                           CNERR                     ;
003274  012737  177777  001302           MOV     #-1,$TMP2         ;SET FLAG SO IT ONLY GETS TYPED ONCE
003302  010137  001264          1$:      MOV     R1,$REG1          ;SAVE R1 FOR TYPEOUT
003306  104416                           CONVRT
003310  003330                           CONTAB                    ;TYPE CSR VALUES
003312  104401                  3$:      TYPE
003314  011050                           KMCM
003316  022626                  4$:      CMP     (SP)+,(SP)+       ;ADJUST STACK
003320  000737                           BR      OK                ;BR TO GET OUT
003322  000001                  ERRPC:   1
003324     006     002                   .BYTE   6,2
003326  001460                           SAVPC
003330  000002                  CONTAB:  2
003332     006     004                   .BYTE   6,4
003334  001264                           $REG1
003336     006     002                   .BYTE   6,2
003340  002066                           KMCSR
003342     007                  DEVTAB:  .BYTE   7                 ;DJ
003343     017                           .BYTE   17                ;DH
003344     007                           .BYTE   7                 ;DQ
003345     007                           .BYTE   7                 ;DU
003346     007                           .BYTE   7                 ;DUP
003347     007                           .BYTE   7                 ;LK
003350     007                           .BYTE   7                 ;DMC
003351     007                           .BYTE   7                 ;DZ
003352     007                           .BYTE   7                 ;KMC
```

```
                003354                          .EVEN
        003354                                  AUDONE:
        003354  012637  000006                  1$:     MOV     (SP)+,@#6               ;RESTORE LOC 6
        003360  012637  000004                          MOV     (SP)+,@#4               ;RESTORE LOC 4
        003364  032737  000010  001446                  BIT     #SW03,STRTSW            ;SELECT SPECIFIC DEVICES??
        003372  001422                                  BEQ     3$                      ;BR IF NO.
        003374  104401  010017                          TYPE    ,MNEW                   ;TYPE THE MESSAGE.
        003400  005000                                  CLR     R0                      ;ZERO DATA LIGHTS
        003402  000000                                  HALT                            ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
        003404  027737  175630  001474                  CMP     @SWR,SAVACT             ;IS THE NUMBER VALID?
        003412  101404                                  BLOS    2$                      ;BR IF NUMBER IS OK.
        003414  104401  007672                          TYPE    ,MERR3                  ;TELL USER OF INVALID NUMBER.
        003420  000000                                  HALT                            ;STOP EVERY THING.
        003422  000776                                  BR      .-2                     ;RESTART THE PROGRAM AGAIN.
        003424  017737  175610  001470  2$:     MOV     @SWR,KMACTV             ;GET NEW DEVICE PATTERN
        003432  013700  001470                          MOV     KMACTV,R0               ;SHOW THE USER WHAT HE SELECTED.
        003436  000000                                  HALT                            ;CONTINUE DYNAMIC SWITCHES.
        003440  012700  000300                  3$:     MOV     #300,R0                 ;PREPARE TO CLEAR THE FLOATING
        003444  012701  000302                          MOV     #302,R1                 ;VECTOR AREA. 300-776
        003450  010120                          4$:     MOV     R1,(R0)+                ;START PUTTING "PC+2 - HALT"
        003452  005021                                  CLR     (R1)+                   ;IN VECTOR AREA.
        003454  022021                                  CMP     (R0)+,(R1)+             ;POP POINTERS
        003456  022700  001000                          CMP     #1000,R0                ;ALL DONE??
        003462  001372                                  BNE     4$                      ;BR IF NO.

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


        003464  012706  001200                  .BEGIN: MOV     #STACK,SP               ;SET UP STACK
        003470  013746  000006                          MOV     @#6,-(SP)               ;SAVE LOC 6
        003474  013746  000004                          MOV     @#4,-(SP)               ;SAVE LOC 4
        003500  005000                                  CLR     R0                      ;START AT 0
        003502  012737  003546  000004                  MOV     #2$,@#4                 ;SET UP FOR TIME OUT
        003510  005037  000006                          CLR     @#6                     ;TO AUTOSIZE MEMORY
        003514  005720                          6$:     TST     (R0)+                   ;CHECK ADDRESS IN R0
        003516  022700  157776                          CMP     #157776,R0              ;IS IT AT LEAST 28K
        003522  001374                                  BNE     6$                      ;BR IF NO
        003524  162700  007776                          SUB     #7776,R0                ;SAVE 2K FOR MONITORS
        003530  010037  001466                  7$:     MOV     R0,MEMLIM               ;STORE MEMORY LIMIT
        003534  012637  000004                          MOV     (SP)+,@#4               ;RESTORE LOC 4
        003540  012637  000006                          MOV     (SP)+,@#6               ;RESTORE LOC 6
        003544  000413                                  BR      10$                     ;CONTINUE
        003546  022626                          2$:     CMP     (SP)+,(SP)+             ;ADJUST STACK
        003550  162700  000004                          SUB     #4,R0                   ;GET LAST GOOD ADDRESS
        003554  162700  007776                          SUB     #7776,R0                ;SAVE 2K FOR MONITORS
        003560  022700  030000                          CMP     #30000,R0               ;IS IT 8K?
        003564  001361                                  BNE     7$                      ;BR IF NO
        003566  012700  037400                          MOV     #37400,R0               ;IF 8K DON'T SAVE 2K
        003572  000756                                  BR      7$                      ;
        003574  012737  000340  177776  10$:    MOV     #340,PS                 ;LOCK OUT INTERRUPTS
        003602  032737  000004  001446                  BIT     #BIT2,STRTSW            ;CHECK FOR LOCK ON TEST
        003610  001406                                  BEQ     1$                      ;BR IF NO LOCK DESIRED.
        003612  104401  007716                          TYPE    ,MLOCK                  ;TYPE LOCK SELECTED.
        003616  012737  000240  004146                  MOV     #NOP,TTST               ;SET UP TO LOCK
        003624  000403                                  BR      3$                      ;CONTINUE ALONG.
        003626  013737  004360  004146  1$:     MOV     BRW,TTST                ;PREPARE NORMAL SCOPE ROUTINE
```

```
   955 003634  012737  011460  001206  3$:     MOV     #CYCLE,$LPADR     ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
   956 003642  032737  000002  001446  4$:     BIT     #SW01,STRTSW      ;IS TEST NO. SELECTED?
   957 003650  001002                          BNE     5$                ;BR IF YES
   958 003652  104401  007642                  TYPE    ,MR               ;TYPE R
   959 003656  000177  175324          5$:     JMP     @$LPADR           ;START TESTING
```

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

                                        .SBTTL  END OF PASS ROUTINE

                                        ;;*********************************************************************
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO CYCLE

 003662                                 $EOP:
 003662  000005                                 RESET
 003664  005237  001324                         INC     $PASS                   ; INCREMENT THE PASS COUNT
 003670  105037  001203                         CLRB    $ERFLG                  ; CLEAR ERROR FLAG
 003674  104401  007620                         TYPE    ,MEPASS                 ; TYPE END PASS.
 003700  104401  007745                         TYPE    ,MCSRX                  ; TYPE "CSR"
 003704  104417  004104                         CNVRT   ,XCSR                   ; SHOW IT.
 003710  104401  007753                         TYPE    ,MVECX                  ; TYPE VECTOR.
 003714  104417  004112                         CNVRT   ,XVEC                   ; SHOW IT.
 003720  104401  007761                         TYPE    ,MPASSX                 ; TYPE " PASSES "
 003724  104417  004120                         CNVRT   ,XPASS                  ; SHOW IT.
 003730  104401  007772                         TYPE    ,MERRX                  ; TYPE " ERRORS "
 003734  104417  004126                         CNVRT   ,XERR                   ; SHOW IT.
 003740  013700  001504                         MOV     MILK,R0                 ; SET POINTER TO PASSCNT.
 003744  013720  001324                         MOV     $PASS,(R0)+             ; SAVE THE PASS COUNT.
 003750  013720  001212                         MOV     $ERTTL,(R0)+            ; SAVE ERROR COUNT
 003754  013777  002060  176074                 MOV     KMRLVL,@KMRVEC          ; RESTORE THE RECEIVER INTERRUPT VECTOR.
 003762  005077  176072                         CLR     @KMRLVL                 ; RESTORE RECEIVER LEVEL
 003766  013777  002064  176066                 MOV     KMTLVL,@KMTVEC          ; RESTORE THE TRANSMIT INTERRUPT VECTOR.
 003774  005077  176064                         CLR     @KMTLVL                 ; RESTORE TRANSMITTER LEVEL
 004000  005337  001476                         DEC     SAVNUM                  ; ALL DEVICE TESTED?
 004004  001035                                 BNE     $DOAGN                  ; BRANCH IF NO.
 004006  112737  000377  001511                 MOVB    #377,QV.FLG             ; SET QUICK VERIFY FLAG.
 004014  013737  001472  001476                 MOV     KMNUM,SAVNUM            ; RESTORE DEVICE COUNT.
 004022  005037  001216                         CLR     $ERRPC                  ; CLEAR LAST ERROR PC
 004026  005037  001310                         CLR     $TIMES                  ;;ZERO THE NUMBER OF ITERATIONS
 004032  005237  001324                         INC     $PASS                   ;;INCREMENT THE PASS NUMBER
 004036  042737  100000  001324                 BIC     #100000,$PASS           ;;DON'T ALLOW A NEG. NUMBER
 004044  005327                                 DEC     (PC)+                   ;;LOOP?
 004046  000001                         $EOPCT: .WORD   1
 004050  003013                                 BGT     $DOAGN                  ;;YES
 004052  012737                                 MOV     (PC)+,@(PC)+            ;;RESTORE COUNTER
 004054  000001                         $ENDCT: .WORD   1
 004056  004046                                         $EOPCT
 004060  013700  000042                 $GET42: MOV     @#42,R0                 ;;GET MONITOR ADDRESS
 004064  001405                                 BEQ     $DOAGN                  ;;BRANCH IF NO MONITOR
 004066  000005                                 RESET                           ;;CLEAR THE WORLD
 004070  004710                         $ENDAD: JSR     PC,(R0)                 ;;GO TO MONITOR
 004072  000240                                 NOP                             ;;SAVE ROOM
 004074  000240                                 NOP                             ;;FOR
 004076  000240                                 NOP                             ;;ACT11
 004100                                 $DOAGN:
 004100  000137                                 JMP     @(PC)+                  ;;RETURN
```

F04

```
004102  011460                       $RTNAD: .WORD   CYCLE
004104  000001                       XCSR:   1
004106     006     002                       .BYTE   6,2
004110  002066                               KMCSR
004112  000001                       XVEC:   1
004114     004     002                       .BYTE   4,2
004116  002056                               KMRVEC
004120  000001                       XPASS:  1
004122     006     002                       .BYTE   6,2
004124  001324                               $PASS
004126  000001                       XERR:   1
004130     006     002                       .BYTE   6,2
004132  001212                               $ERTTL

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


                                     .SBTTL  SCOPE HANDLER ROUTINE

                                     ;;************************************************************
                                     ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                     ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                     ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                     ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                     ;*SW14=1          LOOP ON TEST
                                     ;*SW11=1          INHIBIT ITERATIONS
                                     ;*CALL
                                     ;*       SCOPE           ;;SCOPE=IOT

004134                               $SCOPE:
004134  005037  001216                       CLR     $ERRPC                  ; CLEAR LAST ERROR PC
004140  023716  013734                       CMP     TST1+2,(SP)             ; IS THIS TEST #1 ?
004144  001413                               BEQ     $XTSTR                  ; IF SO DON'T LOOP.
004146  000406                       TTST:   BR      1$                      ;
004150  105777  175070                       TSTB    @$TKS                   ; KEYBOARD DONE ?
004154  100067                               BPL     $OVER                   ; IF NO DONT WAIT.
004156  017766  175064  177776               MOV     @$TKB,-2(SP)            ;
004164  032777  040000  175046       1$:     BIT     #BIT14,@SWR           ;;LOOP ON PRESENT TEST?
004172  001060                               BNE     $OVER                 ;;YES IF SW14=1
                                     ;#####START OF CODE FOR THE XOR TESTER#####
004174  000416                       $XTSTR: BR      6$                    ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                           ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
004176  013746  000004                       MOV     @#ERRVEC,-(SP)        ;;SAVE THE CONTENTS OF THE ERROR VECTOR
004202  012737  004222  000004               MOV     #5$,@#ERRVEC          ;;SET FOR TIMEOUT
004210  005737  177060                       TST     @#177060              ;;TIME OUT ON XOR?
004214  012637  000004                       MOV     (SP)+,@#ERRVEC        ;;RESTORE THE ERROR VECTOR
004220  000436                               BR      $SVLAD                ;;GO TO THE NEXT TEST
004222  022626                       5$:     CMP     (SP)+,(SP)+           ;;CLEAR THE STACK AFTER A TIME OUT
004224  012637  000004                       MOV     (SP)+,@#ERRVEC        ;;RESTORE THE ERROR VECTOR
004230  000441                               BR      $OVER                 ;;LOOP ON THE PRESENT TEST
004232                               6$:;#####END OF CODE FOR THE XOR TESTER#####
004232  105737  001203               2$:     TSTB    $ERFLG                ;;HAS AN ERROR OCCURRED?
004236  001404                               BEQ     3$                    ;;BR IF NO
004240  105037  001203               4$:     CLRB    $ERFLG                ;;ZERO THE ERROR FLAG
004244  005037  001310                       CLR     $TIMES                ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
004250  032777  004000  174762       3$:     BIT     #BIT11,@SWR           ;;INHIBIT ITERATIONS?
```

```
004256  001011                              BNE     1$                      ;;BR IF YES
004260  005737  001324                      TST     $PASS                   ;;IF FIRST PASS OF PROGRAM
004264  001406                              BEQ     1$                      ;;      INHIBIT ITERATIONS
004266  005237  001204                      INC     $ICNT                   ;;INCREMENT ITERATION COUNT
004272  023737  001310  001204              CMP     $TIMES,$ICNT            ;;CHECK THE NUMBER OF ITERATIONS MADE
004300  002015                              BGE     $OVER                   ;;BR IF MORE ITERATION REQUIRED
004302  012737  000001  001204  1$:         MOV     #1,$ICNT                ;;REINITIALIZE THE ITERATION COUNTER
004310  013737  004362  001310              MOV     $MXCNT,$TIMES           ;;SET NUMBER OF ITERATIONS TO DO
004316  105237  001202          $SVLAD:     INCB    $TSTNM                  ;;COUNT TEST NUMBERS
004322  113737  001202  001322              MOVB    $TSTNM,$TESTN           ;;SET TEST NUMBER IN APT MAILBOX
004330  011637  001206                      MOV     (SP),$LPADR             ;;SAVE SCOPE LOOP ADDRESS
004334  013777  001202  174700  $OVER:      MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER
004342  013716  001206                      MOV     $LPADR,(SP)             ;;FUDGE RETURN ADDRESS
004346  005037  001444                      CLR     LOCK                            ; RESET LOCK ON DATA.
004352  013701  002066                      MOV     KMCSR,R1                        ; R1 CONTAINS BASE KMC ADDRESS.
004356  000002                              RTI
004360  000406                  BRW:        .WORD   406
004362  000020                  $MXCNT:     20                              ;;MAX. NUMBER OF ITERATIONS

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

004364  004737  011212          .SCOP1:     JSR     PC,CKSWR                        ;CHECK FOR SOFT SWR
004370  032777  001000  174642              BIT     #SW09,@SWR              ;IS SW09=1(SET)?
004376  001405                              BEQ     1$                      ;BR IF NOT SET.
004400  005737  001444                      TST     LOCK
004404  001402                              BEQ     1$
004406  013716  001444                      MOV     LOCK,(SP)               ;GOTO THE ADDRESS IN LOCK.
004412  000002                  1$:         RTI                             ;GO BACK.

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

                                .SBTTL  TYPE ROUTINE

                                ;;************************************************************
                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*          TYPE    ,MESADR                 ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*          TYPE
                                ;*          MESADR
                                ;*

004414  105737  001257          $TYPE:      TSTB    $TPFLG                  ;;IS THERE A TERMINAL?
004420  100002                              BPL     1$                      ;;BR IF YES
004422  000000                              HALT                            ;;HALT HERE IF NO TERMINAL
004424  000430                              BR      3$                      ;;LEAVE
004426  010046                  1$:         MOV     R0,-(SP)                ;;SAVE R0
004430  017600  000002                      MOV     @2(SP),R0               ;;GET ADDRESS OF ASCIZ STRING
```

```
004434  122737  000001  001336          CMPB    #APTENV,$ENV      ;;RUNNING IN APT MODE
004442  001011                          BNE     62$               ;;NO,GO CHECK FOR APT CONSOLE
004444  132737  000100  001337          BITB    #APTSPOOL,$ENVM   ;;SPOOL MESSAGE TO APT
004452  001405                          BEQ     62$               ;;NO,GO CHECK FOR CONSOLE
004454  010037  004464                  MOV     R0,61$            ;;SETUP MESSAGE ADDRESS FOR APT
004460  004737  004704                  JSR     PC,$ATY3          ;;SPOOL MESSAGE TO APT
004464  000000                  61$:    .WORD   0                 ;;MESSAGE ADDRESS
004466  132737  000040  001337  62$:    BITB    #APTCSUP,$ENVM    ;;APT CONSOLE SUPPRESSED
004474  001003                          BNE     60$               ;;YES,SKIP TYPE OUT
004476  112046                  2$:     MOVB    (R0)+,-(SP)       ;;PUSH CHARACTER TO BE TYPED ONTO STACK
004500  001005                          BNE     4$                ;;BR IF IT ISN'T THE TERMINATOR
004502  005726                          TST     (SP)+             ;;IF TERMINATOR POP IT OFF THE STACK
004504  012600                  60$:    MOV     (SP)+,R0          ;;RESTORE R0
004506  062716  000002          3$:     ADD     #2,(SP)           ;;ADJUST RETURN PC
004512  000002                          RTI                       ;;RETURN
004514  122716  000011          4$:     CMPB    #HT,(SP)          ;;BRANCH IF <HT>
004520  001430                          BEQ     8$
004522  122716  000200                  CMPB    #CRLF,(SP)        ;;BRANCH IF NOT <CRLF>
004526  001006                          BNE     5$
004530  005726                          TST     (SP)+             ;;POP  <CR><LF> EQUIV
004532  104401                          TYPE                      ;;TYPE A CR AND LF
004534  001313                          $CRLF
004536  105037  004672                  CLRB    $CHARCNT          ;;CLEAR CHARACTER COUNT
004542  000755                          BR      2$                ;;GET NEXT CHARACTER
004544  004737  004626          5$:     JSR     PC,$TYPEC         ;;GO TYPE THIS CHARACTER
004550  123726  001256          6$:     CMPB    $FILLC,(SP)+      ;;IS IT TIME FOR FILLER CHARS.?
004554  001350                          BNE     2$                ;;IF NO GO GET NEXT CHAR.
004556  013746  001254                  MOV     $NULL,-(SP)       ;;GET # OF FILLER CHARS. NEEDED
                                                                  ;;AND THE NULL CHAR.
004562  105366  000001          7$:     DECB    1(SP)             ;;DOES A NULL NEED TO BE TYPED?
004566  002770                          BLT     6$                ;;BR IF NO--GO POP THE NULL OFF OF STACK
004570  004737  004626                  JSR     PC,$TYPEC         ;;GO TYPE A NULL
004574  105337  004672                  DECB    $CHARCNT          ;;DO NOT COUNT AS A COUNT
004600  000770                          BR      7$                ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR

004602  112716  000040          8$:     MOVB    #' ,(SP)          ;;REPLACE TAB WITH SPACE
004606  004737  004626          9$:     JSR     PC,$TYPEC         ;;TYPE A SPACE
004612  132737  000007  004672          BITB    #7,$CHARCNT       ;;BRANCH IF NOT AT
004620  001372                          BNE     9$                ;;TAB STOP
004622  005726                          TST     (SP)+             ;;POP SPACE OFF STACK
004624  000724                          BR      2$                ;;GET NEXT CHARACTER
004626  105777  174416          $TYPEC: TSTB    @$TPS             ;;WAIT UNTIL PRINTER IS READY
004632  100375                          BPL     $TYPEC
004634  116677  000002  174410          MOVB    2(SP),@$TPB       ;;LOAD CHAR TO BE TYPED INTO DATA REG.
004642  122766  000015  000002          CMPB    #CR,2(SP)         ;;IS CHARACTER A CARRIAGE RETURN?
004650  001003                          BNE     1$                ;;BRANCH IF NO
004652  105037  004672                  CLRB    $CHARCNT          ;;YES--CLEAR CHARACTER COUNT
004656  000406                          BR      $TYPEX            ;;EXIT
004660  122766  000012  000002  1$:     CMPB    #LF,2(SP)         ;;IS CHARACTER A LINE FEED?
004666  001402                          BEQ     $TYPEX            ;;BRANCH IF YES
004670  105227                          INCB    (PC)+             ;;COUNT THE CHARACTER
004672  000000                  $CHARCNT:.WORD  0                 ;;CHARACTER COUNT STORAGE
004674  000207                  $TYPEX: RTS     PC
```

```
                                        .SBTTL  APT COMMUNICATIONS ROUTINE

                                        ;;*************************************************************
 004676  112737  000001  005142  $ATY1:  MOVB    #1,$FFLG          ;;TO REPORT FATAL ERROR
 004704  112737  000001  005140  $ATY3:  MOVB    #1,$MFLG          ;;TO TYPE A MESSAGE
 004712  000403                          BR      $ATYC
 004714  112737  000001  005142  $ATY4:  MOVB    #1,$FFLG          ;;TO ONLY REPORT FATAL ERROR
 004722                          $ATYC:
 004722  010046                          MOV     R0,-(SP)          ;;PUSH R0 ON STACK
 004724  010146                          MOV     R1,-(SP)          ;;PUSH R1 ON STACK
 004726  105737  005140                  TSTB    $MFLG             ;;SHOULD TYPE A MESSAGE?
 004732  001450                          BEQ     5$                ;;IF NOT:  BR
 004734  122737  000001  001336          CMPB    #APTENV,$ENV      ;;OPERATING UNDER APT?
 004742  001031                          BNE     3$                ;;IF NOT:  BR
 004744  132737  000100  001337          BITB    #APTSPOOL,$ENVM   ;;SHOULD SPOOL MESSAGES?
 004752  001425                          BEQ     3$                ;;IF NOT:  BR
 004754  017600  000004                  MOV     @4(SP),R0         ;;GET MESSAGE ADDR.
 004760  062766  000002  000004          ADD     #2,4(SP)                  ;;BUMP RETURN ADDR.
 004766  005737  001316          1$:     TST     $MSGTYPE          ;;SEE IF DONE W/ LAST XMISSION?
 004772  001375                          BNE     1$                ;;IF NOT:  WAIT
 004774  010037  001332                  MOV     R0,$MSGAD         ;;PUT ADDR IN MAILBOX
 005000  105720                  2$:     TSTB    (R0)+             ;;FIND END OF MESSAGE
 005002  001376                          BNE     2$
 005004  163700  001332                  SUB     $MSGAD,R0         ;;SUB START OF MESSAGE
 005010  006200                          ASR     R0                ;;GET MESSAGE LNGTH IN WORDS
 005012  010037  001334                  MOV     R0,$MSGLGT        ;;PUT LENGTH IN MAILBOX
 005016  012737  000004  001316          MOV     #4,$MSGTYPE       ;;TELL APT TO TAKE MSG.
 005024  000413                          BR      5$
 005026  017637  000004  005052  3$:     MOV     @4(SP),4$         ;;PUT MSG ADDR IN JSR LINKAGE
 005034  062766  000002  000004          ADD     #2,4(SP)                  ;;BUMP RETURN ADDRESS
 005042  013746  177776                  MOV     177776,-(SP)      ;;PUSH 177776 ON STACK
 005046  004737  004414                  JSR     PC,$TYPE          ;;CALL TYPE MACRO
 005052  000000                  4$:     .WORD   0
 005054                          5$:
 005054  105737  005142          10$:    TSTB    $FFLG             ;;SHOULD REPORT FATAL ERROR?
 005060  001416                          BEQ     12$               ;;IF NOT:  BR
 005062  005737  001336                  TST     $ENV              ;;RUNNING UNDER APT?
 005066  001413                          BEQ     12$               ;;IF NOT:  BR
 005070  005737  001316          11$:    TST     $MSGTYPE          ;;FINISHED LAST MESSAGE?
 005074  001375                          BNE     11$               ;;IF NOT:  WAIT
 005076  017637  000004  001320          MOV     @4(SP),$FATAL     ;;GET ERROR #
 005104  062766  000002  000004          ADD     #2,4(SP)                  ;;BUMP RETURN ADDR.
 005112  005237  001316                  INC     $MSGTYPE          ;;TELL APT TO TAKE ERROR
 005116  105037  005142          12$:    CLRB    $FFLG             ;;CLEAR FATAL FLAG
 005122  105037  005141                  CLRB    $LFLG             ;;CLEAR LOG FLAG
 005126  105037  005140                  CLRB    $MFLG             ;;CLEAR MESSAGE FLAG
 005132  012601                          MOV     (SP)+,R1          ;;POP STACK INTO R1
 005134  012600                          MOV     (SP)+,R0          ;;POP STACK INTO R0
 005136  000207                          RTS     PC                ;;RETURN
 005140     000                  $MFLG:  .BYTE   0                 ;;MESSG. FLAG
 005141     000                  $LFLG:  .BYTE   0                 ;;LOG FLAG
 005142     000                  $FFLG:  .BYTE   0                 ;;FATAL FLAG
         005144                          .EVEN
         000200                  APTSIZE=200
         000001                  APTENV=001
         000100                  APTSPOOL=100
```

```
               000040                          APTCSUP=040
                                                       ;---------------------------

                                               .SBTTL  TTY INPUT ROUTINE

                                               ;;*****************************************************************
                                               .ENABL  LSB

                                               .DSABL  LSB


                                               ;;*****************************************************************
                                               ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                               ;*CALL:
                                               ;*       RDCHR                          ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                               ;*       RETURN HERE                    ;;CHARACTER IS ON THE STACK
                                               ;*                                      ;;WITH PARITY BIT STRIPPED OFF
                                               ;

       005144  011646                          $RDCHR: MOV     (SP),-(SP)              ;;PUSH DOWN THE PC
       005146  016666  000004  000002                  MOV     4(SP),2(SP)             ;;SAVE THE PS
       005154  105777  174064                  1$:     TSTB    @$TKS                   ;;WAIT FOR
       005160  100375                                  BPL     1$                      ;;A CHARACTER
       005162  117766  174060  000004                  MOVB    @$TKB,4(SP)             ;;READ THE TTY
       005170  042766  177600  000004                  BIC     #^C<177>,4(SP)          ;;GET RID OF JUNK IF ANY
       005176  026627  000004  000023                  CMP     4(SP),#23               ;;IS IT A CONTROL-S?
       005204  001013                                  BNE     3$                      ;;BRANCH IF NO
       005206  105777  174032                  2$:     TSTB    @$TKS                   ;;WAIT FOR A CHARACTER
       005212  100375                                  BPL     2$                      ;;LOOP UNTIL ITS THERE
       005214  117746  174026                          MOVB    @$TKB,-(SP)             ;;GET CHARACTER
       005220  042716  177600                          BIC     #^C177,(SP)             ;;MAKE IT 7-BIT ASCII
       005224  022627  000021                          CMP     (SP)+,#21               ;;IS IT A CONTROL-Q?
       005230  001366                                  BNE     2$                      ;;IF NOT DISCARD IT
       005232  000750                                  BR      1$                      ;;YES, RESUME
       005234  026627  000004  000140          3$:     CMP     4(SP),#140              ;;IS IT UPPER CASE?
       005242  002407                                  BLT     4$                      ;;BRANCH IF YES
       005244  026627  000004  000175                  CMP     4(SP),#175              ;;IS IT A SPECIAL CHAR?
       005252  003003                                  BGT     4$                      ;;BRANCH IF YES
       005254  042766  000040  000004                  BIC     #40,4(SP)               ;;MAKE IT UPPER CASE
       005262  000002                          4$:     RTI                             ;;GO BACK TO USER
                                               ;;*****************************************************************
                                               ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                               ;*CALL:
                                               ;*       RDLIN                          ;;INPUT A STRING FROM THE TTY
                                               ;*       RETURN HERE                    ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                               ;*                                      ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

       005264  010346                          $RDLIN: MOV     R3,-(SP)                ;;SAVE R3
       005266  005046                                  CLR     -(SP)                   ;;CLEAR THE RUBOUT KEY
       005270  012703  005520                  1$:     MOV     #$TTYIN,R3              ;;GET ADDRESS
       005274  022703  005527                  2$:     CMP     #$TTYIN+7,R3            ;;BUFFER FULL?
       005300  101456                                  BLOS    4$                      ;;BR IF YES
       005302  104402                                  RDCHR                           ;;GO READ ONE CHARACTER FROM THE TTY
       005304  112613                                  MOVB    (SP)+,(R3)              ;;GET CHARACTER
       005306  122713  000177                  10$:    CMPB    #177,(R3)               ;;IS IT A RUBOUT
       005312  001022                                  BNE     5$                      ;;BR IF NO
```

```
005314  005716                                  TST     (SP)                    ;;IS THIS THE FIRST RUBOUT?
005316  001007                                  BNE     6$                      ;;BR IF NO
005320  112737  000134  005516                  MOVB    #'\,9$                  ;;TYPE A BACK SLASH
005326  104401  005516                          TYPE    ,9$
005332  012716  177777                          MOV     #-1,(SP)                ;;SET THE RUBOUT KEY
005336  005303                          6$:     DEC     R3                      ;;BACKUP BY ONE
005340  020327  005520                          CMP     R3,#$TTYIN              ;;STACK EMPTY?
005344  103434                                  BLO     4$                      ;;BR IF YES
005346  111337  005516                          MOVB    (R3),9$                 ;;SETUP TO TYPEOUT THE DELETED CHAR.
005352  104401  005516                          TYPE    ,9$                     ;;GO TYPE
005356  000746                                  BR      2$                      ;;GO READ ANOTHER CHAR.
005360  005716                          5$:     TST     (SP)                    ;;RUBOUT KEY SET?
005362  001406                                  BEQ     7$                      ;;BR IF NO
005364  112737  000134  005516                  MOVB    #'\,9$                  ;;TYPE A BACK SLASH
005372  104401  005516                          TYPE    ,9$
005376  005016                                  CLR     (SP)                    ;;CLEAR THE RUBOUT KEY
005400  122713  000025                  7$:     CMPB    #25,(R3)                ;;IS CHARACTER A CTRL U?
005404  001003                                  BNE     8$                      ;;BR IF NO
005406  104401  005527                          TYPE    ,$CNTLU                 ;;TYPE A CONTROL "U"
005412  000726                                  BR      1$                      ;;GO START OVER
005414  122713  000022                  8$:     CMPB    #22,(R3)                ;;IS CHARACTER A "↑R"?
005420  001011                                  BNE     3$                      ;;BRANCH IF NO
005422  105013                                  CLRB    (R3)                    ;;CLEAR THE CHARACTER
005424  104401  001313                          TYPE    ,$CRLF                  ;;TYPE A "CR" & "LF"
005430  104401  005520                          TYPE    ,$TTYIN                 ;;TYPE THE INPUT STRING
005434  000717                                  BR      2$                      ;;GO PICKUP ANOTHER CHACTER
005436  104401  001312                  4$:     TYPE    ,$QUES                  ;;TYPE A '?'
005442  000712                                  BR      1$                      ;;CLEAR THE BUFFER AND LOOP
005444  111337  005516                  3$:     MOVB    (R3),9$                 ;;ECHO THE CHARACTER
005450  104401  005516                          TYPE    ,9$
005454  122723  000015                          CMPB    #15,(R3)+               ;;CHECK FOR RETURN
005460  001305                                  BNE     2$                      ;;LOOP IF NOT RETURN
005462  105063  177777                          CLRB    -1(R3)                  ;;CLEAR RETURN (THE 15)
005466  104401  001314                          TYPE    ,$LF                    ;;TYPE A LINE FEED
005472  005726                                  TST     (SP)+                   ;;CLEAN RUBOUT KEY FROM THE STACK
005474  012603                                  MOV     (SP)+,R3                ;;RESTORE R3
005476  011646                                  MOV     (SP),-(SP)              ;;ADJUST THE STACK AND PUT ADDRESS OF THE
005500  016666  000004  000002                  MOV     4(SP),2(SP)             ;;      FIRST ASCII CHARACTER ON IT
005506  012766  005520  000004                  MOV     #$TTYIN,4(SP)
005514  000002                                  RTI                             ;;RETURN
005516     000                          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
005517     000                                  .BYTE   0                       ;;TERMINATOR
005520  000007                          $TTYIN: .BLKB   7                       ;;RESERVE 7 BYTES FOR TTY INPUT
005527     136  006525  000012          $CNTLU: .ASCIZ  /↑U/<15><12>            ;;CONTROL "U"
005534  043536  005015     000          $CNTLG: .ASCIZ  /↑G/<15><12>            ;;CONTROL "G"
005541     015  051412  051127          $MSWR:  .ASCIZ  <15><12>/SWR = /
005546  036440  000040
005552  020040  042516  020127          $MNEW:  .ASCIZ  /  NEW = /
005560  020075     000
        005564                          .EVEN
                                        .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

                                        ;;*************************************************************
                                        ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
                                        ;*CHANGE IT TO BINARY.
                                        ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
```

```
                                ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
                                ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
                                ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
                                ;*CALL:
                                ;*       RDOCT                   ;;READ AN OCTAL NUMBER
                                ;*       RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
                                ;*                               ;;HIGH ORDER BITS ARE IN $HIOCT

005564 011646                   $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
005566 016666 000004 000002             MOV     4(SP),2(SP)     ;;INPUT NUMBER
005574 010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
005576 010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
005600 010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
005602 104403                   1$:     RDLIN                   ;;READ AN ASCIZ LINE
005604 012600                           MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
005606 010037 005712                    MOV     R0,5$           ;;AND SAVE IT
005612 005001                           CLR     R1              ;;CLEAR DATA WORD
005614 005002                           CLR     R2
005616 112046                   2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
005620 001420                           BEQ     3$              ;;IF ZERO GET OUT
005622 122716 000060                    CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
005626 003026                           BGT     4$              ;;IS AN OCTAL DIGIT
005630 122716 000067                    CMPB    #'7,(SP)
005634 002423                           BLT     4$
005636 006301                           ASL     R1              ;;*2
005640 006102                           ROL     R2
005642 006301                           ASL     R1              ;;*4
005644 006102                           ROL     R2
005646 006301                           ASL     R1              ;;*8
005650 006102                           ROL     R2
005652 042716 177770                    BIC     #↑C7,(SP)       ;;STRIP THE ASCII JUNK
005656 062601                           ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
005660 000756                           BR      2$              ;;LOOP
005662 005726                   3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
005664 010166 000012                    MOV     R1,12(SP)       ;;SAVE THE RESULT
005670 010237 005722                    MOV     R2,$HIOCT
005674 012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
005676 012601                           MOV     (SP)+,R1        ;;POP STACK INTO R1
005700 012600                           MOV     (SP)+,R0        ;;POP STACK INTO R0
005702 000002                           RTI                     ;;RETURN
005704 005726                   4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
005706 105010                           CLRB    (R0)            ;;SET A TERMINATOR
005710 104401                           TYPE                    ;;TYPE UP THRU THE BAD CHAR.
005712 000000                   5$:     .WORD   0
005714 104401 001312                    TYPE    ,$QUES          ;;"?" "CR" & "LF"
005720 000730                           BR      1$              ;;TRY AGAIN
005722 000000                   $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
                                ;
                                ;       INPUT OCTAL NUMBER ROUTINE
                                ;
                                ;-------------------------------
                                ;
005724 010546                   $INPUT: MOV     R5,-(SP)                ; SAVE REGISTER R5.
005726 016605 000002                    MOV     2(SP),R5                ; GET FIRST PARAMETER ADDRESS.
005732 012537 005770                    MOV     (R5)+,WHAT              ; GET MESSAGE ADDRESS.
005736 012537 006050                    MOV     (R5)+,LOLIM             ; GET LOW LIMIT FOR THE  #
005742 012537 006052                    MOV     (R5)+,HILIM             ; GET HIGH LIMIT FOR THE #.
```

```
 005746  012537  006054                     MOV     (R5)+,WHERE             ; GET ADDRESS OF INBUFFER
 005752  112537  006056                     MOVB    (R5)+,LOBITS            ; GET LOWMASK BITS.
 005756  112537  006057                     MOVB    (R5)+,ADRCNT            ; GET # OF #'S TO BE GENERATED.
 005762  010566  000002                     MOV     R5,2(SP)                ; SAVE THE RETURN ADDRESS.
 005766  104401                     INLP1:  TYPE                            ; TYPE THE MESSAGE.
 005770  000000                     WHAT:   .WORD   0                       ;
 005772  104404                             RDOCT                           ; READ OCTAL # FROM KEYBOARD.
 005774  021637  006052                     CMP     (SP),HILIM              ; IS IT IN HIGH LIMIT?
 006000  003003                             BGT     2$                      ; BRANCH IF NO.
 006002  021637  006050                     CMP     (SP),LOLIM              ; IS IT MORE THAN LOW LIMIT.
 006006  002005                             BGE     3$                      ; BRANCH IF YES.
 006010  104401  001312             2$:     TYPE    ,$QUES                  ; TYPE " ? "
 006014  104401  001313                     TYPE    ,$CRLF                  ; TYPE <CR>,<LF>
 006020  000762                             BR      INLP1                   ;
 006022  013705  006054             3$:     MOV     WHERE,R5                ; GET BUFFER ADDRESS.
 006026  011625                     4$:     MOV     (SP),(R5)+              ; SAVE THE # IN RIGHT PLACE.
 006030  062716  000002                     ADD     #2,(SP)                 ; NEXT SEQUENTIAL NUMBER.
 006034  105337  006057                     DECB    ADRCNT                  ; COUNT BY 1.
 006040  001372                             BNE     4$                      ; BRANCH IF NOT DONE.
 006042  005726                             TST     (SP)+                   ; POP THE STACK POINTER.
 006044  012605                             MOV     (SP)+,R5                ; POP THE REG.5
 006046  000002                             RTI
 006050  000000                     LOLIM:  .WORD   0
 006052  000000                     HILIM:  .WORD   0
 006054  000000                     WHERE:  .WORD   0
 006056     000                     LOBITS: .BYTE   0
 006057     000                     ADRCNT: .BYTE   0


                                            ;  ADVANCE TO NEXT TEST HANDLER
                                            ;-----------------------------------

 006060  013716  001442             .ADVANCE:       MOV     NEXT,(SP)               ; CRUNCH STACK WITH ADDRESSOF SCOPE CALL
 006064  005037  001444                     CLR     LOCK                    ; RESET TIGHT LOOP ADDRESS
 006070  000002                             RTI                             ; CHECK TO SEE IF OLD TEST GETS REPEATED

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

 006072  016637  000004  001460     .SAV05: MOV     4(SP),SAVPC     ;SAVE R7 (PC)

                                            ;SAVE R0-R5

 006100  010537  001274             SV05:   MOV     R5,$REG5        ;SAVE R5
 006104  010437  001272                     MOV     R4,$REG4        ;SAVE R4
 006110  010337  001270                     MOV     R3,$REG3        ;SAVE R3
 006114  010237  001266                     MOV     R2,$REG2        ;SAVE R2
 006120  010137  001264                     MOV     R1,$REG1        ;SAVE R1
 006124  010037  001262                     MOV     R0,$REG0        ;SAVE R0
 006130  000002                             RTI                     ;LEAVE.

                                            ;RESTORE R0-R5

 006132  013700  001262             .RES05: MOV     $REG0,R0        ;RESTORE R0
 006136  013701  001264                     MOV     $REG1,R1        ;RESTORE R1
 006142  013702  001266                     MOV     $REG2,R2        ;RESTORE R2
 006146  013703  001270                     MOV     $REG3,R3        ;RESTORE R3
```

```
006152  013704  001272                    MOV     $REG4,R4                ;RESTORE R4
006156  013705  001274                    MOV     $REG5,R5                ;RESTORE R5
006162  000002                            RTI                             ;LEAVE

                                  ;       ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
                                  ;       ;-------------------------------------------------------
                                  ;
006164  104401  001313            .CONVR: TYPE    ,$CRLF
006170  010046                    .CNVRT: MOV     R0,-(SP)
006172  010146                            MOV     R1,-(SP)
006174  010346                            MOV     R3,-(SP)
006176  010446                            MOV     R4,-(SP)
006200  010546                            MOV     R5,-(SP)
006202  017601  000012                    MOV     @12(SP),R1
006206  062766  000002  000012            ADD     #2,12(SP)
006214  012137  006406                    MOV     (R1)+,WRDCNT
006220  112137  006410            1$:     MOVB    (R1)+,CHRCNT
006224  112137  006411                    MOVB    (R1)+,SPACNT
006230  013137  006412                    MOV     @(R1)+,BINWRD
006234  122737  000003  006410            CMPB    #3,CHRCNT
006242  001003                            BNE     2$
006244  042737  177400  006412            BIC     #177400,BINWRD
006252  013704  006412            2$:     MOV     BINWRD,R4
006256  113705  006410                    MOVB    CHRCNT,R5
006262  012700  011106                    MOV     #TEMP,R0
006266  010403                    3$:     MOV     R4,R3
006270  042703  177770                    BIC     #177770,R3
006274  062703  000060                    ADD     #060,R3
006300  110320                            MOVB    R3,(R0)+
006302  000241                            CLC
006304  006004                            ROR     R4
006306  000241                            CLC
006310  006004                            ROR     R4
006312  000241                            CLC
006314  006004                            ROR     R4
006316  005305                            DEC     R5
006320  001362                            BNE     3$
006322  012703  011150                    MOV     #MDATA,R3
006326  114023                    4$:     MOVB    -(R0),(R3)+
006330  105337  006410                    DECB    CHRCNT
006334  001374                            BNE     4$
006336  105737  006411                    TSTB    SPACNT
006342  001405                            BEQ     6$
006344  112723  000040            5$:     MOVB    #040,(R3)+
006350  105337  006411                    DECB    SPACNT
006354  001373                            BNE     5$
006356  105013                    6$:     CLRB    (R3)
006360  104401  011150                    TYPE    ,MDATA
006364  005337  006406                    DEC     WRDCNT
006370  001313                            BNE     1$
006372  012605                            MOV     (SP)+,R5
006374  012604                            MOV     (SP)+,R4
006376  012603                            MOV     (SP)+,R3
006400  012601                            MOV     (SP)+,R1
006402  012600                            MOV     (SP)+,R0
006404  000002                            RTI
```

```
006406  000000                          WRDCNT: 0
006410  000000                          CHRCNT: 0
        006411                          SPACNT=CHRCNT+1
006412  000000                          BINWRD: 0



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

                                        .SBTTL  TRAP DECODER

                                        ;;*****************************************************************
                                        ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                        ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                        ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                        ;*GO TO THAT ROUTINE.

006414  010046                          $TRAP:  MOV     R0,-(SP)                ;;SAVE R0
006416  016600  000002                          MOV     2(SP),R0                ;;GET TRAP ADDRESS
006422  005740                                  TST     -(R0)                   ;;BACKUP BY 2
006424  111000                                  MOVB    (R0),R0                 ;;GET RIGHT BYTE OF TRAP
006426  006300                                  ASL     R0                      ;;POSITION FOR INDEXING
006430  016000  006450                          MOV     $TRPAD(R0),R0           ;;INDEX TO TABLE
006434  000200                                  RTS     R0                      ;;GO TO ROUTINE


                                        ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

006436  011646                          $TRAP2: MOV     (SP),-(SP)              ;;MOVE THE PC DOWN
006440  016666  000004  000002                  MOV     4(SP),2(SP)             ;;MOVE THE PSW DOWN
006446  000002                                  RTI                             ;;RESTORE THE PSW

                                        .SBTTL  TRAP TABLE

                                        ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                        ;*BY THE "TRAP" INSTRUCTION.

                                        ;       ROUTINE
                                                -------
006450  006436                          $TRPAD: .WORD   $TRAP2
006452  004414                                  $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE


006454  005144                                  $RDCHR  ;;CALL=RDCHR    TRAP+2(104402)  TTY TYPEIN CHARACTER ROUTINE
006456  005264                                  $RDLIN  ;;CALL=RDLIN    TRAP+3(104403)  TTY TYPEIN STRING ROUTINE
006460  005564                                  $RDOCT  ;;CALL=RDOCT    TRAP+4(104404)  READ AN OCTAL NUMBER FROM TTY
006462  004364                                  .SCOP1  ;;CALL=SCOP1    TRAP+5(104405)  CALL TO LOOP ON CURRENT DATA HANDLER
006464  006072                                  .SAVOS  ;;CALL=SAVOS    TRAP+6(104406)  CALL TO REGISTER SAVE ROUTINE
006466  006132                                  .RESOS  ;;CALL=RESOS    TRAP+7(104407)  CALL TO REGISTER RESTORE ROUTINE
006470  007362                                  .MSTCLR ;;CALL=MSTCLR   TRAP+10(104410) CALL TO ISSUE A MASTER CLEAR
006472  007332                                  .DELAY  ;;CALL=DELAY    TRAP+11(104411) CALL TO DELAY
006474  007400                                  .ROMCLK ;;CALL=ROMCLK   TRAP+12(104412) CALL TO CLOCK ROM ONCE
006476  007446                                  .DATACLK        ;;CALL=DATACLK  TRAP+13(104413) CALL TO CLOCK DATA
006500  007512                                  .TIMER  ;;CALL=TIMER    TRAP+14(104414) CALL TO DELAY A CLOCK TICK
```

```
006502  005724                                   $INPUT  ;;CALL=INPUT     TRAP+15(104415) CALL TO OCTAL # INPUT ROUTINE
006504  006164                                   .CONVRT ;;CALL=CONVRT    TRAP+16(104416) CALL TO     .....
006506  006170                                   .CNVRT  ;;CALL=CNVRT     TRAP+17(104417) CALL TO     .......
006510  006060                                   .ADVANCE        ;;CALL=ADVANCE  TRAP+20(104420) CALL TO ADVANCE TO NEXT TEST

                                        ;
                                        ;-------------------------------------------------------------------------
                                        ;;***********************************************************************
                                                 ;ERROR HANDLER
                                                 ;-------------
                                                 ;
006512  004737  011212                  $ERROR:  JSR     PC,CKSWR               ;CHECK FOR SOFT SWR
006516  032777  010000  172514                   BIT     #SW12,@SWR             ;BELL ON ERROR?
006524  001406                                   BEQ     XBX                    ;BR IF NO BELL
006526  105777  172516                           TSTB    @STPS                  ;TTY READY.
006532  100003                                   BPL     XBX                    ;DON'T WAIT IF TTY NOT READY.
006534  112777  000207  172510                   MOVB    #207,@STPB             ;PUSH A BELL AT THE TTY.
006542  032777  020000  172470  XBX:             BIT     #SW13,@SWR             ;DELETE ERROR PRINT OUT?
006550  001107                                   BNE     HALTS                  ;BR IF NO PRINT OUT WANTED.
006552  021637  001216                           CMP     (SP),$ERRPC            ;WAS THIS ERROR FOUND LAST TIME?
006556  001404                                   BEQ     1$                     ;BR IF YES
006560  011637  001216                           MOV     (SP),$ERRPC            ;RECORD BEING HERE
006564  105037  001203                           CLRB    $ERFLG                 ;PREPARE HEADER
006570  104406                          1$:      SAV05                          ;SAVE ALL PROC REGISTERS
006572  011605                                   MOV     (SP),R5                ;GET THE PC OF ERROR
006574  162705  000002                           SUB     #2,R5                  ;GET ADDRESS OF TRAP CALL
006600  011504                                   MOV     (R5),R4                ;GET ERROR INSTRUCTION
006602  110437  001214                           MOVB    R4,$ITEMB              ; COPY ERROR # FOR APT HANDLING
006606  006304                                   ASL     R4                     ;MULT BY TWO
006610  061504                                   ADD     (R5),R4                ;DOUBLE IT
006612  006304                                   ASL     R4                     ;MULT AGAIN
006614  042704  177001                           BIC     #177001,R4             ;CLEAR JUNK
006620  062704  001512                           ADD     #$ERRTB,R4             ;GET POINTER
006624  012437  006740                           MOV     (R4)+,ERRMSG           ;GET ERROR MESSAGE
006630  012437  006752                           MOV     (R4)+,DATAHD           ;GET DATA HEADRER
006634  011437  006764                           MOV     (R4),DATABP            ;GET DATA TABLE
006640  105737  001203                           TSTB    $ERFLG                 ;TYPE HEADREER
006644  001403                                   BEQ     TYPMSG                 ;BR IF YES
006646  005737  006764                           TST     DATABP                 ;DOES DATA TABLE EXIST?
006652  001040                                   BNE     TYPDAT                 ;BR IF YES.
006654  104401  001313                  TYPMSG:  TYPE    ,$CRLF
006660  104401  001313                           TYPE    ,$CRLF
006664  005737  001444                           TST     LOCK
006670  001402                                   BEQ     1$
006672  104401  010015                           TYPE    ,MASTEK
006676  104401  010003                  1$:      TYPE    ,MTSTN
006702  104417  007120                           CNVRT   ,XTSTN                 ;SHOW IT
006706  104401  010072                           TYPE    ,MERRPC                ;TYPE PC.
006712  104417  007112                           CNVRT   ,ERTABO                ;SHOW IT
006716  104401  001313                           TYPE    ,$CRLF                 ;GIVE A CR/LF
006722  112737  177777  001203                   MOVB    #-1,$ERFLG             ;NO MORE HEADER UNLESS NO DATA TABLE.
006730  005737  006740                           TST     ERRMSG                 ;IS THERE AN ERROR MESSAGE?
006734  001402                                   BEQ     WRKO.FM                ;BR IF NO.
006736  104401                                   TYPE                           ;TYPE
006740  000000                          ERRMSG:  0                              ;      ERROR MESSAGE
006742                                  WRKO.FM:
006742  005737  006752                           TST     DATAHD                 ;DATA HEADER?
```

```
006746  001402                            BEQ     TYPDAT           ;BR IF NO
006750  104401                            TYPE                     ;TYPE
006752  000000                   DATAHD:  0                        ;      DATA HEADER
006754  005737  006764           TYPDAT:  TST     DATABP           ;DATA TABLE?
006760  001402                            BEQ     RESREG           ;BR IF NO.
006762  104416                            CONVRT                   ;SHOW
006764  000000                   DATABP:  0                        ;      DATA TABLE
006766  104407                   RESREG:  RES05                    ;RESTORE PROC REGISTERS
006770  122737  000001  001336   HALTS:   CMPB    #APTENV,$ENV     ; IS APT RUNNING ?
006776  001007                            BNE     3$               ; SKIP APT CALL IF NOT.
007000  113737  001214  007012            MOVB    $ITEMB,6$        ; COPY ERROR #.
007006  004737  004714                    JSR     PC,$ATY4         ; CALL APT SERVICES.
007012  000000                   6$:      .WORD   0                ; ERROR # GOES HERE.
007014  000777                   9$:      BR      9$               ; LOCK HERE.
007016  022737  004070  000042   3$:      CMP     #$ENDAD,@#42     ;IF ACT-11 AUTOMATIC MODE, HALT!!
007024  001403                            BEQ     1$
007026  005777  172206                    TST     @SWR             ;HALT ON ERROR?
007032  100005                            BPL     EXITER           ;BR IF NO HALT ON ERROR
007034  010046                   1$:      PUSHR0                   ;SAVE R0
007036  016600  000002                    MOV     2(SP),R0         ;SHOW ERROR PC IN DATA LIGHTS
007042  000000                            HALT                     ;HALT
007044  012600                            POPR0                    ;GET R0
007046  005237  001212           EXITER:  INC     $ERTTL           ;UPDATE ERROR COUNT
007052  032777  000400  172160            BIT     #SW08,@SWR       ;GOTO TOP OF TEST?
007060  001007                            BNE     1$               ;BR IF YES
007062  032777  002000  172150            BIT     #SW10,@SWR       ;GOTO NEXT TEST?
007070  001407                            BEQ     2$               ;BR IF NO
007072  013737  001442  001206            MOV     NEXT,$LPADR      ;SET FOR NEXT TEST
007100  012706  001200           1$:      MOV     #STACK,SP        ;RESET SP
007104  000177  172076                    JMP     @$LPADR          ;GOTO SPECIFIED TEST
007110  000002                   2$:      RTI                      ;$LPADR
007112  000001                   ERTABO:  1
007114     006     002                    .BYTE   6,2
007116  001460                            $AVPC
007120  000001                   XTSTN:   1
007122     003     002                    .BYTE   3,2
007124  001202                            $TSTNM
                                          ;ENTER HERE ON POWER FAILURE
                                          ;---------------------------

                                 .SBTTL   POWER DOWN AND UP ROUTINES

                                 ;;*****************************************************************
                                 ;POWER DOWN ROUTINE
007126  012737  007316  000024   $PWRDN:  MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
007134  012737  000340  000026            MOV     #340,@#PWRVEC+2  ;;PRIO:7
007142  010046                            MOV     R0,-(SP)         ;;PUSH R0 ON STACK
007144  010146                            MOV     R1,-(SP)         ;;PUSH R1 ON STACK
007146  010246                            MOV     R2,-(SP)         ;;PUSH R2 ON STACK
007150  010346                            MOV     R3,-(SP)         ;;PUSH R3 ON STACK
007152  010446                            MOV     R4,-(SP)         ;;PUSH R4 ON STACK
007154  010546                            MOV     R5,-(SP)         ;;PUSH R5 ON STACK
007156  017746  172056                    MOV     @SWR,-(SP)       ;;PUSH @SWR ON STACK
007162  010637  007322                    MOV     SP,$SAVR6        ;;SAVE SP
007166  012737  007200  000024            MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
007174  000000                            HALT
```

```
007176  000776                                  BR      .-2                 ;;HANG UP

                                        ;;*********************************************************************
                                        ;POWER UP ROUTINE
007200  012737  007316  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC    ;;SET FOR FAST DOWN
007206  013706  007322                          MOV     $SAVR6,SP           ;;GET SP
007212  005037  007322                          CLR     $SAVR6              ;;WAIT LOOP FOR THE TTY
007216  005237  007322          1$:             INC     $SAVR6              ;;WAIT FOR THE INC
007222  001375                                  BNE     1$                  ;;OF  WORD
007224  104401  007562                          TYPE    ,MPFAIL             ;
007230  104417  007324                          CNVRT   ,PFTAB              ;
007234  105037  001203                          CLRB    $ERFLG              ;CLEAR ERROR FLAG.
007240  005037  001216                          CLR     $ERRPC              ; CLEAR LAST ERROR PC
007244  013701  002066                          MOV     KMCSR,R1            ; RESTORE DEVICE ADDRESS.
007250  005011                                  CLR     (R1)                ; CLEAR THE CSR.
007252  104410                                  MSTCLR                      ;
007254  012677  171760                          MOV     (SP)+,@SWR          ;;POP STACK INTO @SWR
007260  012605                                  MOV     (SP)+,R5            ;;POP STACK INTO R5
007262  012604                                  MOV     (SP)+,R4            ;;POP STACK INTO R4
007264  012603                                  MOV     (SP)+,R3            ;;POP STACK INTO R3
007266  012602                                  MOV     (SP)+,R2            ;;POP STACK INTO R2
007270  012601                                  MOV     (SP)+,R1            ;;POP STACK INTO R1
007272  012600                                  MOV     (SP)+,R0            ;;POP STACK INTO R0
007274  012737  007126  000024                  MOV     #$PWRDN,@#PWRVEC    ;;SET UP THE POWER DOWN VECTOR
007302  012737  000340  000026                  MOV     #340,@#PWRVEC+2     ;;PRIO:7
007310  104401                                  TYPE                        ;;REPORT THE POWER FAILURE
007312  007562                  $PWRMG: .WORD   MPFAIL                      ;;POWER FAIL MESSAGE POINTER
007314  000002                                  RTI
007316  000000                  $ILLUP: HALT                                ;;THE POWER UP SEQUENCE WAS STARTED
007320  000776                                  BR      .-2                 ;; BEFORE THE POWER DOWN WAS COMPLETE
007322  000000                  $SAVR6: 0                                   ;;PUT THE SP HERE

007324  000001                  PFTAB:  1
007326     003     002          .BYTE   3,2
007330  001202                          $TSTNM

007332                          .DELAY:
007332  012777  000020  172534          MOV     #20,@KMP04
007340  104412                          ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
007342  121111                          121111                      ;POKE CLOCK DELAY BIT
007344                          1$:
007344  104412                          ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
007346  121224                          121224                      ;PORT4+IBUS*11
007350  032777  000020  172516          BIT     #BIT4,@KMP04        ;IS CLOCK BIT SET?
007356  001772                          BEQ     1$                  ;BR IF NO
007360  000002                          RTI

007362                          .MSTCLR:
007362  152777  000100  172500          BISB    #BIT6,@KMCSRH       ;SET MASTER CLEAR
007370  142777  000300  172472          BICB    #BIT6!BIT7,@KMCSRH  ;CLEAR MASTER CLEAR AND RUN
007376  000002                          RTI                         ;RETURN

007400                          .ROMCLK:
007400  152777  000002  172462          BISB    #BIT1,@KMCSRH       ;SET ROMI
007406  013677  172464                  MOV     @(SP)+,@KMP06       ;LOAD INSTRUCTION IN SEL6
007412  062746  000002                  ADD     #2,-(SP)            ;ADJUST STACK
```

```
007416  032777  000100  171614              BIT     #SW06,@SWR          ;HALT IF SW06 =1
007424  001401                              BEQ     1$                  ;BR IF SW06 =0
007426  000000                              HALT                        ;HALT BEFORE CLOCKING INSTRUCTION
007430  152777  000003  172432  1$:         BISB    #BIT1!BIT0,@KMCSRH  ;CLOCK INSTRUCTION
007436  142777  000007  172424              BICB    #BIT2!BIT1!BIT0,@KMCSRH  ;CLEAR ROM0, ROM1, STEP
007444  000002                              RTI

007446                          .DATACLK:
007446  013637  011106                      MOV     @(SP)+,TEMP         ;PUT TICK COUNT IN TEMP
007452  062746  000002                      ADD     #2,-(SP)            ;ADJUST STACK
007456  152777  000020  172404  1$:         BISB    #BIT4,@KMCSRH       ;SET STEP LU
007464  027777  172376  172374              CMP     @KMCSR,@KMCSR       ;WASTE TIME
007472  142777  000020  172370              BICB    #BIT4,@KMCSRH       ;CLEAR STEP LU
007500  005337  011106                      DEC     TEMP                ;DEC TICK COUNT
007504  001364                              BNE     1$                  ;BR IF NOT DONE
007506  000002                              RTI                         ;RETURN
007510  000001                  3$:         .BLKW 1

007512                          .TIMER:
007512  013637  011106                      MOV     @(SP)+,TEMP         ;MOVE COUNT TO TEMP
007516  062746  000002                      ADD     #2,-(SP)            ;ADJUST STACK
007522                          1$:
007522  104412                              ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
007524  021364                              021364                      ;PORT4+IBUS* REG11
007526  032777  000002  172340              BIT     #2,@KMP04           ;IS PGM CLOCK BIT CLEAR?
007534  001772                              BEQ     1$                  ;BR IF YES
007536                          2$:
007536  104412                              ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
007540  021364                              021364                      ;PORT4+IBUS* REG11
007542  032777  000002  172324              BIT     #2,@KMP04           ;IS PGM CLOCK BIT SET?
007550  001372                              BNE     2$                  ;BR IF YES
007552  005337  011106                      DEC     TEMP                ;DEC COUNT
007556  001361                              BNE     1$                  ;BR IF NOT DONE
007560  000002                              RTI                         ;RETURN

007562  050200  051127  043040  MPFAIL:  .ASCIZ  <200>/PWR FAILED. RESTART AT TEST /
007620  042600  042116  050040  MEPASS:  .ASCIZ  <200>/END PASS DZKCE  /
007642  051200     000          MR:      .ASCIZ  <200>/R/
007645     200  047516  042040  MERR2:   .ASCIZ  <200>/NO DEVICES PRESENT./
007672  044600  051516  043125  MERR3:   .ASCIZ  <200>/INSUFFICIENT DATA!/
007716  046200  041517  020113  MLOCK:   .ASCIZ  <200>/LOCK ON SELECTED TEST/
007745     103  051123  020072  MCSRX:   .ASCIZ   /CSR: /
007753     126  041505  020072  MVECX:   .ASCIZ  /VEC: /
007761     120  051501  042523  MPASSX:  .ASCIZ  /PASSES: /
007772  051105  047522  051522  MERRX:   .ASCIZ  /ERRORS: /
010003     124  051505  020124  MTSTN:   .ASCIZ  /TEST NO: /
010015     052     000          MASTEK:  .ASCIZ  /*/
010017     200  042523  020124  MNEW:    .ASCIZ  <200>/SET SWITCH REG TO KMC11'S DESIRED ACTIVE./
010072  041520  020072     000  MERRPC:  .ASCIZ  /PC: /
010077     200  020040  020040  XHEAD:   .ASCII  <200>/              MAP OF KMC11 STATUS/
010136  020200  020040  020040           .ASCII  <200>/              --------------------/
010175     200  020040  041520           .ASCII  <200>/  PC      CSR      STAT1     STAT2     STAT3/
010247     200  026455  026455           .ASCIZ  <200>/------  ------   ------    ------    ------/
010323     200  047510  020127  NUM:     .ASCIZ  <200>/HOW MANY KMC11'S TO BE TESTED?/
010363     200  051503  020122  CSR:     .ASCIZ  <200>/CSR ADDRESS?/
010401     200  042526  052103  VEC:     .ASCIZ  <200>/VECTOR ADDRESS?/
```

```
 (2)  010422  041200  020122  051120  PRIO:   .ASCIZ  <200>/BR PRIORITY LEVEL? (4,5,6,7)?/
 (2)  010461     200  044127  041511  MODU:   .ASCIZ  <200>/WHICH LINE UNIT? IF NONE TYPE "N", IF M8201 TYPE "1", IF M8202 TYP
 (2)  010573     200  053523  052111  LINE:   .ASCIZ  <200>/SWITCH PAC#1 (DDCMP LINE #)?/
 (2)  010631     200  053523  052111  BM:     .ASCIZ  <200>/SWITCH PAC#2 (BM873 BOOT ADD)?/
 (2)  010671     200  051511  052040  CONN:   .ASCIZ  <200>/IS THE LOOP BACK CONNECTOR ON?/
 (2)  010731     200  047516  042040  NOACT:  .ASCIZ  <200>/NO DEVICES ARE SELECTED/
 (2)  010762  100200  046513  030503  CONERR: .ASCIZ  <200><200>/KMC11 AT NONSTANDARD ADDRESS  PC: /
 (2)  011027     200  054105  042520  CNERR:  .ASCIZ  <200>/EXPECTED  FOUND/
 (2)  011050  024040  046513  024503  KMCM:   .ASCIZ  / (KMC) /
 (2)                                          .EVEN
 (2)  011060  000005                          XSTATQ: 5
      011062     006     003                          .BYTE   6,3
      011064  001276                                  $TMP0
      011066     006     003                          .BYTE   6,3
      011070  001300                                  $TMP1
      011072     006     003                          .BYTE   6,3
      011074  001302                                  $TMP2
      011076     006     003                          .BYTE   6,3
      011100  001304                                  $TMP3
      011102     006     002                          .BYTE   6,2
      011104  001306                                  $TMP4
                                              .EVEN

                                                      ;BUFFERS FOR INPUT-OUTPUT

      011106  000000                          TEMP:   0
              011150                          .=.+40
      011150  000000                          MDATA:  0
              011212                          .=.+40



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

      011212  022737  000176  001240  CKSWR:  CMP     #SWREG,SWR      ;IS THE SOFT SWR BEING USED?
      011220  001075                          BNE     CKSWRS          ;BR IF NO
      011222  132737  000001  001336          BITB    #1,$ENV         ; IS IT RUNNING UNDER APT?
      011230  001071                          BNE     CKSWRS          ; EXIT IF YES.
      011232  022777  000007  170006          CMP     #7,@$TKB        ;WAS CTRL G TYPED? (7 BIT ASCII)
      011240  001404                          BEQ     1$              ;BR IF YES
      011242  022777  000207  167776          CMP     #207,@$TKB      ;WAS CTRL G TYPED? (8 BIT ASCII)
      011250  001061                          BNE     CKSWRS          ;BR IF NO
      011252  010246                  1$:     MOV     R2,-(SP)        ;STORE R2
      011254  010346                          MOV     R3,-(SP)        ;STORE R3
      011256  010446                          MOV     R4,-(SP)        ;STORE R4
      011260  012737  177777  011416          MOV     #-1,SWFLG       ;SET SOFT TYPE OUT FLAG
      011266  005002                  CKSWR1: CLR     R2              ;CLEAR NEW SWR CONTENTS
      011270  012704  177777                  MOV     #-1,R4          ;SET FLAG TO ALL ONES
      011274  104401  005541                  TYPE    ,$MSWR          ;TYPE "SWR= "
      011300  104417                  CKSWR2: CNVRT                   ;TYPE OUT PRESENT CONTENTS
      011302  011452                          SOFTSW                  ;OF SOFT SWITCH REGISTER
      011304  104401  005552          CKSWR3: TYPE    ,$MNEW          ;TYPE "NEW? "
      011310  004737  011420          CKSWR4: JSR     PC,INCHAR       ;GET RESPONSE
      011314  022703  000015                  CMP     #15,R3          ;WAS IT A CR?
      011320  001424                          BEQ     5$              ;BR IF YES
```

```
011322  022703  000012                    CMP     #12,R3             ;WAS IT A LF?
011326  001416                            BEQ     4$                 ;BR IF YES
011330  022703  000025                    CMP     #25,R3             ;WAS IT CTRL U?
011334  001754                            BEQ     CKSWR1             ;BR IF YES(START OVER)
011336  022703  000007                    CMP     #7,R3              ;IF CNTL G GET NEXT CHAR
011342  001762                            BEQ     CKSWR4
011344  005004                            CLR     R4                 ;IT MUST BE A DIGIT SO CLR FLAG
011346  042703  177770                    BIC     #177770,R3         ;ONLY 0-7 ARE LEGAL SO MASK OFF BITS
011352  006302                            ASL     R2                 ;SHIFT R2 3 TIMES
011354  006302                            ASL     R2
011356  006302                            ASL     R2
011360  050302                            BIS     R3,R2              ;ADD LAST DIGIT
011362  000752                            BR      CKSWR4             ;GET NEXT CHARACTER
011364  012766  002402  000006  4$:       MOV     #.START,6(SP)      ;LF WAS TYPED SO GO TO START
011372  005704                  5$:       TST     R4                 ;IS FLAG CLEAR?
011374  001002                            BNE     6$                 ;IF NOT DON'T CHANGE SOFT SWR
011376  010277  167636                    MOV     R2,@SWR            ;IF YES THEN WRITE NEW CONTENTS TO SOFT SWR
011402  005037  011416          6$:       CLR     SWFLG              ;CLEAR TYPEOUT FLAG
011406  012604                            MOV     (SP)+,R4           ;RESTORE R4
011410  012603                            MOV     (SP)+,R3           ;RESTORE R3
011412  012602                            MOV     (SP)+,R2           ;RESTORE R2
011414  000207                  CKSWR5:   RTS     PC                 ;RETURN

011416  000000                  SWFLG:    0

011420  105777  167620          INCHAR:   TSTB    @$TKS
011424  100375                            BPL     .-4
011426  017703  167614                    MOV     @$TKB,R3
011432  105777  167612                    TSTB    @$TPS
011436  100375                            BPL     .-4
011440  010377  167606                    MOV     R3,@$TPB
011444  042703  000200                    BIC     #BIT7,R3
011450  000207                            RTS     PC

011452  000001                  SOFTSW:   1
011454     006     002                    .BYTE   6,2
011456  000176                            SWREG
```

```

                                                   ;
                                                   ;ROUTINE USED TO "CYCLE" THROUGH UP TO 16 KMC11'S
                                                   ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
                                                   ;AND RUNS THE SPECIFIED KMC11'S.   THIS ROUTINE *MUST*
                                                   ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
                                                   ;SETUP NECESSARY.
                                                   ;

011460  005737  001470           CYCLE:  TST     KMACTV              ;ARE ANY KMC11'S TO BE TESTED?
011464  001004                           BNE     1$                  ;BR IF OK.
011466  104401  010731                   TYPE    ,NOACT              ;NO KMC11'S SELECTED!!
011472  000000                           HALT                        ;STOP THE SHOW.
011474  000776                           BR      .-2                 ;DISQUALIFY CONT. SW.
011476  000241                   1$:     CLC                         ;CLEAR PROC. CARRY BIT.
011500  006137  001500                   ROL     RUN                 ;UPDATE POINTER
011504  005537  001500                   ADC     RUN                 ;CATCH CARRY FROM RUN
011510  062737  000004  001504           ADD     #4,MILK             ;UPDATE POINTER
011516  062737  000010  001502           ADD     #10,CREAM           ;UPDATE ADDRESS POINTER.
011524  022737  002300  001502           CMP     #KM.MAP+200,CREAM
011532  001006                           BNE     2$                  ;KEEP GOING; NOT ALL TESTED FOR.
011534  012737  002100  001502           MOV     #KM.MAP,CREAM       ;RESET ADDRESS POINTER.
011542  012737  002302  001504           MOV     #CNT.MAP,MILK       ;RESET PASS COUNT POINTER
011550  033737  001500  001470   2$:     BIT     RUN,KMACTV          ;IS THIS ONE ACTIVE?
011556  001747                           BEQ     1$                  ;BR IF NO
011560  013700  001502                   MOV     CREAM,R0            ;GET ADDRESS POINTER
011564  013702  001504                   MOV     MILK,R2             ;GET PASS COUNT POINTER
011570  012037  002066                   MOV     (R0)+,KMCSR         ;LOAD SYSTEM CTRL. REG
011574  011037  002056                   MOV     (R0),KMRVEC         ;LOAD VECTOR
011600  042737  177000  002056           BIC     #177000,KMRVEC      ;CLEAR UNWANTED BITS
011606  012037  002050                   MOV     (R0)+,STAT1         ;LOAD STAT1
011612  012037  002052                   MOV     (R0)+,STAT2         ;LOAD STAT2
011616  012037  002054                   MOV     (R0)+,STAT3         ;LOAD STAT3
011622  012237  001324                   MOV     (R2)+,$PASS         ;LOAD PASS COUNT
011626  012237  001212                   MOV     (R2)+,$ERTTL        ;LOAD ERROR COUNT
011632  012700  000002                   MOV     #2,R0               ;SAVE CORE THIS WAY!
011636  013737  002066  002070           MOV     KMCSR,KMCSRH
011644  005237  002070                   INC     KMCSRH
011650  013737  002070  002072           MOV     KMCSRH,KMCTL
011656  005237  002072                   INC     KMCTL
011662  013737  002072  002074           MOV     KMCTL,KMP04
011670  060037  002074                   ADD     R0,KMP04
011674  013737  002074  002076           MOV     KMP04,KMP06
011702  060037  002076                   ADD     R0,KMP06

011706  013737  002056  002060           MOV     KMRVEC,KMRLVL       ;PTY LVL
011714  060037  002060                   ADD     R0,KMRLVL           ;
011720  013737  002060  002062           MOV     KMRLVL,KMTVEC       ;TX VEC
011726  060037  002062                   ADD     R0,KMTVEC           ;
011732  013737  002062  002064           MOV     KMTVEC,KMTLVL       ;TX LVL
011740  060037  002064                   ADD     R0,KMTLVL

011744  032737  000002  001446           BIT     #SW01,STRTSW        ;IS TEST NO. SELECTED
011752  001447                           BEQ     7$                  ;BR IF NO
011754                           4$:
011754  005737  000042                   TST     @#42                ;RUNNING IN AUTO MODE?
```

```
011760  001044                                   BNE     7$                  ;BR IF YES
011762  104401  001313                           TYPE    ,$CRLF
011766  104415                                   INPUT
011770  010003                                   MTSTN
011772  000001                                   1
011774  001000                                   1000
011776  001202                                   $TSTNM
012000     000                           .BYTE   0
012001     001                           .BYTE   1
012002  012700  013732                           MOV     #TST1,R0
012006  022710                           5$:     CMP     (PC)+,(R0)          ;CMP FIRST WORD TO 12737
012010  012737                                   MOV     (PC)+,@(PC)+
012012  001020                                   BNE     6$                  ;BR IF NOT SAME
012014  023760  001202  000002                   CMP     $TSTNM,2(R0)        ;DOES $TSTNM MATCH?
012022  001014                                   BNE     6$                  ;BR IF NO
012024  022760  001202  000004                   CMP     #$TSTNM,4(R0)       ;IS LAST WORD OK?
012032  001010                                   BNE     6$                  ;BR IF NO
012034  010037  001206                           MOV     R0,$LPADR           ;IT IS A LEGAL TEST SO DO IT
012040  104401  007642                           TYPE    ,MR
012044  042737  000002  001446                   BIC     #SW01,STRTSW
012052  000412                                   BR      8$
012054  005720                           6$:     TST     (R0)+               ;POP R0
012056  020027  027444                           CMP     R0,#TLAST+10        ;AT END YET?
012062  001351                                   BNE     5$                  ;BR IF NO
012064  104401  001312                           TYPE    ,$QUES              ;YES ILLEGAL TEST NO.
012070  000731                                   BR      4$                  ;TRY AGAIN

012072  012737  013732  001206           7$:     MOV     #TST1,$LPADR        ;PREPARE $LPADR ADDRESS
012100  013701  002066                   8$:     MOV     KMCSR,R1            ;R1 = BASE KMC11 ADDRESS
012104  000177  167076                           JMP     @$LPADR             ;GO START TESTING.


                                                 ;ROUTINE USED TO "AUTO SIZE" THE KMC11
                                                 ;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)
                                                 ;

012110                                   AUTO.SIZE:
012110  000005                                   RESET                       ;INSURE A BUS INIT.
012112  012702  002100                   CSRMAP: MOV     #KM.MAP,R2          ;LOAD MAP POINTER.
012116  005022                           1$:     CLR     (R2)+               ;ZERO ENTIRE MAP
012120  022702  002300                           CMP     #KM.END,R2          ;ALL DONE?
012124  001374                                   BNE     1$                  ;BR IF NO
012126  005037  001472                           CLR     KMNUM               ;SET OCTAL NUMBER OF KMC11'S TO 0
012132  012702  002100                           MOV     #KM.MAP,R2          ;R2 POINTS TO KMC MAP
012136  005037  001470                           CLR     KMACTV              ;CLEAR ACTIVE
012142  032737  000001  001446                   BIT     #SW00,STRTSW        ;QUESTIONS?
012150  001002                                   BNE     .+6                 ;BR IF YES
012152  000137  012532                           JMP     7$                  ;IF NO SKIP QUESTIONS
012156  012737  000001  001306                   MOV     #1,$TMP4            ;START WITH 1
012164  104415                                   INPUT
012166  010323                                   NUM
012170  000001                                   1
```

```
012172  000020                                   16.
012174  001302                                   $TMP2
012176     000                                   .BYTE   0
012177     001                                   .BYTE   1
012200  013737  001302  001472                   MOV     $TMP2,KMNUM     ;KMNUM = HOW MANY
012206  104401  001313                  12$:     TYPE    ,$CRLF
012212  104416                                   CONVRT                  ;TYPE WHICH KMC IS BEING DONE
012214  013164                                   WHICH                   ;$TMP4 IS WHICH KMC
012216  005237  001306                           INC     $TMP4
012222  104415                                   INPUT
012224  010363                                   CSR
012226  160000                                   160000
012230  164000                                   164000
012232  001304                                   $TMP3
012234     000                                   .BYTE   0
012235     001                                   .BYTE   1
012236  013722  001304                           MOV     $TMP3,(R2)+     ;STORE CSR IN MAP
012242  104415                                   INPUT
012244  010401                                   VEC
012246  000000                                   0
012250  000776                                   776
012252  001304                                   $TMP3
012254     000                                   .BYTE   0
012255     001                                   .BYTE   1
012256  013712  001304                           MOV     $TMP3,(R2)      ;STORE VECTOR IN MAP
012262  104401                          10$:     TYPE
012264  010422                                   PRIO                    ;ASK WHAT BR LEVEL
012266  004737  013456                           JSR     PC,INTTY        ;GET RESPONSE
012272  022703  000024                           CMP     #24,R3          ;
012276  101014                                   BHI     50$             ;BR IF LESS THAN 4
012300  022703  000027                           CMP     #27,R3          ;
012304  103411                                   BLO     50$             ;BR IF GREATER THAN 7
012306  012704  000011                           MOV     #11,R4          ;R4 = NUMBER OF SHIFTS
012312  006303                                   ASL     R3              ;SHIFT R3 LEFT
012314  005304                                   DEC     R4              ;DEC SHIFT COUNT
012316  001375                                   BNE     .-4             ;BR IF NOT DONE
012320  042703  170777                           BIC     #170777,R3      ;BIC UNWANTED BITS
012324  050312                                   BIS     R3,(R2)         ;PUT BR LEVEL IN STATUS MAP
012326  000403                                   BR      8$              ;CONTINUE
012330  104401                          50$:     TYPE
012332  001312                                   $QUES                   ;RESPONSE IS OUT OF LIMITS
012334  000752                                   BR      10$             ;TRY AGAIN
012336                                  8$:
012336                                  9$:
012336  104401                          16$:     TYPE
012340  010461                                   MODU                    ;ASK WHICH LINE UNIT
012342  004737  013456                           JSR     PC,INTTY        ;GET REPLY
012346  022703  000021                           CMP     #21,R3          ;"1"
012352  001417                                   BEQ     30$
012354  022703  000022                           CMP     #22,R3          ;"2"
012360  001412                                   BEQ     31$
012362  022703  000116                           CMP     #116,R3         ;"N"
012366  001403                                   BEQ     32$
012370  104401                                   TYPE
012372  001312                                   $QUES                   ;IF NOT A 1,2 OR N TYPE "?"
012374  000760                                   BR      16$             ;TRY AGIAN
```

```
012376  052722  010000          32$:    BIS     #BIT12,(R2)+    ;SET BIT 12 IN STAT2 IF NO LU
012402  022222                          CMP     (R2)+,(R2)+     ;POP OVER STAT2 AND STAT3
012404  000445                          BR      33$
012406  052712  020000          31$:    BIS     #BIT13,(R2)     ;SET BIT 13 IN STAT2 IF M8202
012412  104401                  30$:    TYPE
012414  010671                          CONN                    ;ASK IF LOOP-BACK IS ON
012416  004737  013456                  JSR     PC,INTTY        ;GET REPLY
012422  022703  000131                  CMP     #131,R3         ;Y
012426  001406                          BEQ     17$
012430  022703  000116                  CMP     #116,R3         ;N
012434  001406                          BEQ     18$
012436  104401                          TYPE
012440  001312                          $QUES                   ;IF NOT Y OR N TYPE "?"
012442  000763                          BR      30$             ;TRY AGAIN
012444  052722  040000          17$:    BIS     #BIT14,(R2)+    ;TURNAROUND IS CONNECTED
012450  000402                          BR      19$
012452  042722  040000          18$:    BIC     #BIT14,(R2)+    ;NO TURNAROUND
012456                          19$:
012456  104415                          INPUT
012460  010573                          LINE
012462  000000                          0
012464  000377                          377
012466  001304                          $TMP3
012470     000                          .BYTE   0
012471     001                          .BYTE   1
012472  113722  001304                  MOVB    $TMP3,(R2)+     ;STORE SWITCH PAC IN MAP
012476  104415                          INPUT
012500  010631                          BM
012502  000000                          0
012504  000377                          377
012506  001304                          $TMP3
012510     000                          .BYTE   0
012511     001                          .BYTE   1
012512  113722  001304                  MOVB    $TMP3,(R2)+     ;STORE SWITCH PAC IN MAP
012516  005722                          TST     (R2)+           ;POP OVER STAT3
012520  005337  001302          33$:    DEC     $TMP2           ;DEC KMC COUNT
012524  001230                          BNE     12$             ;BR IF MORE TO DO
012526  000137  013064                  JMP     13$             ;CONTINUE
012532  012701  160000          7$:     MOV     #160000,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
012536  012737  013156  000004          MOV     #6$,@#4         ;SET FOR NON-EXISTANT DEVICE TIME OUT
012544  005011                  2$:     CLR     (R1)            ;CLEAR SEL0
012546  005711                          TST     (R1)            ;IF KMC11 KMCSR S/B 0
012550  001135                          BNE     3$              ;IF NO DEV ; TRAP TO 4. IF NO BIT 8 THEN NO KMC11
012552  005061  000006                  CLR     6(R1)           ;CLEAR SEL6
012556  005761  000006                  TST     6(R1)           ;IF KMC11 THEN KMRIC S/B =0!
012562  001130                          BNE     3$              ;BR IF NOT KMC11
012564  012711  002000                  MOV     #BIT10,(R1)     ;SET ROMO
012570  005061  000004                  CLR     4(R1)           ;CLEAR SEL4
012574  012761  125252  000006          MOV     #125252,6(R1)   ;WRITE THIS TO SEL6
012602  052711  020000                  BIS     #BIT13,(R1)     ;WRITE IT!
012606  022761  125252  000004          CMP     #125252,4(R1)   ;WAS IT WRITTEN?
012614  001113                          BNE     3$              ;IF NO IT IS NOT CRAM
                                ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A KMC11 CSR ADDRESS.
012616                          21$:
012616  010122                  22$:    MOV     R1,(R2)+        ;STORE CSR IN CORE TABLE.
012620  012711  001000          15$:    MOV     #BIT9,(R1)      ;CLEAR LINE UNIT LOOP
```

```
012624  005061  000004                  CLR     4(R1)               ;CLEAR PORT4
012630  012761  122113  000006          MOV     #122113,6(R1)       ;LOAD INSTRUCTION (CLR DTR)
012636  052711  000400                  BIS     #BIT8,(R1)          ;CLOCK INSTRUCTION
012642  012761  021264  000006          MOV     #021264,6(R1)       ;LOAD INSTRUCTION
012650  052711  000400                  BIS     #BIT8,(R1)          ;CLOCK INSTRUCTION
012654  122761  000377  000004          CMPB    #377,4(R1)          ;IS IT ALL ONES?
012662  001003                          BNE     .+10                ;BR IF NO
012664  052712  010000                  BIS     #BIT12,(R2)         ;IF YES, NO LINE UNIT, SET STATUS BIT
012670  000436                          BR      20$
012672  032761  000002  000004          BIT     #BIT1,4(R1)         ;IS SWITCH A ONE?
012700  001403                          BEQ     .+10                ;BR IF M8201
012702  052712  060000                  BIS     #BIT13!BIT14,(R2)   ;M8202 ASSUME CONNECTOR
012706  000427                          BR      20$                 ;CONNECTOR ON)
012710  032761  000010  000004          BIT     #BIT3,4(R1)         ;IS MRDY SET
012716  001023                          BNE     20$                 ;BR IF M8201 NO CONNECTOR (ON LINE)
012720  012761  000100  000004          MOV     #BIT6,4(R1)         ;LOAD PORT4
012726  012761  122113  000006          MOV     #122113,6(R1)       ;LOAD INSTRUCTION
012734  052711  000400                  BIS     #BIT8,(R1)          ;CLOCK INSTRUCTION(SET DTR)
012740  012761  021264  000006          MOV     #021264,6(R1)       ;LOAD INSTRUCTION
012746  052711  000400                  BIS     #BIT8,(R1)          ;CLOCK INSTRUCTION(READ MODEM REG)
012752  032761  000010  000004          BIT     #BIT3,4(R1)         ;IS MRDY SET NOW?
012760  001402                          BEQ     20$                 ;BR IF NO CONNECTOR
012762  052712  040000                  BIS     #BIT14,(R2)         ;SET STATUS BIT FOR CONNECTOR
012766  005722                  20$:    TST     (R2)+               ;POP POINTER
012770  012761  021324  000006          MOV     #021324,6(R1)       ;PUT INSTRUCTION IN PORT6
012776  012711  001400                  MOV     #BIT9!BIT8,(R1)     ;PORT4+LU 15
013002  156122  000004                  BISB    4(R1),(R2)+         ;STORE DDCMP LINE # IN TABLE
013006  012761  021344  000006          MOV     #021344,6(R1)       ;PORT6+INSTRUCTION
013014  012711  001400                  MOV     #BIT8!BIT9,(R1)     ;CLOCK INSTR.
013020  156122  000004                  BISB    4(R1),(R2)+         ;STORE BM873 ADD IN TABLE
013024  005722                          TST     (R2)+               ;POP OVER STAT3
013026  005011                          CLR     (R1)                ;CLEAR ROMI
013030  005237  001472                  INC     KMNUM               ;UPDATE DEVICE COUNTER
013034  022737  000020  001472          CMP     #20,KMNUM           ;ARE MAX. NO. OF DEV FOUND?
013042  001410                          BEQ     13$                 ;YES DON'T LOOK FOR ANY MORE.
013044  005011                  3$:     CLR     (R1)                ;CLEAR BIT 10
013046  005061  000006                  CLR     6(R1)               ;CLEAR SEL 6
013052  062701  000010          14$:    ADD     #10,R1              ;UPDATE CSR POINTER ADDRESS
013056  022701  164000                  CMP     #164000,R1
013062  001230                          BNE     2$                  ;BR IF MORE ADDRESS TO CHECK.
013064  005037  001470          13$:    CLR     KMACTV
013070  005737  001472                  TST     KMNUM               ;WERE ANY KMC11'S FOUND AT ALL?
013074  001423                          BEQ     5$                  ;ERROR AUTO SIZER FOUND NO KMC11'S IN THIS SYS.
013076  013701  001472                  MOV     KMNUM,R1
013102  010137  001476                  MOV     R1,SAVNUM           ;SAVE NUMBER OF DEVICES
013106  000241                  4$:     CLC
013110  006137  001470                  ROL     KMACTV              ;GENERATE ACTIVE REGISTER OF DEVICES.
013114  005237  001470                  INC     KMACTV              ;SET THE BIT
013120  005301                          DEC     R1
013122  001371                          BNE     4$                  ;BR IF MORE TO GENERATE
013124  012737  000006  000004          MOV     #6,@#4              ;RESTORE TRAP VECTOR
013132  013737  001470  001474          MOV     KMACTV,SAVACT       ;SAVE ACTIVE REGISTER
013140  000137  013172                  JMP     VECMAP              ;GO FIND THE VECTOR NOW.
013144  104401  007645          5$:     TYPE    ,MERR2              ;NOTIFY OPR THAT NO KMC11'S FOUND.
013150  005000                          CLR     R0                  ;MAKE DATA LIGHTS ZERO
013152  000000                          HALT                        ;STOP THE SHOW
```

```
013154  000776                            BR      .-2               ;DISABLE CONT. SW.
013156  012716  013052            6$:     MOV     #14$,(SP)         ;ENTERED BY NON-EXISTANT TIME-OUT.
013162  000002                            RTI                       ;RETURN TO MAINSTREAM

013164  000001                    WHICH:  1
013166     002     002                    .BYTE   2,2
013170  001306                            $TMP4

013172  032737  000001  001446    VECMAP: BIT     #SW00,STRTSW
013200  001114                            BNE     5$
013202  012737  000340  000022            MOV     #340,@#22         ;SET IOT TRAP PRIO TO 7
013210  012737  013364  000020            MOV     #4$,@#20          ;SET IOT TRAP VECTOR
013216  012702  002100                    MOV     #KM.MAP,R2        ;SET SOFTWARE POINTER
013222  012700  000300                    MOV     #300,R0           ;FLOATING VECTORS START HERE.
013226  012701  000302                    MOV     #302,R1           ;PC OF IOT INSTR.
013232  010120                    1$:     MOV     R1,(R0)+          ;START FILLING VECTOR AREA
013234  012721  000004                    MOV     #4,(R1)+          ;WITH .+2; IOT
013240  022021                            CMP     (R0)+,(R1)+       ;ADD 2 TO R0 +R1
013242  020127  001000                    CMP     R1,#1000
013246  101771                            BLOS    1$                ;BR IF MORE TO FILL
013250  013737  001470  001276            MOV     KMACTV,$TMP0      ;STORE TEMPORALLY
013256  006037  001276            2$:     ROR     $TMP0             ;BRING OUT A BIT
013262  103063                            BCC     5$                ;BR IF ALL DONE
013264  012704  000012                    MOV     #12,R4            ;R4 IS INDEX REGISTER
013270  016437  013442  177776            MOV     BRLVL(R4),PS      ;SET PS TO 7
013276  011201                            MOV     (R2),R1
013300  012761  000200  000004            MOV     #200,4(R1)
013306  012711  001000                    MOV     #BIT9,(R1)        ;SET ROMI
013312  012761  121111  000006            MOV     #121111,6(R1)     ;PUT INSTRUCTION IN PORT6
013320  012711  001400                    MOV     #BIT9!BIT8,(R1)   ;FORCE AN INTERRUPT
013324  105200                    7$:     INCB    R0                ;STALL
013326  001376                            BNE     .-2               ;FOR TIME TO INTERUPT
013330  162704  000002                    SUB     #2,R4             ;GET NEXT LOWEST PS LEVEL
013334  001404                            BEQ     6$                ;BR IF R4 = 0
013336  016437  013442  177776            MOV     BRLVL(R4),PS      ;MOVE NEXT LOWER LEVEL IN PS
013344  000767                            BR      7$                ;BR TO DELAY
013346  052762  005300  000002    6$:     BIS     #5300,2(R2)       ;NO INTERUPT ASSUME 300 AT LEVEL 5 AND FIX KMC11 LATER
013354  005011                    3$:     CLR     (R1)              ;CLEAR ROMI
013356  062702  000010                    ADD     #10,R2            ;POP SOFTWARE POINTER
013362  000735                            BR      2$                ;KEEP GOING
013364  051662  000002            4$:     BIS     (SP),2(R2)        ;GET VECTOR ADDRESS
013370  042762  000007  000002            BIC     #7,2(R2)          ;CLEAR JUNK
013376  016405  013444                    MOV     BRLVL+2(R4),R5    ;GET BR LEVEL OF KMC11
013402  006305                            ASL     R5                ;SHIFT LEVEL 4 PLACES
013404  006305                            ASL     R5                ;TO THE LEFT FOR THE
013406  006305                            ASL     R5                ;STATUS TABLE
013410  006305                            ASL     R5
013412  042705  170777                    BIC     #170777,R5        ;CLEAR UNWANTED BITS
013416  050562  000002                    BIS     R5,2(R2)          ;PUT BR LEVEL IN STATUS TABLE
013422  022626                            CMP     (SP)+,(SP)+       ;POP IOT JUNK OFF STACK
013424  012716  013354                    MOV     #3$,(SP)          ;SET FOR RETURN
013430  000002                            RTI
013432  012737  004134  000020    5$:     MOV     #$SCOPE,@#20      ; RESTORE SCOPE VECTOR
013440  000207                            RTS     PC                ;ALL DONE WITH "AUTO SIZING"

013442  000000                    BRLVL:  PR0     ;LEVEL 0
```

```
     013444  000000                             PR0       ;LEVEL 0
     013446  000200                             PR4       ;LEVEL 4
     013450  000240                             PR5       ;LEVEL 5
     013452  000300                             PR6       ;LEVEL 6
     013454  000340                             PR7       ;LEVEL 7


     013456  105777  165562          INTTY:     TSTB      @$TKS              ;WAIT FOR DONE
     013462  100375                             BPL       .-4
     013464  017703  165556                     MOV       @$TKB,R3           ;PUT CHAR IN R3
     013470  105777  165554                     TSTB      @$TPS              ;WAIT UNTIL PRINTER IS READY
     013474  100375                             BPL       .-4
     013476  010377  165550                     MOV       R3,@$TPB           ;ECHO CHAR
     013502  042703  000240                     BIC       #BIT7!BIT5,R3      ;MASK OFF LOWER CASE
     013506  000207                             RTS       PC                 ;RETURN

     013510                          APT.SIZE:
     013510  000005                             RESET
     013512  010046                             MOV       R0,-(SP)           ;;PUSH R0 ON STACK
     013514  010146                             MOV       R1,-(SP)           ;;PUSH R1 ON STACK
     013516  010246                             MOV       R2,-(SP)           ;;PUSH R2 ON STACK
     013520  010346                             MOV       R3,-(SP)           ;;PUSH R3 ON STACK
     013522  005037  013724                     CLR       VECTR              ; CLEAR THE LOCAL VARIABLE
     013526  005037  013730                     CLR       PRIRTY             ; CLEAN UP LOCAL VARIABLE
     013532  013700  001376                     MOV       $CDW1,R0           ; GET THE DEVICE COUNT
     013536  010037  001476                     MOV       R0,SAVNUM          ; SAVE THE NO. OF DEVICES
     013542  012701  001346                     MOV       #$MAMS1,R1         ; GET EXTRA INFO. BITS POINTER
     013546  013737  001372  013726             MOV       $BASE,BASE         ; GET BASE CSR ADDRESS
     013554  113737  001366  013724             MOVB      $VECT1,VECTR       ; GET THE VECTOR
     013562  113737  001367  013730             MOVB      $VECT1+1,PRIRTY    ; GET THE PRIORITY
     013570  013737  001474  001470             MOV       $DEVM,KMACTV       ; SAVE THE KMC'S SELECTED ACTIVE
     013576  013737  001470  001474             MOV       KMACTV,SAVACT      ; SAVE THE ACTIVE REGISTER
     013604  012702  001402                     MOV       #$DDWD,R2          ; GET ADDRESS OF FIRST DEVICE DESCRIPTOR WORD
     013610  012703  002100                     MOV       #KM.MAP,R3         ; GET POINTER TO DEVICE MAP
     013614  005023                  3$:        CLR       (R3)+              ; CLEAR DEVICE MAP
     013616  022703  002300                     CMP       #KM.END,R3         ; IS WHOLE DEV.MAP CLEARED?
     013622  003374                             BGT       3$                 ; NO, THEN GO ON.
     013624  012703  002100                     MOV       #KM.MAP,R3         ; RESTORE DEV.MAP POINTER.
     013630  013723  013726          1$:        MOV       BASE,(R3)+         ; LOAD CSR ADDRESS
     013634  112163  000001                     MOVB      (R1)+,1(R3)        ; GET EXTRA INFO. BITS
     013640  006213                             ASR       (R3)               ; SET IT IN RIGHT POSITION.
     013642  006213                             ASR       (R3)               ; SET IT IN RIGHT POSITION.
     013644  053713  013730                     BIS       PRIRTY,(R3)        ; GET PRIORITY IN STAT1
     013650  006313                             ASL       (R3)               ; SET THEM IN RIGHT POSITION
     013652  006313                             ASL       (R3)               ;   -    -    -    -    -
     013654  006313                             ASL       (R3)               ;   -    -    -    -    -
     013656  006313                             ASL       (R3)               ;   -    -    -    -    -
     013660  053723  013724                     BIS       VECTR,(R3)+        ; GET THE VECTOR IN STAT1.
     013664  012223                             MOV       (R2)+,(R3)+        ; GET THE STAT2 FROM DDWXX
     013666  005723                             TST       (R3)+              ; SKIP OVER STAT3
     013670  005300                             DEC       R0                 ; COUNT BY 1
     013672  001407                             BEQ       2$                 ; ALL DONE?
     013674  062737  000010  013726             ADD       #10,BASE           ; INCREMENT BASE CSR ADDRESS BY 10
     013702  062737  000010  013724             ADD       #10,VECTR          ; INCREMENT VECTOR ADDRESS BY 10
     013710  000747                             BR        1$                 ; SET THE NEXT MAP ENTRY
     013712                          2$:
```

```
013712  012603                                     MOV     (SP)+,R3        ;;POP STACK INTO R3
013714  012602                                     MOV     (SP)+,R2        ;;POP STACK INTO R2
013716  012601                                     MOV     (SP)+,R1        ;;POP STACK INTO R1
013720  012600                                     MOV     (SP)+,R0        ;;POP STACK INTO R0
013722  000207                                     RTS     PC              ; RETURN
013724  000000                           VECTR:    .WORD   0
013726  000000                           BASE:     .WORD   0
013730  000000                           PRIRTY:   .WORD   0



                                                   ;************************** 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
                                                   ;_____________

                                         ;;*****************************************************************
013732  000004                           TST1:     SCOPE
013734  012737  000001  001202                     MOV     #1,$TSTNM               ; LOAD THE NO. OF THIS TEST
013742  012737  014006  001442                     MOV     #TST2,NEXT              ; POINT TO THE START OF NEXT TEST.
                                                                           ;R1 CONTAINS BASE KMC11 ADDRESS
013750  005077  166112                             CLR     @KMCSR          ;CLEAR SEL0
013754  012702  000011                             MOV     #11,R2          ;SAVE R2 FOR TYPEOUT
013760  104412                                     ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
013762  021224                                     021004!<20*11>          ;PORT4+LINE UNIT REG 11
013764  016104  000004                             MOV     4(R1),R4        ;PUT "FOUND" IN R4
013770  042704  000054                             BIC     #54,R4          ;CLEAR UNKNOWN BITS
013774  012705  000020                             MOV     #20,R5          ;PUT "EXPECTED" IN R5
014000  120504                                     CMPB    R5,R4           ;IS OUT READY SET?
014002  001401                                     BEQ     1$              ;BR IF YES
014004  104002                                     ERROR   2               ;ERROR IN LU 11
014006                                   1$:



                                                   ;************************** 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
                                                   ;_____________

                                         ;;*****************************************************************
014006  000004                           TST2:     SCOPE
014010  012737  000002  001202                     MOV     #2,$TSTNM               ; LOAD THE NO. OF THIS TEST
014016  012737  014054  001442                     MOV     #TST3,NEXT              ; POINT TO THE START OF NEXT TEST.
                                                                           ;R1 CONTAINS BASE KMC11 ADDRESS
014024  012702  000012                             MOV     #12,R2          ;SAVE R2 FOR TYPEOUT
014030  104412                                     ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014032  021244                                     021004!<20*12>          ;PORT4+LINE UNIT REG 12
014034  016104  000004                             MOV     4(R1),R4        ;PUT "FOUND" IN R4
014040  042704  000017                             BIC     #17,R4          ;CLEAR UNKNOWN BITS
014044  005005                                     CLR     R5              ;PUT "EXPECTED" IN R5
```

```
014046  120504                          CMPB    R5,R4                   ;ARE ALL BITS CLEARED?
014050  001401                          BEQ     1$                      ;BR IF YES
014052  104002                          ERROR   2                       ;ERROR IN LU 12
014054                          1$:


                                        ;************************* 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
                                        ;---------------

                                ;;*********************************************************
014054  000004                  TST3:   SCOPE
014056  012737  000003  001202          MOV     #3,$TSTNM                       ; LOAD THE NO. OF THIS TEST
014064  012737  014126  001442          MOV     #TST4,NEXT                      ; POINT TO THE START OF NEXT TEST.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
014072  104410                          MSTCLR                          ;MASTER CLEAR KMC11
014074  012702  000013                  MOV     #13,R2                  ;SAVE R2 FOR TYPEOUT
014100  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014102  021264                          021004!<20*13>                  ;PORT4←LINE UNIT REG 13
014104  016104  000004                  MOV     4(R1),R4                ;PUT "FOUND" IN R4
014110  042704  000213                  BIC     #213,R4                 ;CLEAR UNKNOWN BITS
014114  012705  000100                  MOV     #100,R5                 ;PUT "EXPECTED" IN R5
014120  120504                          CMPB    R5,R4                   ;ARE RING, DTR,  AND MODEM READY SET?
014122  001401                          BEQ     1$                      ;BR IF YES
014124  104002                          ERROR   2                       ;ERROR IN LU 13
014126                          1$:


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

                                        ;  TEST 4
                                        ;---------------

                                ;;*********************************************************
014126  000004                  TST4:   SCOPE
014130  012737  000004  001202          MOV     #4,$TSTNM                       ; LOAD THE NO. OF THIS TEST
014136  012737  014220  001442          MOV     #TST5,NEXT                      ; POINT TO THE START OF NEXT TEST.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
014144  104410                          MSTCLR                          ;MASTER CLEAR KMC11
014146  012702  000017                  MOV     #17,R2                  ;SAVE R2 FOR TYPEOUT
014152  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014154  021364                          021004!<20*17>                  ;PORT4←LINE UNIT REG 17
014156  016104  000004                  MOV     4(R1),R4                ;PUT "FOUND" IN R4
014162  042704  000206                  BIC     #206,R4                 ;CLEAR UNKNOWN BITS
014166  012705  000051                  MOV     #51,R5                  ;PUT "EXPECTED" IN R5
014172  032737  020000  002050          BIT     #BIT13,STAT1            ;IS LU AN M8202 OR M8201?
014200  001404                          BEQ     .+12                    ;BR IF M8201
014202  042704  000040                  BIC     #40,R4                  ;MASK OFF SI BIT IF M8202
014206  042705  000040                  BIC     #BIT5,R5                ;SI BIT IS UNKNOWN ON AN M8202
```

```
014212  120504                                   CMPB    R5,R4                    ;ARE SI AND ICIR SET?
014214  001401                                   BEQ     1$                       ;BR IF YES
014216  104002                                   ERROR   2                        ;ERROR IN LU 17
014220                                   1$:


                                                 ;************************* 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
                                                 ;---------------
                                         ;;*****************************************************
014220  000004                           TST5:   SCOPE
014222  012737  000005  001202                   MOV     #5,$TSTNM                        ; LOAD THE NO. OF THIS TEST
014230  012737  014360  001442                   MOV     #TST6,NEXT                       ; POINT TO THE START OF NEXT TEST.
014236  012737  014252  001444                   MOV     #1$,LOCK                         ; ADDRESS FOR LOCK ON DATA.
                                                                          ;R1 CONTAINS BASE KMC11 ADDRESS
014244  104410                                   MSTCLR                   ;MASTER CLEAR KMC11
014246  012702  000012                           MOV     #12,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
014252  012761  000040  000004           1$:     MOV     #40,4(R1)        ;LOAD PORT4
014260  104412                                   ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014262  122112                                   122112                   ;SET BITS IN LU-12
014264  104412                                   ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014266  021245                                   021245                   ;READ LU-12
014270  012705  000040                           MOV     #40,R5           ;PUT "EXPECTED" IN R5
014274  116104  000005                           MOVB    5(R1),R4         ;PUT "FOUND" IN R4
014300  042704  000337                           BIC     #337,R4          ;CLEAR UNWANTED BITS
014304  120504                                   CMPB    R5,R4            ;IS BITS SET?
014306  001401                                   BEQ     2$               ;BR IF YES
014310  104003                                   ERROR   3                ;ERROR, BIT 5 IS NOT SET
014312  104405                           2$:     SCOP1                    ;SCOPE SUBTEST (SW09=1)
014314  012737  014322  001444                   MOV     #3$,LOCK         ;NEW SCOP1
014322  005061  000004                   3$:     CLR     4(R1)            ;LOAD PORT4
014326  104412                                   ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014330  122112                                   122112                   ;CLEAR BIT 5 IN LU-12
014332  104412                                   ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014334  021245                                   021245                   ;READ LU-12
014336  005005                                   CLR     R5               ;PUT "EXPECTED" IN R5
014340  116104  000005                           MOVB    5(R1),R4         ;PUT "FOUND" IN R4
014344  042704  000337                           BIC     #337,R4          ;CLEAR UNWANTED BITS
014350  120504                                   CMPB    R5,R4            ;IS BITS CLEAR?
014352  001401                                   BEQ     4$               ;BR IF YES
014354  104003                                   ERROR   3                ;ERROR, BITS IS NOT CLEAR
014356  104405                           4$:     SCOP1                    ;SCOPE SUBTEST (SW09=1)


                                                 ;************************* 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
```

F06

```
                                                         ;---------------
                                                 ;;****************************************************************
014360  000004                          TST6:    SCOPE
014362  012737  000006  001202                   MOV     #6,$TSTNM              ; LOAD THE NO. OF THIS TEST
014370  012737  014520  001442                   MOV     #TST7,NEXT             ; POINT TO THE START OF NEXT TEST.
014376  012737  014412  001444                   MOV     #1$,LOCK               ; ADDRESS FOR LOCK ON DATA.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
014404  104410                                   MSTCLR                 ;MASTER CLEAR KMC11
014406  012702  000017                           MOV     #17,R2         ;SAVE REGISTER ADDRESS FOR TYPEOUT
014412  012761  000001  000004          1$:      MOV     #1,4(R1)       ;LOAD PORT4
014420  104412                                   ROMCLK                 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014422  122117                                   122117                 ;SET BIT1 IN LU-17
014424  104412                                   ROMCLK                 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014426  021365                                   021365                 ;READ LU-17
014430  012705  000001                           MOV     #1,R5          ;PUT "EXPECTED" IN R5
014434  116104  000005                           MOVB    5(R1),R4       ;PUT "FOUND" IN R4
014440  042704  000376                           BIC     #376,R4        ;CLEAR UNWANTED BITS
014444  120504                                   CMPB    R5,R4          ;IS BIT1 SET?
014446  001401                                   BEQ     2$             ;BR IF YES
014450  104003                                   ERROR   3              ;ERROR, BIT 1 IS NOT SET
014452  104405                          2$:      SCOP1                  ;SCOPE SUBTEST (SW09=1)
014454  012737  014462  001444                   MOV     #3$,LOCK       ;NEW SCOP1
014462  005061  000004                  3$:      CLR     4(R1)          ;LOAD PORT4
014466  104412                                   ROMCLK                 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014470  122117                                   122117                 ;CLEAR BIT 1 IN LU-17
014472  104412                                   ROMCLK                 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014474  021365                                   021365                 ;READ LU-17
014476  005005                                   CLR     R5             ;PUT "EXPECTED" IN R5
014500  116104  000005                           MOVB    5(R1),R4       ;PUT "FOUND" IN R4
014504  042704  000376                           BIC     #376,R4        ;CLEAR UNWANTED BITS
014510  120504                                   CMPB    R5,R4          ;IS BIT1 CLEAR?
014512  001401                                   BEQ     4$             ;BR IF YES
014514  104003                                   ERROR   3              ;ERROR, BIT1 IS NOT CLEAR
014516  104405                          4$:      SCOP1                  ;SCOPE SUBTEST (SW09=1)



                                                 ;*************************** 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
                                                 ;--------------
                                                 ;;****************************************************************
014520  000004                          TST7:    SCOPE
014522  012737  000007  001202                   MOV     #7,$TSTNM              ; LOAD THE NO. OF THIS TEST
014530  012737  014730  001442                   MOV     #TST10,NEXT            ; POINT TO THE START OF NEXT TEST.
014536  012737  014556  001444                   MOV     #64$,LOCK              ; ADDRESS FOR LOCK ON DATA.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
014544  104410                                   MSTCLR                 ;MASTER CLEAR KMC11
014546  012702  000013                           MOV     #13,R2         ;SAVE REGISTER ADDRESS FOR TYPEOUT
014552  012700  000001                           MOV     #1,R0          ;START WITH BIT 0
014556                                  64$:
014556  010061  000004                           MOV     R0,4(R1)       ;PUT PATTERN INTO PORT4
014562  042761  000257  000004                   BIC     #257,4(R1)     ;CLEAR UNWANTED BITS
```

```
014570  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014572  122113                                  122100!13                       ;MOV DATA TO IBUS REGISTER 13
014574  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014576  021265                                  21005!<13*20>                   ;READ FROM IBUS REGISTER 13
014600  010005                                  MOV     R0,R5                   ;PUT EXPECTED IN R5
014602  042705  000257                          BIC     #257,R5                 ;CLEAR UNWANTED BITS
014606  116104  000005                          MOVB    5(R1),R4                ;PUT "FOUND" INTO R4
014612  042704  000257                          BIC     #257,R4                 ;CLEAR UNWANTED BITS
014616  120504                                  CMPB    R5,R4                   ;DATA CORRECT?
014620  001401                                  BEQ     65$                     ;BR IF YES
014622  104003                                  ERROR   3                       ;ERROR
014624  104405                          65$:    SCOP1                           ;SW09=1?
014626  000241                                  CLC                             ;CLEAR CARRY
014630  106100                                  ROLB    R0                      ;SHIFT BIT IN R0
014632  001351                                  BNE     64$                     ;IF R0=0 THEN DONE
014634  012737  014650  001444                  MOV     #67$,LOCK               ;NEW SCOP1
014642  012700  000001                          MOV     #1,R0                   ;START WITH BIT 0
014646  005100                          69$:    COM     R0                      ;CHANGE TO FLOATING ZERO
014650                                  67$:
014650  010061  000004                          MOV     R0,4(R1)                ;PUT PATTERN INTO PORT4
014654  042761  000257  000004                  BIC     #257,4(R1)              ;CLEAR UNWANTED BITS
014662  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014664  122113                                  122100!13                       ;MOV DATA TO IBUS REGISTER 13
014666  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014670  021265                                  21005!<13*20>                   ;READ FROM IBUS REGISTER 13
014672  010005                                  MOV     R0,R5                   ;PUT EXPECTED IN R5
014674  042705  000257                          BIC     #257,R5                 ;CLEAR UNWANTED BITS
014700  116104  000005                          MOVB    5(R1),R4                ;PUT "FOUND" INTO R4
014704  042704  000257                          BIC     #257,R4                 ;CLEAR UNWANTED BITS
014710  120504                                  CMPB    R5,R4                   ;DATA CORRECT?
014712  001401                                  BEQ     68$                     ;BR IF YES
014714  104003                                  ERROR   3                       ;ERROR
014716  104405                          68$:    SCOP1                           ;SW09=1?
014720  005100                                  COM     R0                      ;CHANGE TO FLOATING 1
014722  000241                                  CLC                             ;CLEAR CARRY
014724  106100                                  ROLB    R0                      ;SHIFT BIT IN R0
014726  001347                                  BNE     69$                     ;IF R0=0 THEN DONE


                                                ;**************************** 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
                                                ;--------------

                                        ;;*******************************************************
014730  000004                          TST10:  SCOPE
014732  012737  000010  001202                  MOV     #10,$TSTNM                     ; LOAD THE NO. OF THIS TEST
014740  012737  015104  001442                  MOV     #TST11,NEXT                    ; POINT TO THE START OF NEXT TEST.
014746  012737  014766  001444                  MOV     #64$,LOCK                      ; ADDRESS FOR LOCK ON DATA.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
014754  104410                                  MSTCLR                          ;MASTER CLEAR KMC11
014756  012702  000014                          MOV     #14,R2                  ;SAVE REGISTER ADDRESS FOR TYPEOUT
014762  012700  000001                          MOV     #1,R0                   ;START WITH BIT 0
```

```
014766                                 64$:
014766  010061  000004                         MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
014772  104412                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
014774  122114                                 122100!14               ;MOV DATA TO IBUS REGISTER 14
014776  104412                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015000  021305                                 21005!<14*20>           ;READ FROM IBUS REGISTER 14
015002  010005                                 MOV     R0,R5           ;PUT EXPECTED IN R5
015004  116104  000005                         MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
015010  120504                                 CMPB    R5,R4           ;DATA CORRECT?
015012  001401                                 BEQ     65$             ;BR IF YES
015014  104003                                 ERROR   3               ;ERROR
015016  104405                         65$:    SCOP1                   ;SW09=1?
015020  000241                                 CLC                     ;CLEAR CARRY
015022  106100                                 ROLB    R0              ;SHIFT BIT IN R0
015024  001360                                 BNE     64$             ;IF R0=0 THEN DONE
015026  012737  015042  001444                 MOV     #67$,LOCK       ;NEW SCOP1
015034  012700  000001                         MOV     #1,R0           ;START WITH BIT 0
015040  005100                         69$:    COM     R0              ;CHANGE TO FLOATING ZERO
015042                                 67$:
015042  010061  000004                         MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
015046  104412                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015050  122114                                 122100!14               ;MOV DATA TO IBUS REGISTER 14
015052  104412                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015054  021305                                 21005!<14*20>           ;READ FROM IBUS REGISTER 14
015056  010005                                 MOV     R0,R5           ;PUT EXPECTED IN R5
015060  116104  000005                         MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
015064  120504                                 CMPB    R5,R4           ;DATA CORRECT?
015066  001401                                 BEQ     68$             ;BR IF YES
015070  104003                                 ERROR   3               ;ERROR
015072  104405                         68$:    SCOP1                   ;SW09=1?
015074  005100                                 COM     R0              ;CHANGE TO FLOATING 1
015076  000241                                 CLC                     ;CLEAR CARRY
015100  106100                                 ROLB    R0              ;SHIFT BIT IN R0
015102  001356                                 BNE     69$             ;IF R0=0 THEN DONE



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


                                               ;  TEST 11
                                               ;--------------

                                       ;;*********************************************************
015104  000004                         TST11:  SCOPE
015106  012737  000011  001202                 MOV     #11,$TSTNM              ; LOAD THE NO. OF THIS TEST
015114  012737  015146  001442                 MOV     #TST12,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                       ;R1 CONTAINS BASE KMC11 ADDRESS
015122  104410                                 MSTCLR                  ;MASTER CLEAR KMC11
015124  104412                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015126  021324                                 021324                  ;PORT4+LU15
015130  016104  000004                         MOV     4(R1),R4        ;PUT "FOUND" IN R4
015134  113705  002052                         MOVB    STAT2,R5        ;PUT "EXPECTED" IN R5
015140  120504                                 CMPB    R5,R4           ;SW OK?
015142  001401                                 BEQ     1$              ;BR IF YES
```

```
015144  104031                          ERROR   31              ;ERROR, SWITCH PAC READ ERROR
015146                          1$:



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

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

                                ;;**************************************************************
015146  000004                  TST12:  SCOPE
015150  012737  000012  001202          MOV     #12,$TSTNM              ; LOAD THE NO. OF THIS TEST
015156  012737  015210  001442          MOV     #TST13,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
015164  104410                          MSTCLR                  ;MASTER CLEAR KMC11
015166  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015170  021344                          021344                  ;PORT4+LU16
015172  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
015176  113705  002053                  MOVB    STAT2+1,R5      ;PUT "EXPECTED" IN R5
015202  120504                          CMPB    R5,R4           ;SW OK?
015204  001401                          BEQ     1$              ;BR IF YES
015206  104031                          ERROR   31              ;ERROR, SWITCH PAC READ ERROR
015210                          1$:


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

                                        ;  TEST 13
                                        ;--------------

                                ;;**************************************************************
015210  000004                  TST13:  SCOPE
015212  012737  000013  001202          MOV     #13,$TSTNM              ; LOAD THE NO. OF THIS TEST
015220  012737  015310  001442          MOV     #TST14,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
015226  104410                          MSTCLR                  ;MASTER CLEAR KMC11
015230  005037  011106                  CLR     TEMP            ;PREPARE FOR DELAY
015234                          1$:
015234  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015236  021364                          021364                  ;PORT4+LU-17
015240  032761  000002  000004          BIT     #2,4(R1)        ;IS CLOCK BIT SET?
015246  001004                          BNE     2$              ;BR IF YES
015250  005237  011106                  INC     TEMP            ;DELAY
015254  001367                          BNE     1$      ;DELAY FINISHED?
015256  104004                          ERROR   4               ;ERROR BIT IS STUCK CLEAR
015260  005037  011106          2$:     CLR     TEMP            ;PREPARE FOR DELAY
015264                          3$:
015264  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015266  021364                          021364                  ;PORT4+LU-17
015270  032761  000002  000004          BIT     #2,4(R1)        ;IS CLOCK BIT CLEAR?
```

```
015276  001404                          BEQ     4$                      ;BR IF YES
015300  005237  011106                  INC     TEMP                    ;DELAY
015304  001367                          BNE     3$                      ;BR IF DELAY NOT DONE
015306  104004                          ERROR   4                       ;ERROR BIT IS STUCK SET
015310                          4$:



                                        ;************************ 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
                                        ;--------------

                                ;;*********************************************************
015310  000004                  TST14:  SCOPE
015312  012737  000014  001202          MOV     #14,$TSTNM              ; LOAD THE NO. OF THIS TEST
015320  012737  015410  001442          MOV     #TST15,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
015326  104410                          MSTCLR                  ;MASTER CLEAR KMC11
015330  012711  004000                  MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
015334  012761  000001  000004          MOV     #1,4(R1)        ;LOAD PORT4 WITH BIT0
015342  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015344  122111                          122111                  ;SET SOM
015346  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015350  122110                          122110                  ;LOAD OUT DATA SILO
015352  104414  000002                  TIMER,  2               ;WAIT FOR OCOR
015356  012702  000017                  MOV     #17,R2          ;SAVE ADDRESS FOR TYPEOUT
015362  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015364  021364                          021364                  ;PORT4←LU 17
015366  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
015372  042704  000357                  BIC     #357,R4         ;CLEAR UNWANTED BITS
015376  012705  000020                  MOV     #20,R5          ;PUT "EXPECTED" IN R5
015402  120504                          CMPB    R5,R4           ;IS OCOR SET?
015404  001401                          BEQ     1$              ;BR IF YES
015406  104005                          ERROR   5
015410                          1$:


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

                                        ;  TEST 15
                                        ;--------------

                                ;;*********************************************************
015410  000004                  TST15:  SCOPE
015412  012737  000015  001202          MOV     #15,$TSTNM              ; LOAD THE NO. OF THIS TEST
015420  012737  015546  001442          MOV     #TST16,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
015426  104410                          MSTCLR                  ;MASTER CLEAR KMC11
```

```
015430  012711  004000                    MOV     #BIT11,(R1)        ;SET LINE UNIT LOOP
015434  012761  000001  000004            MOV     #1,4(R1)           ;LOAD PORT4 WITH BIT0
015442  104412                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015444  122111                            122111                     ;SET SOM
015446  104412                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015450  122110                            122110                     ;LOAD OUT DATA SILO
015452  004737  030006                    JSR     PC,OCOR            ;WAIT FOR OCOR
015456  104413  000002                    DATACLK,        2          ;CLOCK DATA FOUR TIMES
015462  012702  000011                    MOV     #11,R2             ;SAVE ADDRESS FOR TYPEOUT
015466  104412                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015470  021224                            021224                     ;PORT4←LU 11
015472  016104  000004                    MOV     4(R1),R4           ;PUT "FOUND" IN R4
015476  042704  000257                    BIC     #257,R4            ;CLEAR UNWANTED BITS
015502  012705  000120                    MOV     #120,R5            ;PUT "EXPECTED" IN R5
015506  120504                            CMPB    R5,R4              ;IS ACTIVE SET?
015510  001401                            BEQ     1$                 ;BR IF YES
015512  104005                            ERROR   5
015514                            1$:
015514  012702  000013                    MOV     #13,R2             ;SAVE ADDRESS FOR TYPEOUT
015520  104412                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015522  021264                            021264                     ;PORT4←LU 13
015524  016104  000004                    MOV     4(R1),R4           ;PUT EXPECTED IN R4
015530  042704  000337                    BIC     #337,R4            ;CLEAR UNWANTED BITS
015534  012705  000040                    MOV     #BIT5,R5           ;PUT "EXPECTED" IN R5, RTS SHOULD BE SET
015540  120504                            CMPB    R5,R4              ;IS RTS OK?
015542  001401                            BEQ     2$                 ;BR IF YES
015544  104005                            ERROR   5                  ;RTS ERROR
015546                            2$:


                                          ;************************** 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
                                          ;-------------
                                  ;:*****************************************************************
015546  000004                    TST16:  SCOPE
015550  012737  000016  001202            MOV     #16,$TSTNM                ; LOAD THE NO. OF THIS TEST
015556  012737  015744  001442            MOV     #TST17,NEXT               ; POINT TO THE START OF NEXT TEST.
                                                                     ;R1 CONTAINS BASE KMC11 ADDRESS
015564  104410                            MSTCLR                     ;MASTER CLEAR KMC11
015566  012711  004000                    MOV     #BIT11,(R1)        ;SET LINE UNIT LOOP
015572  012761  000001  000004            MOV     #1,4(R1)           ;LOAD PORT4 WITH BIT0
015600  104412                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015602  122111                            122111                     ;SET SOM
015604  104412                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015606  122110                            122110                     ;LOAD OUT DATA SILO
015610  004737  030006                    JSR     PC,OCOR            ;WAIT FOR OCOR
015614  104413  000002                    DATACLK,        2          ;CLOCK DATA FOUR TIMES
015620  012761  000200  000004            MOV     #BIT7,4(R1)        ;SET BIT7 IN PORT4
015626  104412                            ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015630  122111                            122111                     ;SET OUT CLEAR
```

```
015632  104413  000001                       DATACLK,       1          ;GIVE A TICK TO CLEAR RTS
015636  012702  000017                       MOV     #17,R2            ;SAVE ADDRESS FOR TYPEOUT
015642  104412                               ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015644  021364                               021364                    ;PORT4+LU 17
015646  016104  000004                       MOV     4(R1),R4          ;PUT "FOUND" IN R4
015652  042704  000357                       BIC     #357,R4           ;CLEAR UNWANTED BITS
015656  005005                               CLR     R5                ;PUT "EXPECTED" IN R5
015660  120504                               CMPB    R5,R4             ;IS OCOR CLEARED?
015662  001401                               BEQ     1$                ;BR IF YES
015664  104005                               ERROR   5
015666                               1$:
015666  012702  000013                       MOV     #13,R2            ;SAVE ADDRESS FOR TYPEOUT
015672  104412                               ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015674  021264                               021264                    ;PORT4+LU 13
015676  016104  000004                       MOV     4(R1),R4          ;PUT EXPECTED IN R4
015702  042704  000337                       BIC     #337,R4           ;CLEAR UNWANTED BITS
015706  005005                               CLR     R5                ;PUT "EXPECTED" IN R5, RTS SHOULD BE CLEARED
015710  120504                               CMPB    R5,R4             ;IS RTS OK?
015712  001401                               BEQ     2$                ;BR IF YES
015714  104005                               ERROR   5                 ;RTS ERROR
015716                               2$:
015716  012702  000011                       MOV     #11,R2            ;SAVE ADDRESS FOR TYPEOUT
015722  104412                               ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015724  021224                               021224                    ;PORT4+LU11
015726  016104  000004                       MOV     4(R1),R4          ;PUT "FOUND" IN R4
015732  012705  000020                       MOV     #BIT4,R5          ;ONLY OUT READY SHOULD BE SET
015736  120504                               CMPB    R5,R4             ;IS ACTIVE CLEAR?
015740  001401                               BEQ     3$                ;BR IF YES
015742  104005                               ERROR   5                 ;ERROR ACTIVE NOT CLEARED
015744                               3$:


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

                                             ;  TEST 17
                                             ;--------------
                                     ;;***********************************************************************
015744  000004                       TST17:  SCOPE
015746  012737  000017  001202               MOV     #17,$TSTNM                ; LOAD THE NO. OF THIS TEST
015754  012737  016126  001442               MOV     #TST20,NEXT               ; POINT TO THE START OF NEXT TEST.
                                                                       ;R1 CONTAINS BASE KMC11 ADDRESS
015762  104410                               MSTCLR                    ;MASTER CLEAR KMC11
015764  012711  004000                       MOV     #BIT11,(R1)       ;SET LINE UNIT LOOP
015770  004737  030140                       JSR     PC,OUTRDY         ;WAIT FOR OUT-READY
015774  012761  000001  000004               MOV     #1,4(R1)          ;SET BIT0 IN PORT4
016002  104412                               ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016004  122111                               122111                    ;SET SOM!
016006  012705  000000                       MOV     #0,R5    ;LOAD CHARACTER IN R5 FOR TYPEOUT
016012  004737  030140                       JSR     PC,OUTRDY         ;WAIT FOR OUT-READY
016016  010561  000004                       MOV     R5,4(R1)          ;LOAD PORT4 WITH CHARACTER
```

```
016022  104412                                   ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016024  122110                                   122110                       ;LOAD OUT DATA
016026  004737  030006                           JSR     PC,OCOR              ;WAIT FOR OCOR TO SET
016032  005003                                   CLR     R3                   ;CLEAR BIT COUNTER
016034  010502                                   MOV     R5,R2                ;LOAD CHARACTER IN R2
016036  104413  000002                           DATACLK,        2            ;2 TICKS TO SET UP TRANSMITTER
016042  104413  000001                   1$:     DATACLK,          1          ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
016046  106002                                   RORB    R2                   ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
016050  103005                                   BCC     2$                   ;BR IF CARRY CLEAR
016052  004737  027754                           JSR     PC,GETSI             ;GET THE WINDOW
016056  103406                                   BCS     3$                   ;BR IF BIT IS A MARK
016060  104006                                   ERROR   6                    ;ERROR BIT WAS A SPACE
016062  000404                                   BR      3$                   ;CONTINE WITH TEST
016064  004737  027754                   2$:     JSR     PC,GETSI             ;GET THE WINDOW
016070  103001                                   BCC     3$                   ;BR IF BIT IS A SPACE
016072  104006                                   ERROR   6                    ;ERROR BIT WAS A MARK
016074                                   3$:
016074  005203                                   INC     R3                   ;NEXT BIT
016076  022703  000010                           CMP     #10,R3               ;DONE YET?
016102  001357                                   BNE     1$                   ;BR IF NO
016104  104413  000014                           DATACLK,          14         ;CLOCK TRANSMITTER 14 MORE TICKS
016110  104412                                   ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016112  021264                                   021264                       ;PORT4+LU-13
016114  032761  000040  000004                   BIT     #BIT5,4(R1)          ;RTS SHOULD BE CLEAR NOW
016122  001401                                   BEQ     4$                   ;BR IF YES
016124  104034                                   ERROR   34                   ;ERROR, RTS NOT CLEAR
016126                                   4$:



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

                                                 ;  TEST 20
                                                 ;--------------

                                         ;;*********************************************************
016126  000004                           TST20:  SCOPE
016130  012737  000020  001202                   MOV     #20,$TSTNM                ; LOAD THE NO. OF THIS TEST
016136  012737  016310  001442                   MOV     #TST21,NEXT               ; POINT TO THE START OF NEXT TEST.
                                                                              ;R1 CONTAINS BASE KMC11 ADDRESS
016144  104410                                   MSTCLR                       ;MASTER CLEAR KMC11
016146  012711  004000                           MOV     #BIT11,(R1)          ;SET LINE UNIT LOOP
016152  004737  030140                           JSR     PC,OUTRDY            ;WAIT FOR OUT-READY
016156  012761  000001  000004                   MOV     #1,4(R1)             ;SET BIT0 IN PORT4
016164  104412                                   ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016166  122111                                   122111                       ;SET SOM!
016170  012705  000125                           MOV     #125,R5 ;LOAD CHARACTER IN R5 FOR TYPEOUT
016174  004737  030140                           JSR     PC,OUTRDY            ;WAIT FOR OUT-READY
016200  010561  000004                           MOV     R5,4(R1)             ;LOAD PORT4 WITH CHARACTER
016204  104412                                   ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016206  122110                                   122110                       ;LOAD OUT DATA
016210  004737  030006                           JSR     PC,OCOR              ;WAIT FOR OCOR TO SET
```

```
016214  005003                                  CLR     R3                      ;CLEAR BIT COUNTER
016216  010502                                  MOV     R5,R2                   ;LOAD CHARACTER IN R2
016220  104413  000002                          DATACLK,        2               ;2 TICKS TO SET UP TRANSMITTER
016224  104413  000001                  1$:     DATACLK,        1               ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
016230  106002                                  RORB    R2                      ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
016232  103005                                  BCC     2$                      ;BR IF CARRY CLEAR
016234  004737  027754                          JSR     PC,GETSI                ;GET THE WINDOW
016240  103406                                  BCS     3$                      ;BR IF BIT IS A MARK
016242  104006                                  ERROR   6                       ;ERROR BIT WAS A SPACE
016244  000404                                  BR      3$                      ;CONTINE WITH TEST
016246  004737  027754                  2$:     JSR     PC,GETSI                ;GET THE WINDOW
016252  103001                                  BCC     3$                      ;BR IF BIT IS A SPACE
016254  104006                                  ERROR   6                       ;ERROR BIT WAS A MARK
016256                                  3$:
016256  005203                                  INC     R3                      ;NEXT BIT
016260  022703  000010                          CMP     #10,R3                  ;DONE YET?
016264  001357                                  BNE     1$                      ;BR IF NO
016266  104413  000014                          DATACLK,        14              ;CLOCK TRANSMITTER 14 MORE TICKS
016272  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016274  021264                                  021264                          ;PORT4+LU-13
016276  032761  000040  000004                  BIT     #BIT5,4(R1)             ;RTS SHOULD BE CLEAR NOW
016304  001401                                  BEQ     4$                      ;BR IF YES
016306  104034                                  ERROR   34                      ;ERROR, RTS NOT CLEAR
016310                                  4$:


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

                                                ;  TEST 21
                                                ;-------------

                                        ;;*************************************************************
016310  000004                          TST21:  SCOPE
016312  012737  000021  001202                  MOV     #21,$TSTNM                      ; LOAD THE NO. OF THIS TEST
016320  012737  016472  001442                  MOV     #TST22,NEXT                     ; POINT TO THE START OF NEXT TEST.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
016326  104410                                  MSTCLR                          ;MASTER CLEAR KMC11
016330  012711  004000                          MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
016334  004737  030140                          JSR     PC,OUTRDY               ;WAIT FOR OUT-READY
016340  012761  000001  000004                  MOV     #1,4(R1)                ;SET BIT0 IN PORT4
016346  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016350  122111                                  122111                          ;SET SOM!
016352  012705  000252                          MOV     #252,R5 ;LOAD CHARACTER IN R5 FOR TYPEOUT
016356  004737  030140                          JSR     PC,OUTRDY               ;WAIT FOR OUT-READY
016362  010561  000004                          MOV     R5,4(R1)                ;LOAD PORT4 WITH CHARACTER
016366  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016370  122110                                  122110                          ;LOAD OUT DATA
016372  004737  030006                          JSR     PC,OCOR                 ;WAIT FOR OCOR TO SET
016376  005003                                  CLR     R3                      ;CLEAR BIT COUNTER
016400  010502                                  MOV     R5,R2                   ;LOAD CHARACTER IN R2
016402  104413  000002                          DATACLK,        2               ;2 TICKS TO SET UP TRANSMITTER
```

```
016406  104413  000001          1$:     DATACLK,         1              ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
016412  106002                          RORB    R2                      ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
016414  103005                          BCC     2$                      ;BR IF CARRY CLEAR
016416  004737  027754                  JSR     PC,GETSI                ;GET THE WINDOW
016422  103406                          BCS     3$                      ;BR IF BIT IS A MARK
016424  104006                          ERROR   6                       ;ERROR BIT WAS A SPACE
016426  000404                          BR      3$                      ;CONTINE WITH TEST
016430  004737  027754          2$:     JSR     PC,GETSI                ;GET THE WINDOW
016434  103001                          BCC     3$                      ;BR IF BIT IS A SPACE
016436  104006                          ERROR   6                       ;ERROR BIT WAS A MARK
016440                          3$:
016440  005203                          INC     R3                      ;NEXT BIT
016442  022703  000010                  CMP     #10,R3                  ;DONE YET?
016446  001357                          BNE     1$                      ;BR IF NO
016450  104413  000014                  DATACLK,         14             ;CLOCK TRANSMITTER 14 MORE TICKS
016454  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016456  021264                          021264                          ;PORT4<LU-13
016460  032761  000040  000004          BIT     #BIT5,4(R1)             ;RTS SHOULD BE CLEAR NOW
016466  001401                          BEQ     4$                      ;BR IF YES
016470  104034                          ERROR   34                      ;ERROR, RTS NOT CLEAR
016472                          4$:



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

                                        ;   TEST 22
                                        ;-------------

                                ;;*************************************************************
016472  000004                  TST22:  SCOPE
016474  012737  000022  001202          MOV     #22,$TSTNM                      ; LOAD THE NO. OF THIS TEST
016502  012737  016654  001442          MOV     #TST23,NEXT                     ; POINT TO THE START OF NEXT TEST.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
016510  104410                          MSTCLR                          ;MASTER CLEAR KMC11
016512  012711  004000                  MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
016516  004737  030140                  JSR     PC,OUTRDY               ;WAIT FOR OUT-READY
016522  012761  000001  000004          MOV     #1,4(R1)                ;SET BIT0 IN PORT4
016530  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016532  122111                          122111                          ;SET SOM!
016534  012705  000377                  MOV     #377,R5 ;LOAD CHARACTER IN R5 FOR TYPEOUT
016540  004737  030140                  JSR     PC,OUTRDY               ;WAIT FOR OUT-READY
016544  010561  000004                  MOV     R5,4(R1)                ;LOAD PORT4 WITH CHARACTER
016550  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016552  122110                          122110                          ;LOAD OUT DATA
016554  004737  030006                  JSR     PC,OCOR                 ;WAIT FOR OCOR TO SET
016560  005003                          CLR     R3                      ;CLEAR BIT COUNTER
016562  010502                          MOV     R5,R2                   ;LOAD CHARACTER IN R2
016564  104413  000002                  DATACLK,         2              ;2 TICKS TO SET UP TRANSMITTER
016570  104413  000001          1$:     DATACLK,         1              ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
016574  106002                          RORB    R2                      ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
016576  103005                          BCC     2$                      ;BR IF CARRY CLEAR
```

```
016600  004737  027754                  JSR     PC,GETSI                ;GET THE WINDOW
016604  103406                          BCS     3$                      ;BR IF BIT IS A MARK
016606  104006                          ERROR   6                       ;ERROR BIT WAS A SPACE
016610  000404                          BR      3$                      ;CONTINE WITH TEST
016612  004737  027754          2$:     JSR     PC,GETSI                ;GET THE WINDOW
016616  103001                          BCC     3$                      ;BR IF BIT IS A SPACE
016620  104006                          ERROR   6                       ;ERROR BIT WAS A MARK
016622                          3$:
016622  005203                          INC     R3                      ;NEXT BIT
016624  022703  000010                  CMP     #10,R3                  ;DONE YET?
016630  001357                          BNE     1$                      ;BR IF NO
016632  104413  000014                  DATACLK,        14              ;CLOCK TRANSMITTER 14 MORE TICKS
016636  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016640  021264                          021264                          ;PORT4←LU-13
016642  032761  000040  000004          BIT     #BITS,4(R1)             ;RTS SHOULD BE CLEAR NOW
016650  001401                          BEQ     4$                      ;BR IF YES
016652  104034                          ERROR   34                      ;ERROR, RTS NOT CLEAR
016654                          4$:


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

                                        ;  TEST 23
                                        ;______________

                                ;;********************************************************
016654  000004                  TST23:  SCOPE
016656  012737  000023  001202          MOV     #23,$TSTNM                      ; LOAD THE NO. OF THIS TEST
016664  012737  017062  001442          MOV     #TST24,NEXT                     ; POINT TO THE START OF NEXT TEST.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
016672  104410                          MSTCLR                          ;MASTER CLEAR KMC11
016674  012711  004000                  MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
016700  005003                          CLR     R3                      ;R3 CONTAINS BIT COUNT
016702  005004                          CLR     R4                      ;R4 CONTAINS CHAR TO BE LOADED IN SILO
016704  005005                          CLR     R5                      ;R5 CONTAINS CHARACTER CURRENTLY BEING SHIFTED OUT
016706  004737  030140                  JSR     PC,OUTRDY               ;WAIT FOR OUT-READY
016712  012761  000001  000004          MOV     #1,4(R1)                ;SET BIT0 IN PORT4
016720  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016722  122111                          122111                          ;SET SOM!
016724  004737  030140                  JSR     PC,OUTRDY               ;WAIT FOR OUT-READY
016730  010461  000004                  MOV     R4,4(R1)                ;LOAD PORT4 WITH CHARACTER
016734  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016736  122110                          122110                          ;LOAD OUT DATA
016740  005204                          INC     R4                      ;INCREMENT TO NEXT CHARACTER
016742  004737  030140                  JSR     PC,OUTRDY               ;WAIT FOR OUT-READY
016746  010461  000004                  MOV     R4,4(R1)                ;LOAD PORT4 WITH CHARACTER
016752  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016754  122110                          122110                          ;LOAD OUT DATA
016756  004737  030006                  JSR     PC,OCOR                 ;WAIT FOR OCOR TO SET
016762  104413  000002                  DATACLK,        2               ;2 TICKS TO SET UP TRANSMITTER
```

```
016766  005003                          4$:     CLR     R3                      ;CLEAR BIT COUNTER
016770  010502                                  MOV     R5,R2                   ;LOAD CHARACTER IN R2
016772  104413  000001                  1$:     DATACLK,        1               ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
016776  106002                                  RORB    R2                      ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
017000  103005                                  BCC     2$                      ;BR IF CARRY CLEAR
017002  004737  027754                          JSR     PC,GETSI                ;GET THE WINDOW
017006  103406                                  BCS     3$                      ;BR IF BIT IS A MARK
017010  104006                                  ERROR   6                       ;ERROR BIT WAS A SPACE
017012  000404                                  BR      3$                      ;CONTINE WITH TEST
017014  004737  027754                  2$:     JSR     PC,GETSI                ;GET THE WINDOW
017020  103001                                  BCC     3$                      ;BR IF BIT IS A SPACE
017022  104006                                  ERROR   6                       ;ERROR BIT WAS A MARK
017024                                  3$:
017024  005203                                  INC     R3                      ;NEXT BIT
017026  022703  000010                          CMP     #10,R3                  ;DONE YET?
017032  001357                                  BNE     1$                      ;BR IF NO
017034  005204                                  INC     R4                      ;NEXT CHARACTER
017036  004737  030140                          JSR     PC,OUTRDY               ;WAIT FOR OUT-READY
017042  010461  000004                          MOV     R4,4(R1)                ;LOAD PORT4 WITH CHARACTER
017046  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017050  122110                                  122110                          ;LOAD OUT DATA
017052  005205                                  INC     R5                      ;NEXT CHARACTER
017054  022705  000400                          CMP     #400,R5                 ;DONE YET?
017060  001342                                  BNE     4$                      ;BR IF NO
017062                                  5$:



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

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

                                        ;;*************************************************************
017062  000004                          TST24:  SCOPE
017064  012737  000024  001202                  MOV     #24,$TSTNM                      ; LOAD THE NO. OF THIS TEST
017072  012737  017150  001442                  MOV     #TST25,NEXT                     ; POINT TO THE START OF NEXT TEST.
                                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
017100  104410                                  MSTCLR                          ;MASTER CLEAR KMC11
017102  012711  004000                          MOV     #BIT11,(R1)             ;SET LU LOOP
017106  012702  000012                          MOV     #12,R2                  ;SAVE LU REG FOR TYPEOUT
017112  004737  030024                          JSR     PC,SYNC                 ;SINGLE CLOCK 5 SYNC CHARACTERS
017116  000005                                  5
017120  104413  000054                          DATACLK,        54
017124  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017126  021244                                  021244                          ;PORT4+LU12
017130  016104  000004                          MOV     4(R1),R4                ;PUT "FOUND" IN R4
017134  042704  000277                          BIC     #277,R4                 ;CLEAR UNWANTED BITS
017140  005005                                  CLR     R5                      ;PUT "EXPECTED" IN R5
017142  120504                                  CMPB    R5,R4                   ;IS ACTIVE CLEAR?
017144  001401                                  BEQ     1$                      ;BR IF YES
017146  104040                                  ERROR   40                      ;ERROR ACTIVE IS NOT CLEAR
017150                                  1$:
```

```

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

                                                  ;  TEST 25
                                                  ;_____________
                                           ::*****************************************************
 017150  000004                            TST25:  SCOPE
 017152  012737  000025  001202                    MOV     #25,$TSTNM                 ; LOAD THE NO. OF THIS TEST
 017160  012737  017240  001442                    MOV     #TST26,NEXT                ; POINT TO THE START OF NEXT TEST.
                                                                              ;R1 CONTAINS BASE KMC11 ADDRESS
 017166  104410                                    MSTCLR                     ;MASTER CLEAR KMC11
 017170  012711  004000                            MOV     #BIT11,(R1)        ;SET LU LOOP
 017174  012702  000012                            MOV     #12,R2             ;SAVE LU REG FOR TYPEOUT
 017200  004737  030024                            JSR     PC,SYNC            ;SINGLE CLOCK 5 SYNC CHARACTERS
 017204  000005                                    5
 017206  104413  000064                            DATACLK,        64
 017212  104412                                    ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 017214  021244                                    021244                     ;PORT4←LU12
 017216  016104  000004                            MOV     4(R1),R4           ;PUT "FOUND" IN R4
 017222  042704  000277                            BIC     #277,R4            ;CLEAR UNWANTED BITS
 017226  012705  000100                            MOV     #BIT6,R5                   ;PUT "EXPECTED" IN R5
 017232  120504                                    CMPB    R5,R4              ;IS ACTIVE SET?
 017234  001401                                    BEQ     1$                 ;BR IF YES
 017236  104040                                    ERROR   40                 ;ERROR ACTIVE IS NOT SET
 017240                                    1$:


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

                                                  ;  TEST 26
                                                  ;_____________
                                           ::*****************************************************
 017240  000004                            TST26:  SCOPE
 017242  012737  000026  001202                    MOV     #26,$TSTNM                 ; LOAD THE NO. OF THIS TEST
 017250  012737  017326  001442                    MOV     #TST27,NEXT                ; POINT TO THE START OF NEXT TEST.
                                                                              ;R1 CONTAINS BASE KMC11 ADDRESS
 017256  104410                                    MSTCLR                     ;MASTER CLEAR KMC11
 017260  012711  004000                            MOV     #BIT11,(R1)        ;SET LU LOOP
 017264  012702  000012                            MOV     #12,R2             ;SAVE LU REG FOR TYPEOUT
 017270  004737  030024                            JSR     PC,SYNC            ;SINGLE CLOCK 1 SYNC CHARACTERS
 017274  000001                                    1
 017276  104413  000024                            DATACLK,        24
 017302  104412                                    ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 017304  021244                                    021244                     ;PORT4←LU12
 017306  016104  000004                            MOV     4(R1),R4           ;PUT "FOUND" IN R4
 017312  042704  000277                            BIC     #277,R4            ;CLEAR UNWANTED BITS
 017316  005005                                    CLR     R5                 ;PUT "EXPECTED" IN R5
 017320  120504                                    CMPB    R5,R4              ;IS ACTIVE CLEAR?
```

```
017322  001401                          BEQ     1$                      ;BR IF YES
017324  104040                          ERROR   40                      ;ERROR ACTIVE IS NOT CLEAR
017326                          1$:


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

                                        ;  TEST 27
                                        ;--------------

                                ;;*****************************************************************
017326  000004                  TST27:  SCOPE
017330  012737  000027  001202          MOV     #27,$TSTNM                      ; LOAD THE NO. OF THIS TEST
017336  012737  017416  001442          MOV     #TST30,NEXT                     ; POINT TO THE START OF NEXT TEST.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
017344  104410                          MSTCLR                          ;MASTER CLEAR KMC11
017346  012711  004000                  MOV     #BIT11,(R1)             ;SET LU LOOP
017352  012702  000012                  MOV     #12,R2                  ;SAVE LU REG FOR TYPEOUT
017356  004737  030024                  JSR     PC,SYNC                 ;SINGLE CLOCK 2 SYNC CHARACTERS
017362  000002                          2
017364  104413  000034                  DATACLK,        34
017370  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017372  021244                          021244                          ;PORT4←LU12
017374  016104  000004                  MOV     4(R1),R4                ;PUT "FOUND" IN R4
017400  042704  000277                  BIC     #277,R4                 ;CLEAR UNWANTED BITS
017404  012705  000100                  MOV     #BIT6,R5                        ;PUT "EXPECTED" IN R5
017410  120504                          CMPB    R5,R4                   ;IS ACTIVE SET?
017412  001401                          BEQ     1$                      ;BR IF YES
017414  104040                          ERROR   40                      ;ERROR ACTIVE IS NOT SET
017416                          1$:


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

                                        ;  TEST 30
                                        ;--------------

                                ;;*****************************************************************
017416  000004                  TST30:  SCOPE
017420  012737  000030  001202          MOV     #30,$TSTNM                      ; LOAD THE NO. OF THIS TEST
017426  012737  017570  001442          MOV     #TST31,NEXT                     ; POINT TO THE START OF NEXT TEST.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
017434  104410                          MSTCLR                          ;MASTER CLEAR KMC11
017436  012702  000012                  MOV     #12,R2                  ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
017442  012711  004000                  MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
017446  004737  030172                  JSR     PC,CHAR                 ;LOAD SILO WITH 3 SYNCS
017452  000301                          301                             ;AND A NON-SYNC  (301)
017454  104413  000053                  DATACLK,        53              ;SINGLE CLOCK THE DATA
017460  104414  000002                  TIMER,  2                       ;WAIT FOR INRDY
```

```
017464  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017466  021244                          021244                          ;PORT4+LU 12
017470  016104  000004                  MOV     4(R1),R4                ;PUT "FOUND" IN R4
017474  042704  000357                  BIC     #357,R4                 ;CLEAR UNWANTED BITS
017500  012705  000020                  MOV     #BIT4,R5                        ;PUT "EXPECTED" IN R5
017504  120504                          CMPB    R5,R4                   ;IS INRDY SET?
017506  001401                          BEQ     1$
017510  104040                          ERROR   40                      ;ERROR, INRDY IS NOT SET
017512                          1$:
017512  012761  000200  000004          MOV     #BIT7,4(R1)             ;LOAD PORT4
017520  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017522  122112                          122112                          ;SET IN CLEAR
017524  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017526  021244                          021244                          ;PORT4+LU 12
017530  016104  000004                  MOV     4(R1),R4                ;PUT "FOUND" IN R4
017534  042704  000277                  BIC     #277,R4                 ;CLEAR UNWANTED BITS
017540  005005                          CLR     R5                      ;PUT "EXPECTED" IN R5
017542  120504                          CMPB    R5,R4                   ;IS IN ACTIVE CLEAR?
017544  001401                          BEQ     2$
017546  104040                          ERROR   40                      ;ERROR, IN ACTIVE IS NOT CLEAR
017550                          2$:
017550  016104  000004                  MOV     4(R1),R4                ;PUT "FOUND" IN R4
017554  042704  000357                  BIC     #357,R4                 ;CLEAR UNWANTED BITS
017560  005005                          CLR     R5                      ;PUT "EXPECTED" IN R5
017562  120504                          CMPB    R5,R4                   ;IS INRDY CLEARED?
017564  001401                          BEQ     3$
017566  104040                          ERROR   40                      ;ERROR, INRDY IS NOT CLEARED
017570                          3$:



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

                                        ;  TEST 31
                                        ;--------------

                                ;:*************************************************************
017570  000004                  TST31:  SCOPE
017572  012737  000031  001202          MOV     #31,$TSTNM                      ; LOAD THE NO. OF THIS TEST
017600  012737  017704  001442          MOV     #TST32,NEXT                     ; POINT TO THE START OF NEXT TEST.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
017606  104410                          MSTCLR                          ;MASTER CLEAR KMC11
017610  012702  000012                  MOV     #12,R2                  ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
017614  012711  004000                  MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
017620  004737  030172                  JSR     PC,CHAR                 ;LOAD SILO WITH 3 SYNCS
017624  000000                          0                               ;AND THE CHARACTER 0
017626  104413  000053                  DATACLK,        53              ;SINGLE CLOCK THE DATA
017632  104414  000002                  TIMER,  2                       ;WAIT FOR INRDY
017636  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017640  021244                          021244                          ;PORT4+LU 12
017642  016104  000004                  MOV     4(R1),R4                ;PUT "FOUND" IN R4
017646  042704  000357                  BIC     #357,R4                 ;CLEAR UNWANTED BITS
017652  012705  000020                  MOV     #BIT4,R5                        ;PUT "EXPECTED" IN R5
017656  120504                          CMPB    R5,R4                   ;IS INRDY SET?
```

```
017660  001401                          BEQ     1$
017662  104040                          ERROR   40                      ;ERROR, INRDY IS NOT SET
017664                          1$:
017664  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017666  021204                          021204                          ;PORT4←IN DATA
017670  016104  000004                  MOV     4(R1),R4                ;PUT "FOUND" IN R4
017674  005005                          CLR     R5                      ;PUT "EXPECTED" IN R5
017676  120504                          CMPB    R5,R4                   ;WAS A 0 RECEIVED?
017700  001401                          BEQ     2$
017702  104010                          ERROR   10                      ;ERROR, RECEIVED DATA IS WRONG
017704                          2$:


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

                                        ;    TEST 32
                                        ;--------------
                                ;;*****************************************************************
017704  000004                  TST32:  SCOPE
017706  012737  000032  001202          MOV     #32,$TSTNM                      ; LOAD THE NO. OF THIS TEST
017714  012737  020022  001442          MOV     #TST33,NEXT                     ; POINT TO THE START OF NEXT TEST.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
017722  104410                          MSTCLR                          ;MASTER CLEAR KMC11
017724  012702  000012                  MOV     #12,R2                  ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
017730  012711  004000                  MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
017734  004737  030172                  JSR     PC,CHAR                 ;LOAD SILO WITH 3 SYNCS
017740  000125                          125                             ;AND THE CHARACTER 125
017742  104413  000053                  DATACLK,        53              ;SINGLE CLOCK THE DATA
017746  104414  000002                  TIMER,  2                       ;WAIT FOR INRDY
017752  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017754  021244                          021244                          ;PORT4←LU 12
017756  016104  000004                  MOV     4(R1),R4                ;PUT "FOUND" IN R4
017762  042704  000357                  BIC     #357,R4                 ;CLEAR UNWANTED BITS
017766  012705  000020                  MOV     #BIT4,R5                        ;PUT "EXPECTED" IN R5
017772  120504                          CMPB    R5,R4                   ;IS INRDY SET?
017774  001401                          BEQ     1$
017776  104040                          ERROR   40                      ;ERROR, INRDY IS NOT SET
020000                          1$:
020000  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020002  021204                          021204                          ;PORT4←IN DATA
020004  016104  000004                  MOV     4(R1),R4                ;PUT "FOUND" IN R4
020010  012705  000125                  MOV     #125,R5                 ;PUT "EXPECTED" IN R5
020014  120504                          CMPB    R5,R4                   ;WAS A 125 RECEIVED?
020016  001401                          BEQ     2$
020020  104010                          ERROR   10                      ;ERROR, RECEIVED DATA IS WRONG
020022                          2$:


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

```
                                                 ;;*****************************************************************
                                                 
                                                 ;   TEST 33
                                                 ;---------------
                                                 ;;*****************************************************************
020022  000004                          TST33:   SCOPE
020024  012737  000033  001202                   MOV     #33,$TSTNM              ; LOAD THE NO. OF THIS TEST
020032  012737  020140  001442                   MOV     #TST34,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                         ;R1 CONTAINS BASE KMC11 ADDRESS
020040  104410                                   MSTCLR                  ;MASTER CLEAR KMC11
020042  012702  000012                           MOV     #12,R2          ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
020046  012711  004000                           MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
020052  004737  030172                           JSR     PC,CHAR         ;LOAD SILO WITH 3 SYNCS
020056  000252                                   252                     ;AND THE CHARACTER 252
020060  104413  000053                           DATACLK,        53      ;SINGLE CLOCK THE DATA
020064  104414  000002                           TIMER,  2               ;WAIT FOR INRDY
020070  104412                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020072  021244                                   021244                  ;PORT4←LU 12
020074  016104  000004                           MOV     4(R1),R4        ;PUT "FOUND" IN R4
020100  042704  000357                           BIC     #357,R4         ;CLEAR UNWANTED BITS
020104  012705  000020                           MOV     #BIT4,R5                ;PUT "EXPECTED" IN R5
020110  120504                                   CMPB    R5,R4           ;IS INRDY SET?
020112  001401                                   BEQ     1$
020114  104040                                   ERROR   40              ;ERROR, INRDY IS NOT SET
020116                                  1$:
020116  104412                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020120  021204                                   021204                  ;PORT4←IN DATA
020122  016104  000004                           MOV     4(R1),R4        ;PUT "FOUND" IN R4
020126  012705  000252                           MOV     #252,R5         ;PUT "EXPECTED" IN R5
020132  120504                                   CMPB    R5,R4           ;WAS A 252 RECEIVED?
020134  001401                                   BEQ     2$
020136  104010                                   ERROR   10              ;ERROR, RECEIVED DATA IS WRONG
020140                                  2$:


                                                 ;;************************** TEST 34 *************************
                                                 ;*DDCMP BASIC RECEICER TEST
                                                 ;*SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 377
                                                 ;*VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED
                                                 ;;*****************************************************************
                                                 
                                                 ;   TEST 34
                                                 ;---------------
                                                 ;;*****************************************************************
020140  000004                          TST34:   SCOPE
020142  012737  000034  001202                   MOV     #34,$TSTNM              ; LOAD THE NO. OF THIS TEST
020150  012737  020256  001442                   MOV     #TST35,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                         ;R1 CONTAINS BASE KMC11 ADDRESS
020156  104410                                   MSTCLR                  ;MASTER CLEAR KMC11
020160  012702  000012                           MOV     #12,R2          ;SAVE REG ADDRESS IN R2 FOR TYPEOUT
020164  012711  004000                           MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
020170  004737  030172                           JSR     PC,CHAR         ;LOAD SILO WITH 3 SYNCS
020174  000377                                   377                     ;AND THE CHARACTER 377
020176  104413  000053                           DATACLK,        53      ;SINGLE CLOCK THE DATA
020202  104414  000002                           TIMER,  2               ;WAIT FOR INRDY
020206  104412                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
 020210  021244                           021244                  ;PORT4+LU 12
 020212  016104  000004                   MOV     4(R1),R4        ;PUT "FOUND" IN R4
 020216  042704  000357                   BIC     #357,R4         ;CLEAR UNWANTED BITS
 020222  012705  000020                   MOV     #BIT4,R5                ;PUT "EXPECTED" IN R5
 020226  120504                           CMPB    R5,R4           ;IS INRDY SET?
 020230  001401                           BEQ     1$
 020232  104040                           ERROR   40              ;ERROR, INRDY IS NOT SET
 020234                           1$:
 020234  104412                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 020236  021204                           021204                  ;PORT4+IN DATA
 020240  016104  000004                   MOV     4(R1),R4        ;PUT "FOUND" IN R4
 020244  012705  000377                   MOV     #377,R5         ;PUT "EXPECTED" IN R5
 020250  120504                           CMPB    R5,R4           ;WAS A 377 RECEIVED?
 020252  001401                           BEQ     2$
 020254  104010                           ERROR   10              ;ERROR, RECEIVED DATA IS WRONG
 020256                           2$:


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

                                          ;  TEST 35
                                          ;______________

                                  ;;*********************************************************
 020256  000004                   TST35:  SCOPE
 020260  012737  000035  001202           MOV     #35,$TSTNM              ; LOAD THE NO. OF THIS TEST
 020266  012737  020406  001442           MOV     #TST36,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
 020274  104410                           MSTCLR                  ;MASTER CLEAR KMC11
 020276  005037  030610                   CLR     SCHAR           ;START BINARY COUNT AT ZERO
 020302  005037  030612                   CLR     STUFLG          ;CLEAR BITSTUFF FLAG
 020306  005002                           CLR     R2              ;R2 IS "EXPECTED" DATA
 020310  012703  000073                   MOV     #73,R3          ;R3 IS CHARACTER COUNT
 020314  012711  004000                   MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
 020320  004737  030350                   JSR     PC,SILOLD       ;LOAD SILO WITH COUNT PATTERN
 020324  104413  000043                   DATACLK,        43      ;SYNC RECEIVER AND GET IT ACTIVE
 020330  104413  000730           1$:     DATACLK,        730     ;CLOCK IN 73 CHARACTERS
 020334  004737  030614           4$:     JSR     PC,INRDY        ;WAIT FOR INRDY
 020340  104412                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 020342  021204                           021204                  ;PORT4+IN DATA
 020344  016104  000004                   MOV     4(R1),R4        ;PUT "FOUND" IN R4
 020350  010205                           MOV     R2,R5           ;PUT "EXPECTED" IN R5
 020352  120504                           CMPB    R5,R4           ;IS DATA CORRECT?
 020354  001401                           BEQ     2$              ;BR IF YES
 020356  104010                           ERROR   10              ;DATA ERROR
 020360  005202                   2$:     INC     R2              ;NEXT CHARACTER
 020362  022702  000400                   CMP     #400,R2         ;ALL DONE?
 020366  001407                           BEQ     3$              ;BR IF YES
 020370  005303                           DEC     R3              ;DECREMENT CHARACTER COUNT
 020372  001360                           BNE     4$              ;BR IF SILO NOT EMPTY
 020374  004737  030350                   JSR     PC,SILOLD       ;LOAD SILO WITH MORE OF COUNT PATTERN
 020400  012703  000073                   MOV     #73,R3          ;RELOAD CHARACTER COUNT
 020404  000751                           BR      1$              ;CONTINUE
```

```
020406                                  3$:



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

                                                ;  TEST 36
                                                ;--------------
                                        ;;*******************************************************************
020406  000004                          TST36:  SCOPE
020410  012737  000036  001202                  MOV     #36,$TSTNM              ; LOAD THE NO. OF THIS TEST
020416  012737  020546  001442                  MOV     #TST37,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
020424  104410                                  MSTCLR                  ;MASTER CLEAR KMC11
020426  005037  030610                          CLR     SCHAR           ;START BINARY COUNT AT ZERO
020432  005037  030612                          CLR     STUFLG          ;CLEAR BITSTUFF FLAG
020436  005002                                  CLR     R2              ;R2 IS "EXPECTED" DATA
020440  012703  000073                          MOV     #73,R3          ;R3 IS CHARACTER COUNT
020444  005011                                  CLR     (R1)            ;CLEAR LU LOOP IN MAINT REG
020446  012761  000040  000004                  MOV     #BIT5,4(R1)     ;LOAD PORT4
020454  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020456  122112                                  122112                  ;SET LU LOOP IN LU REG 12
020460  004737  030350                          JSR     PC,SILOLD       ;LOAD SILO WITH COUNT PATTERN
020464  104413  000043                          DATACLK,        43      ;SYNC RECEIVER AND GET IT ACTIVE
020470  104413  000730                  1$:     DATACLK,        730     ;CLOCK IN 73 CHARACTERS
020474  004737  030614                  4$:     JSR     PC,INRDY        ;WAIT FOR INRDY
020500  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020502  021204                                  021204                  ;PORT4+IN DATA
020504  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
020510  010205                                  MOV     R2,R5           ;PUT "EXPECTED" IN R5
020512  120504                                  CMPB    R5,R4           ;IS DATA CORRECT?
020514  001401                                  BEQ     2$              ;BR IF YES
020516  104010                                  ERROR   10              ;DATA ERROR
020520  005202                          2$:     INC     R2              ;NEXT CHARACTER
020522  022702  000400                          CMP     #400,R2         ;ALL DONE?
020526  001407                                  BEQ     3$              ;BR IF YES
020530  005303                                  DEC     R3              ;DECREMENT CHARACTER COUNT
020532  001360                                  BNE     4$              ;BR IF SILO NOT EMPTY
020534  004737  030350                          JSR     PC,SILOLD       ;LOAD SILO WITH MORE OF COUNT PATTERN
020540  012703  000073                          MOV     #73,R3          ;RELOAD CHARACTER COUNT
020544  000751                                  BR      1$              ;CONTINUE
020546                                  3$:


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

```

                                                 ;  TEST 37
                                                 ;--------------

                                         ;;*******************************************************
020546  000004                           TST37:   SCOPE
020550  012737  000037  001202                    MOV      #37,$TSTNM               ; LOAD THE NO. OF THIS TEST
020556  012737  020706  001442                    MOV      #TST40,NEXT              ; POINT TO THE START OF NEXT TEST.
                                                                            ;R1 CONTAINS BASE KMC11 ADDRESS
020564  104410                                    MSTCLR                    ;MASTER CLEAR KMC11
020566  012711  004000                            MOV      #BIT11,(R1)      ;SET LINE UNIT LOOP
020572  004737  030172                            JSR      PC,CHAR          ;LOAD SILO WITH 3 SYNCS
020576  000301                                    301                       ;AND A 301
020600  104413  000073                            DATACLK,          73      ;CLOCK THE 301 IN AND 20 EXTRA TICKS
020604  004737  030614                            JSR      PC,INRDY         ;WAIT FOR INRDY
020610  104412                                    ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020612  021204                                    021204                    ;PORT4+IN DATA
020614  016104  000004                            MOV      4(R1),R4         ;PUT "FOUND" IN R4
020620  012705  000301                            MOV      #301,R5          ;PUT "EXPECTED" IN R5
020624  120504                                    CMPB     R5,R4            ;WAS A 301 RECEIVED?
020626  001401                                    BEQ      1$
020630  104010                                    ERROR    10               ;ERROR FIRST CHARACTER INCORRECT
020632  004737  030614                   1$:      JSR      PC,INRDY         ;WAIT FOR INRDY
020636  104412                                    ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020640  021204                                    021204                    ;PORT4+IN DATA
020642  016104  000004                            MOV      4(R1),R4         ;PUT "FOUND" IN R4
020646  012705  000377                            MOV      #377,R5          ;PUT "EXPECTED" IN R5
020652  120504                                    CMPB     R5,R4            ;WAS A 377 RECEIVED?
020654  001401                                    BEQ      2$
020656  104010                                    ERROR    10               ;ERROR, 377 WAS NOT RECEIVED
020660  004737  030614                   2$:      JSR      PC,INRDY         ;WAIT FOR INRDY
020664  104412                                    ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020666  021204                                    021204                    ;PORT4+IN DATA
020670  016104  000004                            MOV      4(R1),R4         ;PUT "FOUND" IN R4
020674  012705  000377                            MOV      #377,R5          ;PUT "EXPECTED" IN R5
020700  120504                                    CMPB     R5,R4            ;WAS A 377 RECEIVED?
020702  001401                                    BEQ      3$
020704  104010                                    ERROR    10               ;ERROR, 177 WAS NOT RECEIVED
020706                                   3$:


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


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

                                         ;;*******************************************************
020706  000004                           TST40:   SCOPE
020710  012737  000040  001202                    MOV      #40,$TSTNM               ; LOAD THE NO. OF THIS TEST
020716  012737  021104  001442                    MOV      #TST41,NEXT              ; POINT TO THE START OF NEXT TEST.
                                                                            ;R1 CONTAINS BASE KMC11 ADDRESS
020724  104410                                    MSTCLR                    ;MASTER CLEAR KMC11
020726  032737  020000  002050                    BIT      #BIT13,STAT1     ;IS LINE UNIT M8202?
```

```
020734  001004                               BNE     .+12            ;BR IF YES (DO TEST EVEN IF NO LOOP-BACK CONN)
020736  032737  040000  002050               BIT     #BIT14,STAT1    ;IS TURNAROUND CONNECTOR ON?
020744  001457                               BEQ     2$              ;SKIP TEST IF NO
020746  005011                               CLR     (R1)            ;CLEAR LINE UNIT LOOP
020750  012761  000100  000004               MOV     #100,4(R1)      ;LOAD PORT4
020756  104412                               ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020760  122113                               122113                  ;SET DTR
020762  104414  000002                       TIMER,  2               ;WAIT
020766  104412                               ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020770  021264                               021264                  ;PORT4+LU13
020772  016104  000004                       MOV     4(R1),R4        ;PUT "FOUND" IN R4
020776  042704  000023                       BIC     #23,R4          ;CLEAR UNWANTED BITS
021002  012705  000310                       MOV     #310,R5         ;PUT "EXPECTED" IN R5
021006  032737  020000  002050               BIT     #BIT13,STAT1    ;IS LINE UNIT M8202?
021014  001402                               BEQ     .+6             ;BR IF NO
021016  042705  000200                       BIC     #BIT7,R5        ;NO RING ON M8202
021022  120504                               CMPB    R5,R4           ;ARE RING AND MRDY SET?
021024  001401                               BEQ     1$
021026  104011                               ERROR   11              ;ERROR, RING OR MRDY NOT SET
021030  005061  000004               1$:     CLR     4(R1)           ;CLEAR PORT4
021034  104412                               ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021036  122113                               122113                  ;CLEAR DTR
021040  104414  000002                       TIMER,  2
021044  104412                               ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021046  021264                               021264                  ;PORT4+LU13
021050  016104  000004                       MOV     4(R1),R4        ;PUT "FOUND" IN R4
021054  042704  000023                       BIC     #23,R4          ;CLEAR UNWANTED BITS
021060  005005                               CLR     R5              ;PUT "EXPECTED" IN R5
021062  032737  020000  002050               BIT     #BIT13,STAT1    ;IS LINE UNIT M8202?
021070  001402                               BEQ     .+6             ;BR IF NO
021072  052705  000010                       BIS     #BIT3,R5        ;MRDY SET ON M8202
021076  120504                               CMPB    R5,R4           ;ARE RING AND MRDY CLEAR?
021100  001401                               BEQ     2$
021102  104011                               ERROR   11              ;ERROR, RING OR MRDY NOT CLEAR
021104                               2$:



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

                                             ;  TEST 41
                                             ;--------------

                                     ;;*************************************************************
021104  000004                       TST41:  SCOPE
021106  012737  000041  001202               MOV     #41,$TSTNM              ; LOAD THE NO. OF THIS TEST
021114  012737  021264  001442               MOV     #TST42,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                     ;R1 CONTAINS BASE KMC11 ADDRESS
021122  104410                               MSTCLR                  ;MASTER CLEAR KMC11
021124  032737  020000  002050               BIT     #BIT13,STAT1    ;IS LINE UNIT M8202?
021132  001004                               BNE     .+12            ;BR IF YES (DO TEST EVEN IF NO LOOP-BACK CONN)
021134  032737  040000  002050               BIT     #BIT14,STAT1    ;IS TURNAROUND CONNECTOR ON?
021142  001450                               BEQ     1$              ;SKIP TEST IF NO
021144  012711  004000                       MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
```

```
 021150  012761  000100  000004            MOV     #100,   4(R1)            ;LOAD PORT4
 021156  104412                            ROMCLK                           ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 021160  122113                            122113                           ;CLEAR ALL MODEM SIGNALS,EXCEPT DTR
 021162  104414  000002                    TIMER,  2                        ;WAIT
 021166  012761  000001  000004            MOV     #1,4(R1)                 ;LOAD PORT4
 021174  104412                            ROMCLK                           ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 021176  122111                            122111                           ;SET SOM
 021200  004537  031256                    JSR     R5,MESLD                 ;FILL OUT DATA SILO
 021204  031540                            MESDAT                           ;WITH 64 CHARACTERS
 021206  000100                            64.
 021210  012700  000050                    MOV     #50,R0                   ;PREPARE FOR DELAY
 021214  005011                            CLR     (R1)                     ;CLEAR LINE UNIT LOOP
 021216                           2$:
 021216  104412                            ROMCLK                           ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 021220  021264                            021264                           ;PORT4←LU13
 021222  016104  000004                    MOV     4(R1),R4                 ;PUT "FOUND" IN R4
 021226  042704  000023                    BIC     #23,R4                   ;CLEAR UNWANTED BITS
 021232  012705  000354                    MOV     #354,R5                  ;PUT "EXPECTED" IN R5
 021236  032737  020000  002050            BIT     #BIT13,STAT1             ;IS LINE UNIT M8202?
 021244  001402                            BEQ     .+6                      ;BR IF NO
 021246  042705  000200                    BIC     #BIT7,R5                 ;NO RING ON M8202
 021252  120504                            CMPB    R5,R4                    ;COMPARE EXPECTED AND FOUND
 021254  001403                            BEQ     1$                       ;BR IF OK
 021256  005300                            DEC     R0                       ;DEC DELAY COUNT
 021260  001356                            BNE     2$                       ;BR IF NOT ZERO
 021262  104011                            ERROR   11                       ;ERROR, ALL SIGNALS ARE NOT SET
 021264                           1$:


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

                                           ;  TEST 42
                                           ;--------------

                                  ;;*************************************************************
 021264  000004                   TST42:   SCOPE
 021266  012737  000042  001202            MOV     #42,$TSTNM                  ; LOAD THE NO. OF THIS TEST
 021274  012737  021600  001442            MOV     #TST43,NEXT                 ; POINT TO THE START OF NEXT TEST.
 021302  012737  021316  001444            MOV     #64$,LOCK                   ; ADDRESS FOR LOCK ON DATA.
                                                                            ;R1 CONTAINS BASE KMC11 ADDRESS
 021310  104410                            MSTCLR                           ;MASTER CLEAR KMC11
 021312  012711  004000                    MOV     #BIT11,(R1)              ;SET LU LOOP
 021316  004737  031320           64$:     JSR     PC,CLRIO                 ;CLEAR BCC REGISTERS
 021322  005000                            CLR     R0                       ;START SHIFT COUNTER AT ZERO
 021324  012737  120001  030754            MOV     #CRC16,XPOLY             ;LOAD POLYNOMIAL FOR SOFTWARE BCC
 021332  012737  000000  021372            MOV     #0,66$                   ;LOAD CHAR FOR SOFTWARE BCC
 021340  005037  021374                    CLR     67$                      ;CLEAR OLD SOFTWARE BCC
 021344  004737  030760                    JSR     PC,BCCLD                 ;LOAD OUT SILO WITH 2 SYNCS
 021350  000000                            0                                ;AND THE CHARACTER 0
 021352  104413  000021                    DATACLK,        21               ;GET TRANSMITTER ACTIVE
 021356  104413  000001           65$:     DATACLK,        1                ;SHIFT BCC ONCE
 021362  005200                            INC     R0                       ;BUMP SHIFT COUNT
```

```
021364  004537  030650                          JSR     R5,SIMBCC       ;CALCULATE SOFTWARE BCC LSB
021370  000001                                  1                       ;ONE SHIFT
021372  000000                          66$:    0                       ;DATA CHARACTER
021374  000000                          67$:    0                       ;OLD BCC
021376  103405                                  BCS     68$             ;BR IF SOFT BCC LSB IS SET
021400  004737  031072                          JSR     PC,GETQO        ;GET HARDWARE TRANSMITTER BCC LSB
021404  103006                                  BCC     69$             ;BR IF HARD BCC LSB IS CLEAR
021406  104012                                  ERROR   12              ;ERROR, BCC LSB IS SET
021410  000404                                  BR      69$             ;CONTINUE
021412  004737  031072                  68$:    JSR     PC,GETQO        ;GET HARDWARE TRANSMITTER BCC LSB
021416  103401                                  BCS     69$             ;BR IF HARD BCC LSB IS SET
021420  104016                                  ERROR   16              ;ERROR, HARD BCC LSB IS CLEAR
021422                                  69$:
021422  006037  021372                          ROR     66$             ;SHIFT SOFT DATA
021426  013737  030756  021374                  MOV     CALBCC,67$      ;LOAD OLD SOFT BCC
021434  022700  000010                          CMP     #10,R0          ;DONE YET?
021440  001346                                  BNE     65$             ;BR IF NOT DONE
021442  104405                                  SCOP1                   ;SCOPE SUBTEST (SW09=1)
021444  012737  021452  001444                  MOV     #71$,LOCK       ;NEW SCOPE1
021452  004737  031320                  71$:    JSR     PC,CLRIO        ;CLEAR BCC REGISTERS
021456  005000                                  CLR     R0              ;START SHIFT COUNTER AT ZERO
021460  012737  120001  030754                  MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL FOR SOFTWARE BCC
021466  012737  000000  021526                  MOV     #0,73$          ;LOAD CHAR FOR SOFTWARE BCC
021474  005037  021530                          CLR     74$             ;CLEAR OLD SOFTWARE BCC
021500  004737  030760                          JSR     PC,BCCLD        ;LOAD OUT SILO WITH 2 SYNCS
021504  000000                                  0                       ;AND THE CHARACTER 0
021506  104413  000032                          DATACLK,        32      ;GET RECEIVER ACTIVE
021512  104413  000001                  72$:    DATACLK,        1       ;SHIFT BCC ONCE
021516  005200                                  INC     R0              ;BUMP SHIFT COUNT
021520  004537  030650                          JSR     R5,SIMBCC       ;CALCULATE SOFTWARE BCC LSB
021524  000001                                  1                       ;ONE SHIFT
021526  000000                          73$:    0                       ;DATA CHARACTER
021530  000000                          74$:    0                       ;OLD BCC
021532  103405                                  BCS     75$             ;BR IF SOFT BCC LSB IS SET
021534  004737  031104                          JSR     PC,GETQI        ;GET HARDWARE RECEIVER BCC LSB
021540  103006                                  BCC     76$             ;BR IF HARD BCC LSB IS CLEAR
021542  104013                                  ERROR   13              ;ERROR, BCC LSB IS SET
021544  000404                                  BR      76$             ;CONTINUE
021546  004737  031104                  75$:    JSR     PC,GETQI        ;GET HARDWARE RECEIVER BCC LSB
021552  103401                                  BCS     76$             ;BR IF HARD BCC LSB IS SET
021554  104017                                  ERROR   17              ;ERROR, BCC LSB IS CLEAR
021556                                  76$:
021556  006037  021526                          ROR     73$             ;SHIFT SOFT DATA
021562  013737  030756  021530                  MOV     CALBCC,74$      ;LOAD OLD SOFT BCC
021570  022700  000010                          CMP     #10,R0          ;DONE YET?
021574  001346                                  BNE     72$             ;BR IF NOT DONE
021576  104405                                  SCOP1                   ;SCOPE SUBTEST (SW09=1)
021600                                  77$:



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

```

                                        ;  TEST 43
                                        ;--------------
                                ;;*************************************************************
021600  000004                  TST43:  SCOPE
021602  012737  000043  001202          MOV     #43,$TSTNM              ; LOAD THE NO. OF THIS TEST
021610  012737  022114  001442          MOV     #TST44,NEXT             ; POINT TO THE START OF NEXT TEST.
021616  012737  021632  001444          MOV     #64$,LOCK               ; ADDRESS FOR LOCK ON DATA.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
021624  104410                          MSTCLR                  ;MASTER CLEAR KMC11
021626  012711  004000                  MOV     #BIT11,(R1)     ;SET LU LOOP
021632  004737  031320          64$:    JSR     PC,CLRIO        ;CLEAR BCC REGISTERS
021636  005000                          CLR     R0              ;START SHIFT COUNTER AT ZERO
021640  012737  120001  030754          MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL FOR SOFTWARE BCC
021646  012737  000377  021706          MOV     #377,66$;       ;LOAD CHAR FOR SOFTWARE BCC
021654  005037  021710                  CLR     67$             ;CLEAR OLD SOFTWARE BCC
021660  004737  030760                  JSR     PC,BCCLD        ;LOAD OUT SILO WITH 2 SYNCS
021664  000377                          377                     ;AND THE CHARACTER 377
021666  104413  000021                  DATACLK,        21      ;GET TRANSMITTER ACTIVE
021672  104413  000001          65$:    DATACLK,        1       ;SHIFT BCC ONCE
021676  005200                          INC     R0              ;BUMP SHIFT COUNT
021700  004537  030650                  JSR     R5,SIMBCC       ;CALCULATE SOFTWARE BCC LSB
021704  000001                          1                       ;ONE SHIFT
021706  000000                  66$:    0                       ;DATA CHARACTER
021710  000000                  67$:    0                       ;OLD BCC
021712  103405                          BCS     68$             ;BR IF SOFT BCC LSB IS SET
021714  004737  031072                  JSR     PC,GETQ0        ;GET HARDWARE TRANSMITTER BCC LSB
021720  103006                          BCC     69$             ;BR IF HARD BCC LSB IS CLEAR
021722  104012                          ERROR   12              ;ERROR, BCC LSB IS SET
021724  000404                          BR      69$             ;CONTINUE
021726  004737  031072          68$:    JSR     PC,GETQ0        ;GET HARDWARE TRANSMITTER BCC LSB
021732  103401                          BCS     69$             ;BR IF HARD BCC LSB IS SET
021734  104016                          ERROR   16              ;ERROR, HARD BCC LSB IS CLEAR
021736                          69$:
021736  006037  021706                  ROR     66$             ;SHIFT SOFT DATA
021742  013737  030756  021710          MOV     CALBCC,67$      ;LOAD OLD SOFT BCC
021750  022700  000010                  CMP     #10,R0          ;DONE YET?
021754  001346                          BNE     65$             ;BR IF NOT DONE
021756  104405                          SCOP1                   ;SCOPE SUBTEST (SW09=1)
021760  012737  021766  001444          MOV     #71$,LOCK       ;NEW SCOPE1
021766  004737  031320          71$:    JSR     PC,CLRIO        ;CLEAR BCC REGISTERS
021772  005000                          CLR     R0              ;START SHIFT COUNTER AT ZERO
021774  012737  120001  030754          MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL FOR SOFTWARE BCC
022002  012737  000377  022042          MOV     #377,73$;       ;LOAD CHAR FOR SOFTWARE BCC
022010  005037  022044                  CLR     74$             ;CLEAR OLD SOFTWARE BCC
022014  004737  030760                  JSR     PC,BCCLD        ;LOAD OUT SILO WITH 2 SYNCS
022020  000377                          377                     ;AND THE CHARACTER 377
022022  104413  000032                  DATACLK,        32      ;GET RECEIVER ACTIVE
022026  104413  000001          72$:    DATACLK,        1       ;SHIFT BCC ONCE
022032  005200                          INC     R0              ;BUMP SHIFT COUNT
022034  004537  030650                  JSR     R5,SIMBCC       ;CALCULATE SOFTWARE BCC LSB
022040  000001                          1                       ;ONE SHIFT
022042  000000                  73$:    0                       ;DATA CHARACTER
022044  000000                  74$:    0                       ;OLD BCC
022046  103405                          BCS     75$             ;BR IF SOFT BCC LSB IS SET
022050  004737  031104                  JSR     PC,GETQI        ;GET HARDWARE RECEIVER BCC LSB
```

```
022054  103006                          BCC     76$                     ;BR IF HARD BCC LSB IS CLEAR
022056  104013                          ERROR   13                      ;ERROR, BCC LSB IS SET
022060  000404                          BR      76$                     ;CONTINUE
022062  004737  031104          75$:    JSR     PC,GETQI                ;GET HARDWARE RECEIVER BCC LSB
022066  103401                          BCS     76$                     ;BR IF HARD BCC LSB IS SET
022070  104017                          ERROR   17                      ;ERROR, BCC LSB IS CLEAR
022072                          76$:
022072  006037  022042                  ROR     73$                     ;SHIFT SOFT DATA
022076  013737  030756  022044          MOV     CALBCC,74$              ;LOAD OLD SOFT BCC
022104  022700  000010                  CMP     #10,R0                  ;DONE YET?
022110  001346                          BNE     72$                     ;BR IF NOT DONE
022112  104405                          SCOP1                           ;SCOPE SUBTEST (SW09=1)
022114                          77$:


                                        ;************************** TEST 44 **************************
                                        ;*TEST OF CRC OPERATION
                                        ;*USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
                                        ;*125, VERIFY THE LSB OF THE BCC ON EACH SHIFT
                                        ;*TEST TRANSMITTER FIRST THEN THE RECEIVER BCC
                                        ;:************************************************************

                                        ;  TEST 44
                                        ;  -------------

                                ;;*************************************************************
022114  000004                  TST44:  SCOPE
022116  012737  000044  001202          MOV     #44,$TSTNM                      ; LOAD THE NO. OF THIS TEST
022124  012737  022430  001442          MOV     #TST45,NEXT                     ; POINT TO THE START OF NEXT TEST.
022132  012737  022146  001444          MOV     #64$,LOCK                       ; ADDRESS FOR LOCK ON DATA.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
022140  104410                          MSTCLR                          ;MASTER CLEAR KMC11
022142  012711  004000                  MOV     #BIT11,(R1)             ;SET LU LOOP
022146  004737  031320          64$:    JSR     PC,CLRIO                ;CLEAR BCC REGISTERS
022152  005000                          CLR     R0                      ;START SHIFT COUNTER AT ZERO
022154  012737  120001  030754          MOV     #CRC16,XPOLY            ;LOAD POLYNOMIAL FOR SOFTWARE BCC
022162  012737  000125  022222          MOV     #125,66$;               ;LOAD CHAR FOR SOFTWARE BCC
022170  005037  022224                  CLR     67$                     ;CLEAR OLD SOFTWARE BCC
022174  004737  030760                  JSR     PC,BCCLD                ;LOAD OUT SILO WITH 2 SYNCS
022200  000125                          125                             ;AND THE CHARACTER 125
022202  104413  000021                  DATACLK,        21              ;GET TRANSMITTER ACTIVE
022206  104413  000001          65$:    DATACLK,        1               ;SHIFT BCC ONCE
022212  005200                          INC     R0                      ;BUMP SHIFT COUNT
022214  004537  030650                  JSR     R5,SIMBCC               ;CALCULATE SOFTWARE BCC LSB
022220  000001                          1                               ;ONE SHIFT
022222  000000                  66$:    0                               ;DATA CHARACTER
022224  000000                  67$:    0                               ;OLD BCC
022226  103405                          BCS     68$                     ;BR IF SOFT BCC LSB IS SET
022230  004737  031072                  JSR     PC,GETQO                ;GET HARDWARE TRANSMITTER BCC LSB
022234  103006                          BCC     69$                     ;BR IF HARD BCC LSB IS CLEAR
022236  104012                          ERROR   12                      ;ERROR, BCC LSB IS SET
022240  000404                          BR      69$                     ;CONTINUE
022242  004737  031072          68$:    JSR     PC,GETQO                ;GET HARDWARE TRANSMITTER BCC LSB
022246  103401                          BCS     69$                     ;BR IF HARD BCC LSB IS SET
022250  104016                          ERROR   16                      ;ERROR, HARD BCC LSB IS CLEAR
022252                          69$:
022252  006037  022222                  ROR     66$                     ;SHIFT SOFT DATA
```

```
022256  013737  030756  022224             MOV     CALBCC,67$         ;LOAD OLD SOFT BCC
022264  022700  000010                     CMP     #10,R0             ;DONE YET?
022270  001346                             BNE     65$                ;BR IF NOT DONE
022272  104405                             SCOP1                      ;SCOPE SUBTEST (SW09=1)
022274  012737  022302  001444             MOV     #71$,LOCK          ;NEW SCOPE1
022302  004737  031320             71$:    JSR     PC,CLRIO           ;CLEAR BCC REGISTERS
022306  005000                             CLR     R0                 ;START SHIFT COUNTER AT ZERO
022310  012737  120001  030754             MOV     #CRC16,XPOLY       ;LOAD POLYNOMIAL FOR SOFTWARE BCC
022316  012737  000125  022356             MOV     #125,73$;          ;LOAD CHAR FOR SOFTWARE BCC
022324  005037  022360                     CLR     74$                ;CLEAR OLD SOFTWARE BCC
022330  004737  030760                     JSR     PC,BCCLD           ;LOAD OUT SILO WITH 2 SYNCS
022334  000125                             125                        ;AND THE CHARACTER 125
022336  104413  000032                     DATACLK,          32       ;GET RECEIVER ACTIVE
022342  104413  000001             72$:    DATACLK,          1        ;SHIFT BCC ONCE
022346  005200                             INC     R0                 ;BUMP SHIFT COUNT
022350  004537  030650                     JSR     R5,SIMBCC          ;CALCULATE SOFTWARE BCC LSB
022354  000001                             1                          ;ONE SHIFT
022356  000000                     73$:    0                          ;DATA CHARACTER
022360  000000                     74$:    0                          ;OLD BCC
022362  103405                             BCS     75$                ;BR IF SOFT BCC LSB IS SET
022364  004737  031104                     JSR     PC,GETQI           ;GET HARDWARE RECEIVER BCC LSB
022370  103006                             BCC     76$                ;BR IF HARD BCC LSB IS CLEAR
022372  104013                             ERROR   13                 ;ERROR, BCC LSB IS SET
022374  000404                             BR      76$                ;CONTINUE
022376  004737  031104             75$:    JSR     PC,GETQI           ;GET HARDWARE RECEIVER BCC LSB
022402  103401                             BCS     76$                ;BR IF HARD BCC LSB IS SET
022404  104017                             ERROR   17                 ;ERROR, BCC LSB IS CLEAR
022406                             76$:
022406  006037  022356                     ROR     73$                ;SHIFT SOFT DATA
022412  013737  030756  022360             MOV     CALBCC,74$         ;LOAD OLD SOFT BCC
022420  022700  000010                     CMP     #10,R0             ;DONE YET?
022424  001346                             BNE     72$                ;BR IF NOT DONE
022426  104405                             SCOP1                      ;SCOPE SUBTEST (SW09=1)
022430                             77$:



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

                                           ;  TEST 45
                                           ;-------------

                                   ;;*****************************************************************
022430  000004                     TST45:  SCOPE
022432  012737  000045  001202             MOV     #45,$TSTNM               ; LOAD THE NO. OF THIS TEST
022440  012737  022744  001442             MOV     #TST46,NEXT              ; POINT TO THE START OF NEXT TEST.
022446  012737  022462  001444             MOV     #64$,LOCK                ; ADDRESS FOR LOCK ON DATA.
                                                                      ;R1 CONTAINS BASE KMC11 ADDRESS
022454  104410                             MSTCLR                     ;MASTER CLEAR KMC11
022456  012711  004000                     MOV     #BIT11,(R1)        ;SET LU LOOP
022462  004737  031320             64$:    JSR     PC,CLRIO           ;CLEAR BCC REGISTERS
022466  005000                             CLR     R0                 ;START SHIFT COUNTER AT ZERO
022470  012737  120001  030754             MOV     #CRC16,XPOLY       ;LOAD POLYNOMIAL FOR SOFTWARE BCC
```

```
022476  012737  000252  022536           MOV     #252,66$;               ;LOAD CHAR FOR SOFTWARE BCC
022504  005037  022540                   CLR     67$                     ;CLEAR OLD SOFTWARE BCC
022510  004737  030760                   JSR     PC,BCCLD                ;LOAD OUT SILO WITH 2 SYNCS
022514  000252                           252                             ;AND THE CHARACTER 252
022516  104413  000021                   DATACLK,        21              ;GET TRANSMITTER ACTIVE
022522  104413  000001          65$:     DATACLK,        1               ;SHIFT BCC ONCE
022526  005200                           INC     R0                      ;BUMP SHIFT COUNT
022530  004537  030650                   JSR     R5,SIMBCC               ;CALCULATE SOFTWARE BCC LSB
022534  000001                           1                               ;ONE SHIFT
022536  000000                  66$:     0                               ;DATA CHARACTER
022540  000000                  67$:     0                               ;OLD BCC
022542  103405                           BCS     68$                     ;BR IF SOFT BCC LSB IS SET
022544  004737  031072                   JSR     PC,GETQ0                ;GET HARDWARE TRANSMITTER BCC LSB
022550  103006                           BCC     69$                     ;BR IF HARD BCC LSB IS CLEAR
022552  104012                           ERROR   12                      ;ERROR, BCC LSB IS SET
022554  000404                           BR      69$                     ;CONTINUE
022556  004737  031072          68$:     JSR     PC,GETQ0                ;GET HARDWARE TRANSMITTER BCC LSB
022562  103401                           BCS     69$                     ;BR IF HARD BCC LSB IS SET
022564  104016                           ERROR   16                      ;ERROR, HARD BCC LSB IS CLEAR
022566                          69$:
022566  006037  022536                   ROR     66$                     ;SHIFT SOFT DATA
022572  013737  030756  022540           MOV     CALBCC,67$              ;LOAD OLD SOFT BCC
022600  022700  000010                   CMP     #10,R0                  ;DONE YET?
022604  001346                           BNE     65$                     ;BR IF NOT DONE
022606  104405                           SCOP1                           ;SCOPE SUBTEST (SW09=1)
022610  012737  022616  001444           MOV     #71$,LOCK               ;NEW SCOPE1
022616  004737  031320          71$:     JSR     PC,CLRIO                ;CLEAR BCC REGISTERS
022622  005000                           CLR     R0                      ;START SHIFT COUNTER AT ZERO
022624  012737  120001  030754           MOV     #CRC16,XPOLY            ;LOAD POLYNOMIAL FOR SOFTWARE BCC
022632  012737  000252  022672           MOV     #252,73$;               ;LOAD CHAR FOR SOFTWARE BCC
022640  005037  022674                   CLR     74$                     ;CLEAR OLD SOFTWARE BCC
022644  004737  030760                   JSR     PC,BCCLD                ;LOAD OUT SILO WITH 2 SYNCS
022650  000252                           252                             ;AND THE CHARACTER 252
022652  104413  000032                   DATACLK,        32              ;GET RECEIVER ACTIVE
022656  104413  000001          72$:     DATACLK,        1               ;SHIFT BCC ONCE
022662  005200                           INC     R0                      ;BUMP SHIFT COUNT
022664  004537  030650                   JSR     R5,SIMBCC               ;CALCULATE SOFTWARE BCC LSB
022670  000001                           1                               ;ONE SHIFT
022672  000000                  73$:     0                               ;DATA CHARACTER
022674  000000                  74$:     0                               ;OLD BCC
022676  103405                           BCS     75$                     ;BR IF SOFT BCC LSB IS SET
022700  004737  031104                   JSR     PC,GETQI                ;GET HARDWARE RECEIVER BCC LSB
022704  103006                           BCC     76$                     ;BR IF HARD BCC LSB IS CLEAR
022706  104013                           ERROR   13                      ;ERROR, BCC LSB IS SET
022710  000404                           BR      76$                     ;CONTINUE
022712  004737  031104          75$:     JSR     PC,GETQI                ;GET HARDWARE RECEIVER BCC LSB
022716  103401                           BCS     76$                     ;BR IF HARD BCC LSB IS SET
022720  104017                           ERROR   17                      ;ERROR, BCC LSB IS CLEAR
022722                          76$:
022722  006037  022672                   ROR     73$                     ;SHIFT SOFT DATA
022726  013737  030756  022674           MOV     CALBCC,74$              ;LOAD OLD SOFT BCC
022734  022700  000010                   CMP     #10,R0                  ;DONE YET?
022740  001346                           BNE     72$                     ;BR IF NOT DONE
022742  104405                           SCOP1                           ;SCOPE SUBTEST (SW09=1)
022744                          77$:
```

```

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

                                                     ;  TEST 46
                                                     ;  -------------
                                             ;;**********************************************************
022744  000004                               TST46:  SCOPE
022746  012737  000046  001202                       MOV     #46,$TSTNM                ; LOAD THE NO. OF THIS TEST
022754  012737  023202  001442                       MOV     #TST47,NEXT               ; POINT TO THE START OF NEXT TEST.
                                                                               ;R1 CONTAINS BASE KMC11 ADDRESS
022762  104410                                       MSTCLR                    ;MASTER CLEAR KMC11
022764  012711  004000                               MOV     #BIT11,(R1)       ;SET LINE UNIT LOOP
022770  005003                                       CLR     R3                ;ZERO BIT COUNT
022772  005004                                       CLR     R4                ;R4 CONTAINS CHAR TO BE LOADED IN SILO
022774  005005                                       CLR     R5                ;R5 CONTAINS CHAR CURRENTLY BEING SHIFTED OUT
022776  005037  023100                               CLR     4$                ;CLEAR SOFT BCC
023002  012737  120001  030754                       MOV     #CRC16,XPOLY      ;LOAD POLYNOMINAL
023010  004737  031122                               JSR     PC,SYNLD          ;LOAD SILO WITH 2 SYNCS, SOM SET
023014  010461  000004                               MOV     R4,4(R1)          ;PORT4+CHAR
023020  104412                                       ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023022  122110                                       122110                    ;LOAD OUT DATA
023024  005204                                       INC     R4                ;INCREMENT TO NEXT CHARACTER
023026  010461  000004                               MOV     R4,4(R1)          ;PORT4+CHAR
023032  104412                                       ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023034  122110                                       122110                    ;LOAD OUT DATA
023036  005204                                       INC     R4                ;INCREMENT TO NEXT CHARACTER
023040  010461  000004                               MOV     R4,4(R1)          ;PORT4+CHAR
023044  104412                                       ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023046  122110                                       122110                    ;LOAD OUT DATA
023050  004737  030006                               JSR     PC,OCOR           ;WAIT FOR OCOR
023054  104413  000021                               DATACLK,21                ;CLOCK DATA
023060  010537  023076                       1$:     MOV     R5,3$             ;LOAD CHAR FOR SOFT CRC
023064  104413  000001                       2$:     DATACLK,1                 ;SHIFT BCC ONCE
023070  004537  030650                               JSR     R5,SIMBCC         ;CALCULATE SOFT BCC
023074  000001                                       1                         ;SOFT SHIFT COUNT
023076  000000                               3$:     0                         ;SOFT CHARACTER
023100  000000                               4$:     0                         ;OLD SOFT BCC
023102  103405                                       BCS     5$                ;BR IF SOFT BCC LSB IS SET
023104  004737  031072                               JSR     PC,GETQ0                  ;GET HARDWARE TRANSMITTER BCC LSB
023110  103006                                       BCC     6$                ;BR IF OK (CLEARED)
023112  104020                                       ERROR   20                ;ERROR, BCC LSB WAS SET
023114  000404                                       BR      6$                ;CONTINUE WITH TEST
023116  004737  031072                       5$:     JSR     PC,GETQ0                  ;GET HARDWARE TRANSMITTER BCC LSB
023122  103401                                       BCS     6$                ;BR IF OK (SET)
023124  104021                                       ERROR   21                ;ERROR, BCC LSB WAS CLEAR

023126                                       6$:
023126  006037  023076                               ROR     3$                ;SHIFT SOFT DATA
023132  013737  030756  023100                       MOV     CALBCC,4$         ;LOAD OLD SOFT BCC
023140  005203                                       INC     R3                ;INCREMENT BIT COUNTER
023142  022703  000010                               CMP     #10,R3            ;DONE A FULL CHARACTER YET?
023146  001346                                       BNE     2$                ;BR IF NO
```

```
023150  005003                                CLR     R3              ;RESTART BIT COUNTER
023152  005204                                INC     R4              ;INCREMENT DATA FOR SILO
023154  022704  000400                        CMP     #400,R4         ;DONE BINARY COUNT YET?
023160  003404                                BLE     9$              ;BR IF YES
023162  010461  000004                        MOV     R4,4(R1)        ;PORT4+DATA
023166  104412                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023170  122110                                122110                  ;LOAD OUT DATA
023172  005205                        9$:     INC     R5              ;INCREMENT DATA
023174  022705  000400                        CMP     #400,R5         ;DONE BINARY PATTERN YET?
023200  001327                                BNE     1$              ;BR IF NO
023202                                7$:



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

                                              ;  TEST 47
                                              ;--------------

                                      ;;*************************************************************
023202  000004                        TST47:  SCOPE
023204  012737  000047  001202                MOV     #47,$TSTNM                  ; LOAD THE NO. OF THIS TEST
023212  012737  023440  001442                MOV     #TST50,NEXT                 ; POINT TO THE START OF NEXT TEST.
                                                                      ;R1 CONTAINS BASE KMC11 ADDRESS
023220  104410                                MSTCLR                  ;MASTER CLEAR KMC11
023222  012711  004000                        MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
023226  005003                                CLR     R3              ;ZERO BIT COUNT
023230  005004                                CLR     R4              ;R4 CONTAINS CHAR TO BE LOADED IN SILO
023232  005005                                CLR     R5              ;R5 CONTAINS CHAR CURRENTLY BEING SHIFTED OUT
023234  005037  023336                        CLR     4$              ;CLEAR SOFT BCC
023240  012737  120001  030754                MOV     #CRC16,XPOLY    ;LOAD POLYNOMINAL
023246  004737  031122                        JSR     PC,SYNLD        ;LOAD SILO WITH 2 SYNCS, SOM SET
023252  010461  000004                        MOV     R4,4(R1)        ;PORT4+CHAR
023256  104412                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023260  122110                                122110                  ;LOAD OUT DATA
023262  005204                                INC     R4              ;INCREMENT TO NEXT CHARACTER
023264  010461  000004                        MOV     R4,4(R1)        ;PORT4+CHAR
023270  104412                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023272  122110                                122110                  ;LOAD OUT DATA
023274  005204                                INC     R4              ;INCREMENT TO NEXT CHARACTER
023276  010461  000004                        MOV     R4,4(R1)        ;PORT4+CHAR
023302  104412                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023304  122110                                122110                  ;LOAD OUT DATA
023306  004737  030006                        JSR     PC,OCOR         ;WAIT FOR OCOR
023312  104413  000032                        DATACLK,32              ;CLOCK DATA
023316  010537  023334                1$:     MOV     R5,3$           ;LOAD CHAR FOR SOFT CRC
023322  104413  000001                2$:     DATACLK,1               ;SHIFT BCC ONCE
023326  004537  030650                        JSR     R5,SIMBCC       ;CALCULATE SOFT BCC
023332  000001                                1                       ;SOFT SHIFT COUNT
023334  000000                        3$:     0                       ;SOFT CHARACTER
023336  000000                        4$:     0                       ;OLD SOFT BCC
023340  103405                                BCS     5$              ;BR IF SOFT BCC LSB IS SET
023342  004737  031104                        JSR     PC,GETQI                ;GET HARDWARE RECEIVER BCC LSB
023346  103006                                BCC     6$              ;BR IF OK (CLEARED)
```

```
023350  104022                           ERROR   22             ;ERROR, BCC LSB WAS SET
023352  000404                           BR      6$             ;CONTINUE WITH TEST
023354  004737  031104           5$:     JSR     PC,GETQI               ;GET HARDWARE RECEIVER BCC LSB
023360  103401                           BCS     6$             ;BR IF OK (SET)
023362  104023                           ERROR   23             ;ERROR, BCC LSB WAS CLEAR

023364                           6$:
023364  006037  023334                   ROR     3$             ;SHIFT SOFT DATA
023370  013737  030756  023336           MOV     CALBCC,4$      ;LOAD OLD SOFT BCC
023376  005203                           INC     R3             ;INCREMENT BIT COUNTER
023400  022703  000010                   CMP     #10,R3         ;DONE A FULL CHARACTER YET?
023404  001346                           BNE     2$             ;BR IF NO
023406  005003                           CLR     R3             ;RESTART BIT COUNTER
023410  005204                           INC     R4             ;INCREMENT DATA FOR SILO
023412  022704  000400                   CMP     #400,R4        ;DONE BINARY COUNT YET?
023416  003404                           BLE     9$             ;BR IF YES
023420  010461  000004                   MOV     R4,4(R1)       ;PORT4+DATA
023424  104412                           ROMCLK                 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023426  122110                           122110                 ;LOAD OUT DATA
023430  005205                   9$:     INC     R5             ;INCREMENT DATA
023432  022705  000400                   CMP     #400,R5        ;DONE BINARY PATTERN YET?
023436  001327                           BNE     1$             ;BR IF NO
023440                           7$:


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

                                         ;   TEST 50
                                         ;--------------

                                 ;;**************************************************************
023440  000004                   TST50:  SCOPE
023442  012737  000050  001202           MOV     #50,$TSTNM             ; LOAD THE NO. OF THIS TEST
023450  012737  023772  001442           MOV     #TST51,NEXT            ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
023456  104410                           MSTCLR                 ;MASTER CLEAR KMC11

                                         ;LOAD OUT DATA SILO

023460  012711  004000                   MOV     #BIT11,(R1)    ;SET LINE UNIT LOOP
023464  012704  031540                   MOV     #MESDAT,R4     ;LOAD POINTER TO DATA
023470  005037  023564                   CLR     10$            ;CLEAR SOFT BCC
023474  012700  000004                   MOV     #4,R0          ;LOAD CHARACTER COUNT
023500  004737  031122                   JSR     PC,SYNLD       ;LOAD 2 SYNCS IN OUT SILO
023504  004737  030140                   JSR     PC,OUTRDY      ;WAIT FOR OUTRDY
023510  004537  031256                   JSR     R5,MESLD       ;LOAD SILO WITH 4 CHAR MESS
023514  031540                           MESDAT                 ;ADDRESS OF MESSAGE
023516  000004                           4                      ;NUMBER OF CHARACTERS
023520  004737  031232                   JSR     PC,EOM         ;LOAD GARBAGE CHARACTER, WITH EOM SET
023524  004737  030006                   JSR     PC,OCOR        ;WAIT FOR OCOR
023530  005003                           CLR     R3             ;CLEAR BIT COUNTER
```

```
023532  104413  000022                    DATACLK,22                  ;CLOCK DATA
023536  112405                    12$:    MOVB    (R4)+,R5            ;LOAD R5 WITH CHAR
023540  010502                            MOV     R5,R2               ;LOAD R2 WITH CHAR

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

023542  012737  120001  030754            MOV     #CRC16,XPOLY        ;LOAD POLYNOMIAL
023550  010537  023562                    MOV     R5,67$              ;LOAD SOFT CHAR FOR BCC
023554  004537  030650                    JSR     R5,SIMBCC           ;CALCULATE SOFT BCC
023560  000010                            10                          ;SHIFT COUNT
023562  000000                    67$:    0                           ;CHARACTER
023564  000000                    10$:    0                           ;OLD BCC
023566  013737  030756  023564            MOV     CALBCC,10$          ;LOAD SOFT BCC FOR NEXT SHIFT
023574  104413  000001            64$:    DATACLK,        1           ;SHIFT DATA IN TO BIT WINDOW
023600  106002                            RORB    R2                  ;SHIFT SOFT DATA
023602  103005                            BCC     65$                 ;BR IF A SPACE
023604  004737  027754                    JSR     PC,GETSI            ;LOOK AT BIT WINDOW
023610  103406                            BCS     66$                 ;BR IF OK (MARK)
023612  104006                            ERROR   6                   ;ERROR, BIT WINDOW WAS A SPACE
023614  000404                            BR      66$                 ;CONTINUE
023616  004737  027754            65$:    JSR     PC,GETSI            ;LOOK AT BIT WINDOW
023622  103001                            BCC     66$                 ;BR IF OK (SPACE)
023624  104006                            ERROR   6                   ;ERROR, BIT WINDOW WAS A MARK
023626                            66$:
023626  005203                            INC     R3                  ;BUMP BIT COUNTER
023630  022703  000010                    CMP     #10,R3              ;DONE FULL 8 BITS YET
023634  001357                            BNE     64$                 ;BR IF NO
023636  005003                            CLR     R3                  ;CLEAR BIT COUNTER
023640  005300                            DEC     R0                  ;DEC CHARACTER COUNT
023642  001335                            BNE     12$                 ;BR IF NOT DONE YET

                                          ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

023644  013700  030756                    MOV     CALBCC,R0           ;PUT BCC IN R0
023650  104413  000001            68$:    DATACLK,1                   ;SHIFT HARDWARE BCC
023654  006000                            ROR     R0                  ;SHIFT SOFT BCC
023656  103005                            BCC     69$                 ;BR IF CARRY CLEAR
023660  004737  027754                    JSR     PC,GETSI            ;LOOK AT BIT WINDOW
023664  103406                            BCS     70$                 ;BR IF OK (MARK)
023666  104014                            ERROR   14                  ;ERROR, CRC WRONG (SPACE)
023670  000404                            BR      70$                 ;CONTINUE
023672  004737  027754            69$:    JSR     PC,GETSI            ;LOOK AT BIT WINDOW
023676  103001                            BCC     70$                 ;BR IF OK (SPACE)
023700  104014                            ERROR   14                  ;ERROR, CRC WRONG (MARK)
023702                            70$:
023702  005203                            INC     R3                  ;BUMP BIT COUNTER
023704  022703  000020                    CMP     #20,R3              ;FINISHED BCC YET?
023710  001357                            BNE     68$                 ;BR IF NO
023712  005003                            CLR     R3                  ;CLEAR BIT COUNTER

                                          ;CHECK TO SEE IF TRANSMITTER IS MARKING

023714  104413  000001            2$:     DATACLK,        1           ;CLOCK TRANSMITTER
023720  004737  027754                    JSR     PC,GETSI            ;LOOK AT WINDOW
023724  103401                            BCS     3$                  ;IT SHOULD BE MARKING
```

K08

```
023726  104024                                   ERROR   24              ;ERROR, BIT HAS A SPACE
023730  005203                           3$:     INC     R3              ;BUMP BIT COUNTER
023732  022703  000007                           CMP     #7,R3           ;DONE YET
023736  001366                                   BNE     2$              ;BR IF NO
023740  104413  000010                           DATACLK,         10     ;GIVE ENOUGH TICKS TO CLEAR OUT ACTIVE
023744  005003                                   CLR     R3              ;CLEAR BIT COUNTER
023746  104413  000001                   4$:     DATACLK,         1      ;SHIFT OUT NEXT BIT
023752  004737  027754                           JSR     PC,GETSI        ;LOOK AT BIT WINDOW
023756  103401                                   BCS     .+4             ;BR IF IT IS A MARK
023760  104024                                   ERROR   24              ;ERROR, TRANSMITTER IS NOT MARKING
023762  005203                                   INC     R3              ;INC BIT COUNT
023764  022703  000020                           CMP     #20,R3          ;DONE YET?
023770  001366                                   BNE     4$              ;BR IF NO
023772                                   5$:



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

                                                 ;  TEST 51
                                                 ;--------------

                                         ;;*****************************************************************
023772  000004                           TST51:  SCOPE
023774  012737  000051  001202                   MOV     #51,$TSTNM                ; LOAD THE NO. OF THIS TEST
024002  012737  024174  001442                   MOV     #TST52,NEXT               ; POINT TO THE START OF NEXT TEST.
                                                                         ;R1 CONTAINS BASE KMC11 ADDRESS
024010  104410                                   MSTCLR                  ;MASTER CLEAR KMC11
024012  012711  004000                           MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
024016  012702  031540                           MOV     #MESDAT,R2      ;LOAD POINTER TO DATA
024022  012700  000004                           MOV     #4,R0           ;LOAD CHARACTER COUNT
024026  004737  031122                           JSR     PC,SYNLD        ;LOAD 2 SYNCS IN OUT SILO
024032  004737  030140                           JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
024036  004537  031256                           JSR     R5,MESLD        ;LOAD SILO WITH 4 CHAR MESS
024042  031540                                   MESDAT                  ;ADDRESS OF MESSAGE
024044  000004                                   4                       ;NUMBER OF CHARACTERS
024046  004737  031232                           JSR     PC,EOM          ;LOAD GARBAGE CHARACTER, WITH EOM SET
024052  004737  030006                           JSR     PC,OCOR         ;WAIT FOR OCOR
024056  104413  000114                           DATACLK,114             ;CLOCK DATA
024062  004737  030614                   3$:     JSR     PC,INRDY        ;WAIT FOR INRDY
024066  104412                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024070  021204                                   021204                  ;GET IN DATA
024072  016104  000004                           MOV     4(R1),R4        ;PUT "FOUND" IN R4
024076  112205                                   MOVB    (R2)+,R5        ;PUT "EXPECTED" IN R5
024100  120504                                   CMPB    R5,R4           ;COMPARE RECEIVED DATA
024102  001401                                   BEQ     1$              ;BR IF OK
024104  104010                                   ERROR   10              ;DATA ERROR
024106  005300                           1$:     DEC     R0              ;DEC CHARACTER COUNT
024110  001364                                   BNE     3$              ;BR IF NOT DONE YET

                                                 ;CHECK TO SEE THAT IN BCC MATCH IS SET

024112  004737  030614                           JSR     PC,INRDY        ;WAIT FOR INRDY
```

```
024116  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024120  021204                                  021204                          ;GET FIRST HALF OF CRC
024122  116137  000004  001302                  MOVB    4(R1),$TMP2             ;PUT IN $TMP2
024130  042737  177400  001302                  BIC     #177400,$TMP2           ;CLEAR HI BYTE
024136  004737  030614                          JSR     PC,INRDY                ;WAIT FOR INRDY
024142  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024144  021244                                  021244
024146  016104  000004                          MOV     4(R1),R4                ;PUT "FOUND" IN R4
024152  042704  000376                          BIC     #376,R4                 ;CLEAR UNWANTED BITS
024156  012705  000001                          MOV     #1,R5                   ;PUT "EXPECTED" IN R5
024162  120504                                  CMPB    R5,R4                   ;IS IN BCC MATCH SET?
024164  001401                                  BEQ     25$
024166  104015                                  ERROR   15                      ;IN BCC MATCH ERROR
024170                                  25$:
024170  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024172  021204                                  021204                          ;GET LAST HALF
024174                                  2$:



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

                                                ;       TEST 52
                                                ;_______________

                                        ;;*******************************************************************
024174  000004                          TST52:  SCOPE
024176  012737  000052  001202                  MOV     #52,$TSTNM                      ; LOAD THE NO. OF THIS TEST
024204  012737  025274  001442                  MOV     #TST53,NEXT                     ; POINT TO THE START OF NEXT TEST.
                                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
024212  104410                                  MSTCLR                          ;MASTER CLEAR KMC11

                                                ;LOAD OUT DATA SILO

024214  012711  004000                          MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
024220  012704  031540                          MOV     #MESDAT,R4              ;LOAD POINTER TO DATA
024224  005037  024334                          CLR     10$                     ;CLEAR SOFT BCC
024230  012700  000004                          MOV     #4,R0                   ;LOAD CHARACTER COUNT
024234  004737  031122                          JSR     PC,SYNLD                ;LOAD 2 SYNCS IN OUT SILO
024240  004737  030140                          JSR     PC,OUTRDY               ;WAIT FOR OUTRDY
024244  004537  031256                          JSR     R5,MESLD                ;LOAD SILO WITH 4 CHAR MESS
024250  031540                                  MESDAT                          ;ADDRESS OF MESSAGE
024252  000004                                  4                               ;NUMBER OF CHARACTERS
024254  004737  031232                          JSR     PC,EOM                  ;LOAD GARBAGE CHARACTER, WITH EOM SET
024260  004537  031256                          JSR     R5,MESLD                ;LOAD FOUR MORE CHARACTERS
024264  031540                                  MESDAT                          ;ADDRESS OF MESSAGE
024266  000004                                  4                               ;NUMBER OF CHACTERS
024270  004737  031232                          JSR     PC,EOM                  ;SET EOM
024274  004737  030006                          JSR     PC,OCOR                 ;WAIT FOR OCOR
```

```
024300  005003                                  CLR     R3              ;CLEAR BIT COUNTER
024302  104413  000022                          DATACLK,22              ;CLOCK DATA
024306  112405                          12$:    MOVB    (R4)+,R5        ;LOAD R5 WITH CHAR
024310  010502                                  MOV     R5,R2           ;LOAD R2 WITH CHAR

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

024312  012737  120001  030754                  MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL
024320  010537  024332                          MOV     R5,67$          ;LOAD SOFT CHAR FOR BCC
024324  004537  030650                          JSR     R5,SIMBCC       ;CALCULATE SOFT BCC
024330  000010                                  10                      ;SHIFT COUNT
024332  000000                          67$:    0                       ;CHARACTER
024334  000000                          10$:    0                       ;OLD BCC
024336  013737  030756  024334                  MOV     CALBCC,10$      ;LOAD SOFT BCC FOR NEXT SHIFT
024344  104413  000001                  64$:    DATACLK,        1       ;SHIFT DATA IN TO BIT WINDOW
024350  106002                                  RORB    R2              ;SHIFT SOFT DATA
024352  103005                                  BCC     65$             ;BR IF A SPACE
024354  004737  027754                          JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024360  103406                                  BCS     66$             ;BR IF OK (MARK)
024362  104006                                  ERROR   6               ;ERROR, BIT WINDOW WAS A SPACE
024364  000404                                  BR      66$             ;CONTINUE
024366  004737  027754                  65$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024372  103001                                  BCC     66$             ;BR IF OK (SPACE)
024374  104006                                  ERROR   6               ;ERROR, BIT WINDOW WAS A MARK
024376                                  66$:
024376  005203                                  INC     R3              ;BUMP BIT COUNTER
024400  022703  000010                          CMP     #10,R3          ;DONE FULL 8 BITS YET
024404  001357                                  BNE     64$             ;BR IF NO
024406  005003                                  CLR     R3              ;CLEAR BIT COUNTER
024410  005300                                  DEC     R0              ;DEC CHARACTER COUNT
024412  001335                                  BNE     12$             ;BR IF NOT DONE YET

                                                ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

024414  013700  030756                          MOV     CALBCC,R0       ;PUT BCC IN R0
024420  104413  000001                  68$:    DATACLK,1               ;SHIFT HARDWARE BCC
024424  006000                                  ROR     R0              ;SHIFT SOFT BCC
024426  103005                                  BCC     69$             ;BR IF CARRY CLEAR
024430  004737  027754                          JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024434  103406                                  BCS     70$             ;BR IF OK (MARK)
024436  104014                                  ERROR   14              ;ERROR, CRC WRONG (SPACE)
024440  000404                                  BR      70$             ;CONTINUE
024442  004737  027754                  69$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024446  103001                                  BCC     70$             ;BR IF OK (SPACE)
024450  104014                                  ERROR   14              ;ERROR, CRC WRONG (MARK)
024452                                  70$:
024452  005203                                  INC     R3              ;BUMP BIT COUNTER
024454  022703  000020                          CMP     #20,R3          ;FINISHED BCC YET?
024460  001357                                  BNE     68$             ;BR IF NO
024462  005003                                  CLR     R3              ;CLEAR BIT COUNTER
024464  012700  000004                          MOV     #4,R0           ;RESET CHARACTER COUNTER
024470  012704  031540                          MOV     #MESDAT,R4      ;LOAD MESSAGE POINTER
024474  005037  024526                          CLR     11$             ;CLR SOFT BCC
024500  112405                          13$:    MOVB    (R4)+,R5        ;LOAD CHAR IN R5
024502  010502                                  MOV     R5,R2           ;LOAD CHAR IN R2
```

```

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

024504  012737  120001  030754          MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL
024512  010537  024524                  MOV     R5,76$          ;LOAD SOFT CHAR FOR BCC
024516  004537  030650                  JSR     R5,SIMBCC       ;CALCULATE SOFT BCC
024522  000010                          10                      ;SHIFT COUNT
024524  000000                  76$:    0                       ;CHARACTER
024526  000000                  11$:    0                       ;OLD BCC
024530  013737  030756  024526          MOV     CALBCC,11$      ;LOAD SOFT BCC FOR NEXT SHIFT
024536  104413  000001          73$:    DATACLK,        1       ;SHIFT DATA IN TO BIT WINDOW
024542  106002                          RORB    R2              ;SHIFT SOFT DATA
024544  103005                          BCC     74$             ;BR IF A SPACE
024546  004737  027754                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024552  103406                          BCS     75$             ;BR IF OK (MARK)
024554  104006                          ERROR   6               ;ERROR, BIT WINDOW WAS A SPACE
024556  000404                          BR      75$             ;CONTINUE
024560  004737  027754          74$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024564  103001                          BCC     75$             ;BR IF OK (SPACE)
024566  104006                          ERROR   6               ;ERROR, BIT WINDOW WAS A MARK
024570                          75$:
024570  005203                          INC     R3              ;BUMP BIT COUNTER
024572  022703  000010                  CMP     #10,R3          ;DONE FULL 8 BITS YET
024576  001357                          BNE     73$             ;BR IF NO
024600  005003                          CLR     R3              ;CLEAR BIT COUNTER
024602  005300                          DEC     R0              ;DEC CHARACTER COUNT
024604  001335                          BNE     13$             ;BR IF NOT DONE YET

                                                ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

024606  013700  030756                  MOV     CALBCC,R0       ;PUT BCC IN R0
024612  104413  000001          77$:    DATACLK,1               ;SHIFT HARDWARE BCC
024616  006000                          ROR     R0              ;SHIFT SOFT BCC
024620  103005                          BCC     78$             ;BR IF CARRY CLEAR
024622  004737  027754                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024626  103406                          BCS     79$             ;BR IF OK (MARK)
024630  104014                          ERROR   14              ;ERROR, CRC WRONG (SPACE)
024632  000404                          BR      79$             ;CONTINUE
024634  004737  027754          78$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024640  103001                          BCC     79$             ;BR IF OK (SPACE)
024642  104014                          ERROR   14              ;ERROR, CRC WRONG (MARK)
024644                          79$:
024644  005203                          INC     R3              ;BUMP BIT COUNTER
024646  022703  000020                  CMP     #20,R3          ;FINISHED BCC YET?
024652  001357                          BNE     77$             ;BR IF NO
024654  005003                          CLR     R3              ;CLEAR BIT COUNTER

                                                ;CHECK TO SEE IF TRANSMITTER IS MARKING

024656  104413  000001          2$:     DATACLK,        1       ;CLOCK TRANSMITTER
024662  004737  027754                  JSR     PC,GETSI        ;LOOK AT WINDOW
024666  103401                          BCS     3$              ;IT SHOULD BE MARKING
024670  104024                          ERROR   24              ;ERROR, BIT WAS A SPACE
024672  005203                  3$:     INC     R3              ;BUMP BIT COUNTER
024674  022703  000007                  CMP     #7,R3           ;DONE YET
024700  001366                          BNE     2$              ;BR IF NO
```

B09

```
024702  104413  000010                  DATACLK,        10      ;GIVE ENOUGH TICKS TO CLEAR OUT ACTIVE
024706  005003                          CLR     R3              ;CLEAR BIT COUNTER
024710  104413  000001          4$:     DATACLK,        1       ;SHIFT OUT NEXT BIT
024714  004737  027754                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
024720  103401                          BCS     .+4             ;BR IF IT IS A MARK
024722  104024                          ERROR   24              ;ERROR, TRANSMITTER IS NOT MARKING
024724  005203                          INC     R3              ;INC BIT COUNT
024726  022703  000020                  CMP     #20,R3          ;DONE YET?
024732  001366                          BNE     4$              ;BR IF NO

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

024734  104413  000001                  DATACLK,        1       ;GET LAST BIT IN RECEIVER
024740  012703  000004                  MOV     #4,R3           ;R3=CHARACTER COUNT
024744  012702  031540                  MOV     #MESDAT,R2      ;LOAD MESSAGE POINTER IN R2
024750  004737  030614          40$:    JSR     PC,INRDY        ;WAIT FOR INRDY
024754  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024756  021204                          021204
024760  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
024764  112205                          MOVB    (R2)+,R5        ;PUT "EXPECTED" IN R5
024766  120504                          CMPB    R5,R4           ;IS RECEIVED DATA CORRECT?
024770  001401                          BEQ     41$             ;BR IF YES
024772  104010                          ERROR   10              ;RECEIVE DATA ERROR
024774  005303                  41$:    DEC     R3              ;DEC CHARACTER COUNT
024776  001364                          BNE     40$             ;BR IF NOT DONE YET

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

025000  004737  030614                  JSR     PC,INRDY        ;WAIT FOR INRDY
025004  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025006  021204                          021204                  ;GET FIRST HALF OF CRC
025010  116137  000004  001302          MOVB    4(R1),$TMP2     ;PUT IN $TMP2
025016  042737  177400  001302          BIC     #177400,$TMP2   ;CLEAR HI BYTE
025024  004737  030614                  JSR     PC,INRDY        ;WAIT FOR INRDY
025030  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025032  021244                          021244
025034  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
025040  042704  000376                  BIC     #376,R4         ;CLEAR UNWANTED BITS
025044  012705  000001                  MOV     #1,R5           ;PUT "EXPECTED" IN R5
025050  120504                          CMPB    R5,R4           ;IS IN BCC MATCH SET?
025052  001401                          BEQ     50$
025054  104015                          ERROR   15              ;IN BCC MATCH ERROR
025056                          50$:
025056  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025060  021204                          021204                  ;GET LAST HALF
025062  116137  000004  001301          MOVB    4(R1),$TMP1+1   ;PUT IN $TMP1
025070  042737  000377  001300          BIC     #377,$TMP1      ;CLEAR LO BYTE
025076  053737  001300  001302          BIS     $TMP1,$TMP2     ;16 BIT BCC NOW IN $TMP2
025104  023737  030756  001302          CMP     CALBCC,$TMP2    ;IS IT CORRECT?
025112  001401                          BEQ     42$             ;BR IF OK
025114  104027                          ERROR   27

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

```

025116  012703  000004          42$:    MOV     #4,R3           ;R3=CHARACTER COUNT
025122  012702  031540                  MOV     #MESDAT,R2      ;LOAD MESSAGE POINTER IN R2
025126  004737  030614          43$:    JSR     PC,INRDY        ;WAIT FOR INRDY
025132  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025134  021204                          021204
025136  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
025142  112205                          MOVB    (R2)+,R5        ;PUT "EXPECTED" IN R5
025144  120504                          CMPB    R5,R4           ;IS RECEIVED DATA CORRECT?
025146  001401                          BEQ     44$             ;BR IF YES
025150  104010                          ERROR   10              ;RECEIVE DATA ERROR
025152  005303                  44$:    DEC     R3              ;DEC CHARACTER COUNT
025154  001364                          BNE     43$             ;BR IF NOT DONE YET

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

025156  004737  030614                  JSR     PC,INRDY        ;WAIT FOR INRDY
025162  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025164  021204                          021204                  ;GET FIRST HALF OF CRC
025166  116137  000004  001302          MOVB    4(R1),$TMP2     ;PUT IN $TMP2
025174  042737  177400  001302          BIC     #177400,$TMP2   ;CLEAR HI BYTE
025202  004737  030614                  JSR     PC,INRDY        ;WAIT FOR INRDY
025206  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025210  021244                          021244
025212  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
025216  042704  000376                  BIC     #376,R4         ;CLEAR UNWANTED BITS
025222  012705  000001                  MOV     #1,R5           ;PUT "EXPECTED" IN R5
025226  120504                          CMPB    R5,R4           ;IS IN BCC MATCH SET?
025230  001401                          BEQ     51$
025232  104015                          ERROR   15              ;IN BCC MATCH ERROR
025234                          51$:
025234  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025236  021204                          021204                  ;GET LAST HALF
025240  116137  000004  001301          MOVB    4(R1),$TMP1+1   ;PUT IN $TMP1
025246  042737  000377  001300          BIC     #377,$TMP1      ;CLEAR LO BYTE
025254  053737  001300  001302          BIS     $TMP1,$TMP2     ;16 BIT BCC NOW IN $TMP2
025262  023737  030756  001302          CMP     CALBCC,$TMP2    ;IS IT CORRECT?
025270  001401                          BEQ     5$              ;BR IF OK
025272  104027                          ERROR   27
025274                          5$:


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


                                        ;  TEST 53
```

```
                                                 ;---------------
                                        ;;*************************************************************
        025274  000004                  TST53:   SCOPE
        025276  012737  000053  001202           MOV     #53,$TSTNM                   ; LOAD THE NO. OF THIS TEST
        025304  012737  026474  001442           MOV     #TST54,NEXT                  ; POINT TO THE START OF NEXT TEST.
                                                                            ;R1 CONTAINS BASE KMC11 ADDRESS
        025312  104410                           MSTCLR                     ;MASTER CLEAR KMC11

                                                 ;LOAD OUT DATA SILO

        025314  012711  004000                   MOV     #BIT11,(R1)        ;SET LINE UNIT LOOP
        025320  012704  031540                   MOV     #MESDAT,R4         ;LOAD POINTER TO DATA
        025324  005037  025440                   CLR     10$                ;CLEAR SOFT BCC
        025330  012700  000004                   MOV     #4,R0              ;LOAD CHARACTER COUNT
        025334  004737  031122                   JSR     PC,SYNLD           ;LOAD 2 SYNCS IN OUT SILO
        025340  004737  030140                   JSR     PC,OUTRDY          ;WAIT FOR OUTRDY
        025344  004537  031256                   JSR     R5,MESLD           ;LOAD SILO WITH 4 CHAR MESS
        025350  031540                           MESDAT                     ;ADDRESS OF MESSAGE
        025352  000004                           4                          ;NUMBER OF CHARACTERS
        025354  004737  031232                   JSR     PC,EOM             ;LOAD GARBAGE CHARACTER, WITH EOM SET
        025360  004737  031202                   JSR     PC,SOM             ;LOAD GARBAGE CHAR WITH SOM SET
        025364  004537  031256                   JSR     R5,MESLD           ;LOAD FOUR MORE CHARACTERS
        025370  031540                           MESDAT                     ;ADDRESS OF MESSAGE
        025372  000004                           4                          ;NUMBER OF CHACTERS
        025374  004737  031232                   JSR     PC,EOM             ;SET EOM
        025400  004737  030006                   JSR     PC,OCOR            ;WAIT FOR OCOR
        025404  005003                           CLR     R3                 ;CLEAR BIT COUNTER
        025406  104413  000022                   DATACLK,22                 ;CLOCK DATA
        025412  112405                  12$:     MOVB    (R4)+,R5           ;LOAD R5 WITH CHAR
        025414  010502                           MOV     R5,R2              ;LOAD R2 WITH CHAR

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

        025416  012737  120001  030754           MOV     #CRC16,XPOLY       ;LOAD POLYNOMIAL
        025424  010537  025436                   MOV     R5,67$             ;LOAD SOFT CHAR FOR BCC
        025430  004537  030650                   JSR     R5,SIMBCC          ;CALCULATE SOFT BCC
        025434  000010                           10                         ;SHIFT COUNT
        025436  000000                  67$:     0                          ;CHARACTER
        025440  000000                  10$:     0                          ;OLD BCC
        025442  013737  030756  025440           MOV     CALBCC,10$         ;LOAD SOFT BCC FOR NEXT SHIFT
        025450  104413  000001          64$:     DATACLK,        1          ;SHIFT DATA IN TO BIT WINDOW
        025454  106002                           RORB    R2                 ;SHIFT SOFT DATA
        025456  103005                           BCC     65$                ;BR IF A SPACE
        025460  004737  027754                   JSR     PC,GETSI           ;LOOK AT BIT WINDOW
        025464  103406                           BCS     66$                ;BR IF OK (MARK)
        025466  104006                           ERROR   6                  ;ERROR, BIT WINDOW WAS A SPACE
        025470  000404                           BR      66$                ;CONTINUE
        025472  004737  027754          65$:     JSR     PC,GETSI           ;LOOK AT BIT WINDOW
        025476  103001                           BCC     66$                ;BR IF OK (SPACE)
        025500  104006                           ERROR   6                  ;ERROR, BIT WINDOW WAS A MARK
        025502                          66$:
        025502  005203                           INC     R3                 ;BUMP BIT COUNTER
        025504  022703  000010                   CMP     #10,R3             ;DONE FULL 8 BITS YET
        025510  001357                           BNE     64$                ;BR IF NO
        025512  005003                           CLR     R3                 ;CLEAR BIT COUNTER
```

```
025514  005300                          DEC     R0              ;DEC CHARACTER COUNT
025516  001335                          BNE     12$             ;BR IF NOT DONE YET

                                ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

025520  013700  030756                  MOV     CALBCC,R0       ;PUT BCC IN R0
025524  104413  000001          68$:    DATACLK,1               ;SHIFT HARDWARE BCC
025530  006000                          ROR     R0              ;SHIFT SOFT BCC
025532  103005                          BCC     69$             ;BR IF CARRY CLEAR
025534  004737  027754                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
025540  103406                          BCS     70$             ;BR IF OK (MARK)
025542  104014                          ERROR   14              ;ERROR, CRC WRONG (SPACE)
025544  000404                          BR      70$             ;CONTINUE
025546  004737  027754          69$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
025552  103001                          BCC     70$             ;BR IF OK (SPACE)
025554  104014                          ERROR   14              ;ERROR, CRC WRONG (MARK)
025556                          70$:
025556  005203                          INC     R3              ;BUMP BIT COUNTER
025560  022703  000020                  CMP     #20,R3          ;FINISHED BCC YET?
025564  001357                          BNE     68$             ;BR IF NO
025566  005003                          CLR     R3              ;CLEAR BIT COUNTER

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

025570  005005                          CLR     R5              ;CHARACTER LOADED WITH SOM
025572  010502                          MOV     R5,R2           ;LOAD R2 WITH CHAR
025574  104413  000001          32$:    DATACLK,        1       ;CLOCK TRANSMITTER
025600  106002                          RORB    R2              ;SHIFT SOFT DATA
025602  103005                          BCC     30$             ;BR IF SPACE
025604  004737  027754                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
025610  103406                          BCS     31$             ;BR IF OK (MARK)
025612  104006                          ERROR   6               ;ERROR,BIT WINDOW WAS A SPACE
025614  000404                          BR      31$             ;CONTINUE
025616  004737  027754          30$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
025622  103001                          BCC     31$             ;BR IF OK (SPACE)
025624  104006                          ERROR   6               ;ERROR,BIT WINDOW WAS A MARK
025626  005203                  31$:    INC     R3              ;BUMP BIT COUNTER
025630  022703  000010                  CMP     #10,R3          ;DONE CHARACTER YET?
025634  001357                          BNE     32$             ;BR IF NO
025636  005003                          CLR     R3              ;RESET BIT COUNTER
025640  012700  000004                  MOV     #4,R0           ;RESET CHARACTER COUNTER
025644  012704  031540                  MOV     #MESDAT,R4      ;LOAD MESSAGE POINTER
025650  005037  025702                  CLR     11$             ;CLR SOFT BCC
025654  112405                  13$:    MOVB    (R4)+,R5        ;LOAD CHAR IN R5
025656  010502                          MOV     R5,R2           ;LOAD CHAR IN R2

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

025660  012737  120001  030754          MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL
025666  010537  025700                  MOV     R5,76$          ;LOAD SOFT CHAR FOR BCC
025672  004537  030650                  JSR     R5,SIMBCC       ;CALCULATE SOFT BCC
025676  000010                          10                      ;SHIFT COUNT
025700  000000                  76$:    0                       ;CHARACTER
025702  000000                  11$:    0                       ;OLD BCC
025704  013737  030756  025702          MOV     CALBCC,11$      ;LOAD SOFT BCC FOR NEXT SHIFT
025712  104413  000001          73$:    DATACLK,        1       ;SHIFT DATA IN TO BIT WINDOW
```

```
025716  106002                          RORB    R2              ;SHIFT SOFT DATA
025720  103005                          BCC     74$             ;BR IF A SPACE
025722  004737  027754                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
025726  103406                          BCS     75$             ;BR IF OK (MARK)
025730  104006                          ERROR   6               ;ERROR, BIT WINDOW WAS A SPACE
025732  000404                          BR      75$             ;CONTINUE
025734  004737  027754          74$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
025740  103001                          BCC     75$             ;BR IF OK (SPACE)
025742  104006                          ERROR   6               ;ERROR, BIT WINDOW WAS A MARK
025744                          75$:
025744  005203                          INC     R3              ;BUMP BIT COUNTER
025746  022703  000010                  CMP     #10,R3          ;DONE FULL 8 BITS YET
025752  001357                          BNE     73$             ;BR IF NO
025754  005003                          CLR     R3              ;CLEAR BIT COUNTER
025756  005300                          DEC     R0              ;DEC CHARACTER COUNT
025760  001335                          BNE     13$             ;BR IF NOT DONE YET

                                        ;CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW

025762  013700  030756                  MOV     CALBCC,R0       ;PUT BCC IN R0
025766  104413  000001          77$:    DATACLK,1               ;SHIFT HARDWARE BCC
025772  006000                          ROR     R0              ;SHIFT SOFT BCC
025774  103005                          BCC     78$             ;BR IF CARRY CLEAR
025776  004737  027754                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
026002  103406                          BCS     79$             ;BR IF OK (MARK)
026004  104014                          ERROR   14              ;ERROR, CRC WRONG (SPACE)
026006  000404                          BR      79$             ;CONTINUE
026010  004737  027754          78$:    JSR     PC,GETSI        ;LOOK AT BIT WINDOW
026014  103001                          BCC     79$             ;BR IF OK (SPACE)
026016  104014                          ERROR   14              ;ERROR, CRC WRONG (MARK)
026020                          79$:
026020  005203                          INC     R3              ;BUMP BIT COUNTER
026022  022703  000020                  CMP     #20,R3          ;FINISHED BCC YET?
026026  001357                          BNE     77$             ;BR IF NO
026030  005003                          CLR     R3              ;CLEAR BIT COUNTER

                                        ;CHECK TO SEE IF TRANSMITTER IS MARKING

026032  104413  000001          2$:     DATACLK,        1       ;CLOCK TRANSMITTER
026036  004737  027754                  JSR     PC,GETSI        ;LOOK AT WINDOW
026042  103401                          BCS     3$              ;IT SHOULD BE MARKING
026044  104024                          ERROR   24              ;ERROR, BIT WAS A SPACE
026046  005203                  3$:     INC     R3              ;BUMP BIT COUNTER
026050  022703  000007                  CMP     #7,R3           ;DONE YET
026054  001366                          BNE     2$              ;BR IF NO
026056  104413  000010                  DATACLK,        10      ;GIVE ENOUGH TICKS TO CLEAR OUT ACTIVE
026062  005003                          CLR     R3              ;CLEAR BIT COUNTER
026064  104413  000001          4$:     DATACLK,        1       ;SHIFT OUT NEXT BIT
026070  004737  027754                  JSR     PC,GETSI        ;LOOK AT BIT WINDOW
026074  103401                          BCS     .+4             ;BR IF IT IS A MARK
026076  104024                          ERROR   24              ;ERROR, TRANSMITTER IS NOT MARKING
026100  005203                          INC     R3              ;INC BIT COUNT
026102  022703  000020                  CMP     #20,R3          ;DONE YET?
026106  001366                          BNE     4$              ;BR IF NO

                                        ;CHECK TO SEE THAT FIRST FOUR CHARACTER MESSAGE
```

```
                                                ;WAS RECEIVED CORRECTLY (0,125,252,377)

026110  104413  000001                          DATACLK,        1       ;GET LAST BIT IN RECEIVER
026114  012703  000004                          MOV     #4,R3           ;R3=CHARACTER COUNT
026120  012702  031540                          MOV     #MESDAT,R2      ;LOAD MESSAGE POINTER IN R2
026124  004737  030614                  40$:    JSR     PC,INRDY        ;WAIT FOR INRDY
026130  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026132  021204                                  021204
026134  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
026140  112205                                  MOVB    (R2)+,R5        ;PUT "EXPECTED" IN R5
026142  120504                                  CMPB    R5,R4           ;IS RECEIVED DATA CORRECT?
026144  001401                                  BEQ     41$             ;BR IF YES
026146  104010                                  ERROR   10              ;RECEIVE DATA ERROR
026150  005303                          41$:    DEC     R3              ;DEC CHARACTER COUNT
026152  001364                                  BNE     40$             ;BR IF NOT DONE YET

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

026154  004737  030614                          JSR     PC,INRDY        ;WAIT FOR INRDY
026160  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026162  021204                                  021204                  ;GET FIRST HALF OF CRC
026164  116137  000004  001302                  MOVB    4(R1),$TMP2     ;PUT IN $TMP2
026172  042737  177400  001302                  BIC     #177400,$TMP2   ;CLEAR HI BYTE
026200  004737  030614                          JSR     PC,INRDY        ;WAIT FOR INRDY
026204  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026206  021244                                  021244
026210  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
026214  042704  000376                          BIC     #376,R4         ;CLEAR UNWANTED BITS
026220  012705  000001                          MOV     #1,R5           ;PUT "EXPECTED" IN R5
026224  120504                                  CMPB    R5,R4           ;IS IN BCC MATCH SET?
026226  001401                                  BEQ     50$
026230  104015                                  ERROR   15              ;IN BCC MATCH ERROR
026232                                  50$:
026232  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026234  021204                                  021204                  ;GET LAST HALF
026236  116137  000004  001301                  MOVB    4(R1),$TMP1+1   ;PUT IN $TMP1
026244  042737  000377  001300                  BIC     #377,$TMP1      ;CLEAR LO BYTE
026252  053737  001300  001302                  BIS     $TMP1,$TMP2     ;16 BIT BCC NOW IN $TMP2
026260  023737  030756  001302                  CMP     CALBCC,$TMP2    ;IS IT CORRECT?
026266  001401                                  BEQ     45$             ;BR IF OK
026270  104027                                  ERROR   27

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

026272  004737  030614                  45$:    JSR     PC,INRDY        ;WAIT FOR INRDY
026276  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026300  021204                                  021204                  ;GET RECEIVE DATA
026302  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
026306  005005                                  CLR     R5              ;PUT "EXPECTED" IN R5
026310  120504                                  CMPB    R5,R4           ;IS RECEVIED DATA CORRECT?
026312  001401                                  BEQ     42$             ;BR IF YES
026314  104010                                  ERROR   10              ;RECEIVE DATA ERROR

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

```

026316  012703  000004          42$:    MOV     #4,R3                   ;R3=CHARACTER COUNT
026322  012702  031540                  MOV     #MESDAT,R2              ;LOAD MESSAGE POINTER IN R2
026326  004737  030614          43$:    JSR     PC,INRDY                ;WAIT FOR INRDY
026332  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026334  021204                          021204
026336  016104  000004                  MOV     4(R1),R4                ;PUT "FOUND" IN R4
026342  112205                          MOVB    (R2)+,R5                ;PUT "EXPECTED" IN R5
026344  120504                          CMPB    R5,R4                   ;IS RECEIVED DATA CORRECT?
026346  001401                          BEQ     44$                     ;BR IF YES
026350  104010                          ERROR   10                      ;RECEIVE DATA ERROR
026352  005303                  44$:    DEC     R3                      ;DEC CHARACTER COUNT
026354  001364                          BNE     43$                     ;BR IF NOT DONE YET

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

026356  004737  030614                  JSR     PC,INRDY                ;WAIT FOR INRDY
026362  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026364  021204                          021204                          ;GET FIRST HALF OF CRC
026366  116137  000004  001302          MOVB    4(R1),$TMP2             ;PUT IN $TMP2
026374  042737  177400  001302          BIC     #177400,$TMP2           ;CLEAR HI BYTE
026402  004737  030614                  JSR     PC,INRDY                ;WAIT FOR INRDY
026406  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026410  021244                          021244
026412  016104  000004                  MOV     4(R1),R4                ;PUT "FOUND" IN R4
026416  042704  000376                  BIC     #376,R4                 ;CLEAR UNWANTED BITS
026422  012705  000001                  MOV     #1,R5                   ;PUT "EXPECTED" IN R5
026426  120504                          CMPB    R5,R4                   ;IS IN BCC MATCH SET?
026430  001401                          BEQ     51$
026432  104015                          ERROR   15                      ;IN BCC MATCH ERROR
026434                          51$:
026434  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026436  021204                          021204                          ;GET LAST HALF
026440  116137  000004  001301          MOVB    4(R1),$TMP1+1           ;PUT IN $TMP1
026446  042737  000377  001300          BIC     #377,$TMP1              ;CLEAR LO BYTE
026454  053737  001300  001302          BIS     $TMP1,$TMP2             ;16 BIT BCC NOW IN $TMP2
026462  023737  030756  001302          CMP     CALBCC,$TMP2            ;IS IT CORRECT?
026470  001401                          BEQ     5$                      ;BR IF OK
026472  104027                          ERROR   27
026474                          5$:


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

                                        ;  TEST 54
                                        ;--------------

                                ;;*************************************************************
026474  000004                  TST54:  SCOPE
026476  012737  000054  001202          MOV     #54,$TSTNM              ; LOAD THE NO. OF THIS TEST
```

```
026504  012737  026726  001442          MOV     #TST55,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
026512  104410                          MSTCLR                          ;MASTER CLEAR KMC11
026514  012711  004000                  MOV     #BIT11,(R1)             ;SET LINE UNIT LOOP
026520  012702  031540                  MOV     #MESDAT,R2              ;R2 POINTS TO MESSAGE
026524  012700  000004                  MOV     #4,R0                   ;R0 = CHAR COUNT
026530  004737  031122                  JSR     PC,SYNLD                ;LOAD SILO WITH TWO SYNCS
026534  004737  030140                  JSR     PC,OUTRDY               ;WAIT FOR OUTRDY
026540  004537  031256                  JSR     R5,MESLD                ;LOAD MESSAGE IN SILO
026544  031540                          MESDAT                          ;START OF MESSAGE
026546  000004                          4                               ;CHARACTER COUNT
026550  004737  030006                  JSR     PC,OCOR                 ;WAIT FOR OCOR
026554  104413  000065                  DATACLK,        65              ;CLOCK DATA (EMPTY SILO)
026560  004537  031256                  JSR     R5,MESLD                ;PUT MORE CHARACTERS IN SILO
026564  031540                          MESDAT
026566  000004                          4
026570  004737  030006                  JSR     PC,OCOR
026574  104413  000005                  DATACLK,        5               ;CLOCK UNTIL RTS IS CLEARED
026600  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026602  021264                          021264                          ;GET RTS
026604  032761  000040  000004          BIT     #BIT5,4(R1)             ;IS IT CLEAR?
026612  001401                          BEQ     5$                      ;BR IF YES
026614  104034                          ERROR   34                      ;ERROR, RTS NOT CLEAR
026616  104413  000041          5$:     DATACLK,        41              ;CLOCK XMITTER SOME MORE
026622  004737  030614          1$:     JSR     PC,INRDY                ;OK LETS CHECK WHAT WAS RECEIVED
026626  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026630  021204                          021204                          ;GET RECEIVE DATA
026632  016104  000004                  MOV     4(R1),R4                ;PUT IT IN R4
026636  112205                          MOVB    (R2)+,R5                ;R5 = "EXPECTED"
026640  120504                          CMPB    R5,R4                   ;IS DATA CORRECT?
026642  001401                          BEQ     2$                      ;BR IF OK
026644  104010                          ERROR   10                      ;DATA ERROR
026646  005300                  2$:     DEC     R0                      ;DEC CHAR COUNT
026650  001364                          BNE     1$                      ;BR IF NOT DONE YET
026652  004737  030614          3$:     JSR     PC,INRDY                ;WAIT FOR INRDY
026656  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026660  021204                          021204                          ;GET RECEIVE DATA
026662  016104  000004                  MOV     4(R1),R4                ;PUT IT IN "FOUND"
026666  012705  000377                  MOV     #377,R5                 ;R5 = "EXPECTED"
026672  120504                          CMPB    R5,R4                   ;SHOULD SEE 377
026674  001401                          BEQ     4$                      ;BR IF OK
026676  104010                          ERROR   10                      ;ERROR, TRANSMITTER DID NOT ABORT
026700  004737  030614          4$:     JSR     PC,INRDY                ;WAIT FOR INRDY
026704  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026706  021204                          021204                          ;GET RECEIVE DATA
026710  016104  000004                  MOV     4(R1),R4                ;PUT IT IN "FOUND"
026714  012705  000377                  MOV     #377,R5                 ;R5 = "EXPECTED"
026720  120504                          CMPB    R5,R4                   ;SHOULD SEE 377
026722  001401                          BEQ     10$                     ;BR IF OK
026724  104010                          ERROR   10                      ;ERROR, TRANSMITTER DID NOT ABORT
026726                          10$:



                                        ;**************************** TEST 55 ****************************
                                        ;*HALF DUPLEX TEST
                                        ;*SET LINE UNIT LOOP AND HALF DUPLEX, SEND SYNCS AND A
```

```
                                                  ;*MESSAGE, VERIFY THAT IN-ACTIVE AND IN-READY ARE CLEAR
                                                  ;;*************************************************************

                                                  ;  TEST 55
                                                  ;--------------

                                       ;;*************************************************************
        026726  000004                 TST55:     SCOPE
        026730  012737  000055  001202            MOV     #55,$TSTNM             ; LOAD THE NO. OF THIS TEST
        026736  012737  027044  001442            MOV     #TST56,NEXT            ; POINT TO THE START OF NEXT TEST.
                                                                          ;R1 CONTAINS BASE KMC11 ADDRESS
        026744  104410                            MSTCLR                  ;MASTER CLEAR KMC11
        026746  012702  000012                    MOV     #12,R2          ;SAVE R2 FOR TYPEOUT
        026752  012711  004000                    MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
        026756  012761  000020  000004            MOV     #BIT4,4(R1)     ;LOAD PORT4
        026764  104412                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
        026766  122113                            122113                  ;SET H/D BIT
        026770  004737  031122                    JSR     PC,SYNLD        ;LOAD 2 SYNCS
        026774  004737  030140                    JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
        027000  004537  031256                    JSR     R5,MESLD        ;LOAD 4 CHAR MESSAGE
        027004  031540                            MESDAT                  ;ADDRESS OF MESSAGE
        027006  000004                            4                       ;CHARACTER COUNT
        027010  004737  030006                    JSR     PC,OCOR         ;WAIT FOR OCOR
        027014  104413  000073                    DATACLK,        73      ;SEND MESSAGE
        027020  104412                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
        027022  021244                            021244                  ;READ LU-12
        027024  016104  000004                    MOV     4(R1),R4        ;PUT "FOUND" IN R4
        027030  042704  000257                    BIC     #257,R4         ;CLEAR UNWANTED BITS
        027034  005005                            CLR     R5              ;R5 = "EXPECTED"
        027036  120504                            CMPB    R5,R4           ;IN-ACTIVE AND IN-RDY SHOULD BE CLEAR
        027040  001401                            BEQ     1$              ;BR IF OK
        027042  104035                            ERROR   35              ;ERROR BOTH ARE NOT CLEAR
        027044                         1$:



                                                  ;**************************** TEST 56 **************************
                                                  ;*DDCMP CABLE DATA TEST
                                                  ;*THIS TEST LOADS OUT SILO WITH THE FOLLOWING:
                                                  ;*4 SYNCS,16 CHAR,EOM,16 CHAR,EOM,16 CHAR,EOM
                                                  ;*THE 16 CHARACTERS INCLUDE A FLOATING ONE AND ZERO
                                                  ;*THE DATA IS TRANSMITTED OVER THE CABLE USING THE INTERNAL CLOCK
                                                  ;*RECEIVED DATA IS VERIFIED AS IS IN BCC MATCH
                                                  ;*LOOP-BACK CONNECTOR MUST BE ON TO RUN THIS TEST
                                                  ;;*************************************************************

                                                  ;  TEST 56
                                                  ;--------------

                                       ;;*************************************************************
        027044  000004                 TST56:     SCOPE
        027046  012737  000056  001202            MOV     #56,$TSTNM             ; LOAD THE NO. OF THIS TEST
        027054  012737  027434  001442            MOV     #TST57,NEXT            ; POINT TO THE START OF NEXT TEST.
                                                                          ;R1 CONTAINS BASE KMC11 ADDRESS
        027062  104410                            MSTCLR                  ;MASTER CLEAR KMC11
        027064  032737  040000  002050            BIT     #BIT14,STAT1    ;SKIP TEST IF NO
        027072  001557                            BEQ     3$              ;LOOPBACK CONNECTOR ON
        027074  012711  004000                    MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
        027100  004737  031122                    JSR     PC,SYNLD        ;LOAD 2 SYNCS
```

```
027104  004737  031122                          JSR     PC,SYNLD                ;LOAD 2 MORE SYNCS
027110  012737  120001  030754                  MOV     #CRC16,XPOLY            ;LOAD POLYNOMIAL FOR SOFT CRC CALC
027116  005037  027146                          CLR     6$                      ;CLEAR OLD BCC
027122  012703  000020                          MOV     #16.,R3                 ;CHARACTER COUNT
027126  012702  031544                          MOV     #FLTDAT,R2              ;R2= POINTER
027132  112237  027144                  7$:     MOVB    (R2)+,5$                ;LOAD CHAR FOR SOFT BCC CALC.
027136  004537  030650                          JSR     R5,SIMBCC               ;CALC SOFT BCC
027142  000010                                  10                              ;SHIFT COUNT
027144  000000                          5$:     0                               ;CHARACTER
027146  000000                          6$:     0                               ;OLD BCC
027150  013737  030756  027146                  MOV     CALBCC,6$               ;LOAD OLD BCC
027156  005303                                  DEC     R3                      ;DEC COUNT
027160  001364                                  BNE     7$                      ;BR IF NOT DONE YET
027162  004537  031256                          JSR     R5,MESLD                ;LOAD SILO
027166  031544                                  FLTDAT                          ;MESSAGE ADDRESS
027170  000020                                  16.                             ;CHARACTER COUNT
027172  004737  031232                          JSR     PC,EOM                  ;LOAD AN EOM
027176  004537  031256                          JSR     R5,MESLD                ;LOAD SILO
027202  031544                                  FLTDAT                          ;MESSAGE ADDRESS
027204  000020                                  16.                             ;CHARACTER COUNT
027206  004737  031232                          JSR     PC,EOM                  ;LOAD AN EOM
027212  004537  031256                          JSR     R5,MESLD                ;LOAD SILO
027216  031544                                  FLTDAT                          ;MESSAGE ADDRESS
027220  000020                                  16.                             ;CHARACTER COUNT
027222  004737  031232                          JSR     PC,EOM                  ;LOAD AN EOM
027226  004737  030006                          JSR     PC,OCOR                 ;WAIT FOR OCOR
027232  005011                                  CLR     (R1)                    ;CLEAR LINE UNIT LOOP
027234  012700  000003                          MOV     #3,R0                   ;R0 = MESSAGE COUNT
027240  012703  000020                          MOV     #16.,R3                 ;R3= CHARACTER COUNT
027244  012702  031544                          MOV     #FLTDAT,R2              ;LOAD MESSAGE POINTER IN R2
027250  004737  030614                  1$:     JSR     PC,INRDY                ;WAIT FOR INRDY
027254  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027256  021204                                  021204                          ;GET DATA FROM IN SILO
027260  016104  000004                          MOV     4(R1),R4                ;PUT CHARACTER IN "FOUND"
027264  112205                                  MOVB    (R2)+,R5                ;PUT "EXPECTED" IN R5
027266  120504                                  CMPB    R5,R4                   ;IS RECEIVED DATA CORRECT
027270  001401                                  BEQ     2$                      ;BR IF OK
027272  104025                                  ERROR   25                      ;DATA ERROR
027274                                  2$:
027274  005303                                  DEC     R3                      ;DEC CHARACTER COUNT
027276  001364                                  BNE     1$                      ;BR IF NOT DONE THIS MESSAGE
027300  012703  000020                          MOV     #16.,R3                 ;RESET CHARACTER COUNT

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

027304  004737  030614                          JSR     PC,INRDY                ;WAIT FOR INRDY
027310  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027312  021204                                  021204                          ;GET FIRST HALF OF CRC
027314  116137  000004  001302                  MOVB    4(R1),$TMP2             ;PUT IN $TMP2
027322  042737  177400  001302                  BIC     #177400,$TMP2           ;CLEAR HI BYTE
027330  004737  030614                          JSR     PC,INRDY                ;WAIT FOR INRDY
027334  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027336  021244                                  021244
027340  016104  000004                          MOV     4(R1),R4                ;PUT "FOUND" IN R4
027344  042704  000376                          BIC     #376,R4                 ;CLEAR UNWANTED BITS
```

```
027350  012705  000001                   MOV     #1,R5           ;PUT "EXPECTED" IN R5
027354  120504                           CMPB    R5,R4           ;IS IN BCC MATCH SET?
027356  001401                           BEQ     25$
027360  104015                           ERROR   15              ;IN BCC MATCH ERROR
027362                           25$:
027362  104412                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027364  021204                           021204                  ;GET LAST HALF
027366  116137  000004  001301           MOVB    4(R1),$TMP1+1   ;PUT IN $TMP1
027374  042737  000377  001300           BIC     #377,$TMP1      ;CLEAR LO BYTE
027402  053737  001300  001302           BIS     $TMP1,$TMP2     ;16 BIT BCC NOW IN $TMP2
027410  023737  030756  001302           CMP     CALBCC,$TMP2    ;IS IT CORRECT?
027416  001401                           BEQ     4$              ;BR IF OK
027420  104027                           ERROR   27
027422  012702  031544           4$:     MOV     #FLTDAT,R2      ;RESET MESSAGE POINTER
027426  005300                           DEC     R0              ;DECREMENT COUNTER
027430  001307                           BNE     1$              ;BR IF NOT DONE
027432  104420                   3$:     ADVANCE         ; ADVANCE TO NEXT TEST


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


                                         ;  TEST 57
                                         ;--------------

                                 ;;*****************************************************************
027434  000004                   TST57:  SCOPE
027436  012737  000057  001202           MOV     #57,$TSTNM              ; LOAD THE NO. OF THIS TEST
027444  012737  003662  001442           MOV     #$EOP,NEXT              ; POINT TO THE END OF PASS HANDLER.
                                                                 ;R1 CONTAINS BASE KMC11 ADDRESS
027452  104410                           MSTCLR                  ;MASTER CLEAR KMC11
027454  032737  040000  002050           BIT     #BIT14,STAT1    ;SKIP TEST IF NO
027462  001533                           BEQ     3$              ;LOOPBACK CONNECTOR ON
027464  012711  004000                   MOV     #BIT11,(R1)     ;SET LINE UNIT LOOP
027470  004737  031122                   JSR     PC,SYNLD        ;LOAD 2 SYNCS
027474  004737  031122                   JSR     PC,SYNLD        ;LOAD 2 MORE SYNCS
027500  012737  120001  030754           MOV     #CRC16,XPOLY    ;LOAD POLYNOMIAL FOR SOFT CRC CALC
027506  005037  027536                   CLR     6$              ;CLEAR OLD BCC
027512  012703  000073                   MOV     #59.,R3         ;CHARACTER COUNT
027516  012702  031540                   MOV     #MESDAT,R2      ;R2= POINTER
027522  112237  027534           7$:     MOVB    (R2)+,5$        ;LOAD CHAR FOR SOFT BCC CALC.
027526  004537  030650                   JSR     R5,SIMBCC       ;CALC SOFT BCC
027532  000010                           10                      ;SHIFT COUNT
027534  000000                   5$:     0                       ;CHARACTER
027536  000000                   6$:     0                       ;OLD BCC
027540  013737  030756  027536           MOV     CALBCC,6$       ;LOAD OLD BCC
027546  005303                           DEC     R3              ;DEC COUNT
027550  001364                           BNE     7$              ;BR IF NOT DONE YET
027552  004537  031256                   JSR     R5,MESLD        ;LOAD SILO
027556  031540                           MESDAT                  ;MESSAGE ADDRESS
027560  000073                           59.                     ;CHARACTER COUNT
```

```
027562  004737  031232                          JSR     PC,EOM          ;LOAD AN EOM
027566  004737  030006                          JSR     PC,OCOR         ;WAIT FOR OCOR
027572  005011                                  CLR     (R1)            ;CLEAR LINE UNIT LOOP
027574  012700  000073                          MOV     #59.,R0         ;R0= CHARACTER COUNT
027600  012702  031540                          MOV     #MESDAT,R2      ;LOAD MESSAGE POINTER IN R2
027604  004737  030614                  1$:     JSR     PC,INRDY        ;WAIT FOR INRDY
027610  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027612  021204                                  021204                  ;GET DATA FROM IN SILO
027614  016104  000004                          MOV     4(R1),R4        ;PUT CHARACTER IN "FOUND"
027620  112205                                  MOVB    (R2)+,R5        ;PUT "EXPECTED" IN R5
027622  120504                                  CMPB    R5,R4           ;IS RECEIVED DATA CORRECT
027624  001401                                  BEQ     2$              ;BR IF OK
027626  104025                                  ERROR   25              ;DATA ERROR
027630                                  2$:
027630  005300                                  DEC     R0              ;DECREMENT COUNTER
027632  001364                                  BNE     1$              ;BR IF NOT DONE

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

027634  004737  030614                          JSR     PC,INRDY        ;WAIT FOR INRDY
027640  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027642  021204                                  021204                  ;GET FIRST HALF OF CRC
027644  116137  000004  001302                  MOVB    4(R1),$TMP2     ;PUT IN $TMP2
027652  042737  177400  001302                  BIC     #177400,$TMP2   ;CLEAR HI BYTE
027660  004737  030614                          JSR     PC,INRDY        ;WAIT FOR INRDY
027664  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027666  021244                                  021244
027670  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
027674  042704  000376                          BIC     #376,R4         ;CLEAR UNWANTED BITS
027700  012705  000001                          MOV     #1,R5           ;PUT "EXPECTED" IN R5
027704  120504                                  CMPB    R5,R4           ;IS IN BCC MATCH SET?
027706  001401                                  BEQ     25$
027710  104015                                  ERROR   15              ;IN BCC MATCH ERROR
027712                                  25$:
027712  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027714  021204                                  021204                  ;GET LAST HALF
027716  116137  000004  001301                  MOVB    4(R1),$TMP1+1   ;PUT IN $TMP1
027724  042737  000377  001300                  BIC     #377,$TMP1      ;CLEAR LO BYTE
027732  053737  001300  001302                  BIS     $TMP1,$TMP2     ;16 BIT BCC NOW IN $TMP2
027740  023737  030756  001302                  CMP     CALBCC,$TMP2    ;IS IT CORRECT?
027746  001401                                  BEQ     3$              ;BR IF OK
027750  104027                                  ERROR   27
027752  104420                          3$:     ADVANCE         ; ADVANCE TO NEXT TEST



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

027754                                  GETSI:
                                                ;THIS SUBROUTINE READS LU 17, AND PUTS IT INTO NITCH.
                                                ;NITCH IS ROTATED LEFT UNTILL THE SI BIT IS IN CARRY

027754  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027756  021364                                  021364                  ;PORT4+LU 17
027760  017737  152110  030004                  MOV     @KMP04,NITCH    ;STORE LU 17
```

```
 027766  106137  030004                     ROLB    NITCH
 027772  106137  030004                     ROLB    NITCH
 027776  106137  030004                     ROLB    NITCH                ;PUT SI IN THE CARRY BIT
 030002  000207                             RTS     PC
 030004  000000                    NITCH:   0


 030006                            OCOR:
                                            ;THIS SUBROUTINE SPINS ON OCOR

 030006  104412                             ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 030010  021364                             021364                       ;PORT4+LU 17
 030012  032777  000020  152054             BIT     #BIT4,@KMP04         ;IS OCOR SET?
 030020  001772                             BEQ     OCOR                 ;BR IF NO
 030022  000207                             RTS     PC                   ;OK OCOR IS SET, GO BACK


 030024                            SYNC:
                                            ;THIS SUBROUNTINE LOADS THE SILO WITH THE NUMBER OF SYNC
                                            ;CHARACTERS PASSED TO IT IN THE WORD AFTER THE JSR CALL
                                            ;AND A NON-SYNC CHARACTER (301)

 030024  013637  001276                     MOV     @(SP)+,$TMP0         ;GET COUNT
 030030  062746  000002                     ADD     #2,-(SP)             ;ADJUST STACK
 030034  012761  000026  000004             MOV     #26,4(R1)            ;LOAD PORT4
 030042  104412                             ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 030044  122114                             122114                       ;LOAD SYNC REGISTER
 030046  004737  030140            1$:      JSR     PC,OUTRDY            ;WAIT FOR OUTRDY
 030052  012761  000001  000004             MOV     #1,4(R1)             ;LOAD PORT4
 030060  104412                             ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 030062  122111                             122111                       ;SET SOM
 030064  012761  000026  000004             MOV     #26,4(R1)            ;LOAD PORT4
 030072  104412                             ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 030074  122110                             122110                       ;LOAD OUT DATA
 030076  005337  001276                     DEC     $TMP0                ;ALL DONE?
 030102  001361                             BNE     1$                   ;BR IF NOT
 030104  004737  030140                     JSR     PC,OUTRDY            ;WAIT FOR OUTRDY
 030110  005061  000004                     CLR     4(R1)                ;LOAD PORT4
 030114  104412                             ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 030116  122111                             122111                       ;SET SOM
 030120  012761  000301  000004             MOV     #301,4(R1)           ;LOAD PORT4
 030126  104412                             ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 030130  122110                             122110                       ;LOAD OUT DATA
 030132  004737  030006                     JSR     PC,OCOR              ;WAIT FOR OCOR
 030136  000207                             RTS     PC


 030140                            OUTRDY:
                                            ;THIS SUBROUTINE SPINS ON OUT READY

 030140  005037  001306                     CLR     $TMP4                ;CLEAR TIMER
 030144                            1$:
 030144  104412                             ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 030146  021224                             021224                       ;PORT4+LU11
 030150  032777  000020  151716             BIT     #BIT4,@KMP04         ;IS OUT RDY SET?
 030156  001004                             BNE     2$                   ;BR IF YES
```

```
      030160  005237  001306                   INC     $TMP4            ;INC TIMER
      030164  001367                           BNE     1$               ;KEEP CHECKING IF NOT DONE
      030166  104036                           ERROR   36               ;ERROR, OUT READY NOT SET
      030170  000207                   2$:     RTS     PC


      030172                           CHAR:
                                               ;THIS SUBROUTINE LOADS THE SILO WITH 3 SYNCS
                                               ;AND THE CHARACTER PASSED TO IT.

      030172  013637  001300                   MOV     @(SP)+,$TMP1     ;GET CHARACTER
      030176  062746  000002                   ADD     #2,-(SP)         ;ADJUST STACK
      030202  012737  000003  001276           MOV     #3,$TMP0         ;SET FOR 3 SYNCS
      030210  012761  000026  000004           MOV     #26,4(R1)        ;LOAD PORT4
      030216  104412                           ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      030220  122114                           122114                   ;LOAD SYNC REGISTER
      030222  004737  030140           1$:     JSR     PC,OUTRDY        ;WAIT FOR OUTRDY
      030226  012761  000001  000004           MOV     #1,4(R1)         ;LOAD PORT4
      030234  104412                           ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      030236  122111                           122111                   ;SET SOM
      030240  012761  000026  000004           MOV     #26,4(R1)        ;LOAD PORT4
      030246  104412                           ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      030250  122110                           122110                   ;LOAD OUT DATA
      030252  005337  001276                   DEC     $TMP0            ;ALL DONE?
      030256  001361                           BNE     1$               ;BR IF NOT
      030260  004737  030140                   JSR     PC,OUTRDY        ;WAIT FOR OUTRDY
      030264  013761  001300  000004           MOV     $TMP1,4(R1)      ;LOAD PORT4
      030272  104412                           ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      030274  122110                           122110                   ;LOAD OUT DATA
      030276  004737  030006                   JSR     PC,OCOR          ;WAIT FOR OCOR
      030302  000207                           RTS     PC


      030304                           CHARSD:
                                               ;THIS SUBROUTINE LOADS THE SILO WITH THE CHARACTER PASSED TO IT.

      030304  013637  001300                   MOV     @(SP)+,$TMP1     ;GET CHARACTER
      030310  062746  000002                   ADD     #2,-(SP)         ;ADJUST STACK
      030314  004737  030140                   JSR     PC,OUTRDY        ;WAIT FOR OUTRDY
      030320  013761  001300  000004           MOV     $TMP1,4(R1)      ;LOAD PORT4
      030326  104412                           ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      030330  122110                           122110                   ;LOAD OUT DATA
      030332  004737  030140                   JSR     PC,OUTRDY        ;WAIT FOR OUTRDY
      030336  104412                           ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      030340  122110                           122110                   ;LOAD GARBAGE CHAR
      030342  004737  030006                   JSR     PC,OCOR          ;WAIT FOR OCOR
      030346  000207                           RTS     PC


      030350                           SILOLD:
                                               ;THIS SUBROUTINE FILLS THE OUT SILO
                                               ; WITH A BINARY COUNT PATTERN

      030350  012737  000073  001300           MOV     #73,$TMP1        ;LOAD COUNT
      030356  005737  030610                   TST     SCHAR            ;FIRST TIME HERE?
      030362  100470                           BMI     4$               ;BR IF BITSTUFF
```

```
030364  001032                                  BNE     2$              ;BR IF NO
030366  062737  000002  001300                  ADD     #2,$TMP1        ;ADD 2 TO CHARACTER COUNT
030374  012737  000003  001276                  MOV     #3,$TMPO        ;SET FOR 3 SYNCS
030402  012761  000026  000004                  MOV     #26,4(R1)       ;LOAD PORT4
030410  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030412  122114                                  122114                  ;LOAD SYNC REGISTER
030414  004737  030140                  1$:     JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
030420  012761  000001  000004                  MOV     #1,4(R1)        ;LOAD PORT4
030426  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030430  122111                                  122111                  ;SET SOM
030432  012761  000026  000004                  MOV     #26,4(R1)       ;LOAD PORT4
030440  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030442  122110                                  122110                  ;LOAD OUT DATA
030444  005337  001276                          DEC     $TMPO           ;ALL DONE?
030450  001361                                  BNE     1$              ;BR IF NOT
030452  004737  030140                  2$:     JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
030456  013761  030610  000004                  MOV     SCHAR,4(R1)     ;LOAD PORT4
030464  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030466  122110                                  122110                  ;LOAD OUT DATA
030470  005737  030612                          TST     STUFLG          ;BITSTUFF???
030474  001407                                  BEQ     6$              ;BR IF NO
030476  013737  030610  030510                  MOV     SCHAR,5$        ;IT IS SDLD SO CHECK BITSTUFFING
030504  004537  031340                          JSR     R5,STFFCL       ;ADD ANY BIT STUFF CLOCK TICKS
030510  000000                          5$:     0                       ;CHARACTER
030512  000010                                  10                      ;CHIFT COUNT
030514  005237  030610                  6$:     INC     SCHAR           ;NEXT CHARACTER
030520  022737  000400  030610                  CMP     #400,SCHAR      ;ALL DONE?
030526  001403                                  BEQ     3$
030530  005337  001300                          DEC     $TMP1           ;DECREMENT COUNT
030534  001346                                  BNE     2$              ;BR IF NOT DONE
030536  004737  030006                  3$:     JSR     PC,OCOR         ;WAIT FOR OCOR
030542  000207                                  RTS     PC
030544  005037  030610                  4$:     CLR     SCHAR           ;START PATTERN AT ZERO
030550  012737  177777  030612                  MOV     #-1,STUFLG      ;SET BITSTUFF FLAG
030556  005037  031536                          CLR     BITCON          ;CLEAR STUFF COUNT
030562  062737  000002  001300                  ADD     #2,$TMP1        ;ADD 2 TO CHARACTER COUNT
030570  012761  000001  000004                  MOV     #1,4(R1)        ;SET BIT0 IN PORT4
030576  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030600  122111                                  122111                  ;SET SOM!
030602  104412                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030604  122110                                  122110                  ;LOAD GARBAGE CHAR
030606  000721                                  BR      2$              ;GO LOAD SILO
030610  000000                          SCHAR:  0
030612  000000                          STUFLG: 0


030614                                  INRDY:
                                                ;THIS SUBROUTINE SPINS ON INRDY
                                                ;IF INRDY FAILS TO SET THE DELAY TIMES OUT AND AN
                                                ;ERROR IS REPORTED. FOR BETTER SCOPE LOOPS THIS
                                                ;DELAY CAN BE MADE SHORTER BY ALTERING THE NUMBER
                                                ;INITIALLY LOADED INTO $TMPO, THE SMALLER THE NUMBER
                                                ;THE SHORTER THE DELAY. 0 IS THE LONGEST DELAY.

030614  012737  000000  001276                  MOV     #0,$TMPO        ;SET UP DELAY COUNTER
030622                                  1$:
```

```
030622  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030624  021244                                  021244                          ;PORT4+LU12
030626  032777  000020  151240                  BIT     #BIT4,@KMP04            ;IS INRDY SET?
030634  001004                                  BNE     2$                      ;BR IF YES
030636  005237  001276                          INC     $TMP0                   ;INC DELAY
030642  001367                                  BNE     1$                      ;TRY AGAIN
030644  104037                                  ERROR   37                      ;ERROR,NO INRDY
030646  000207                          2$:     RTS     PC                      ;RETURN


030650                                  SIMBCC:
                                                ;THIS SUBROUTINE CALCULATES THE CRC USING POLYNOMIAL GIVEN
                                                ;IN XPOLY. THE CORRECT CRC IS $LPADRED IN CALBCC, AND THE
                                                ;STATE OF THE LSB OF THE BCC IS $LPADRED IN THE C BIT.

030650  010046                                  MOV     R0,-(SP)                ;SAVE R0 ON STACK
030652  012537  001276                          MOV     (R5)+,$TMP0             ;$TMP0 = SHIFT COUNT
030656  012537  001300                          MOV     (R5)+,$TMP1             ;$TMP1 = CHARACTER
030662  012537  030756                          MOV     (R5)+,CALBCC            ;CALBCC = OLD BCC
030666  013700  030756                  1$:     MOV     CALBCC,R0               ;PUT OLD BCC IN R0
030672  000241                                  CLC
030674  006037  030756                          ROR     CALBCC                  ;SHIFT OLD BCC
030700  006037  001300                          ROR     $TMP1                   ;SHIFT CHARACTER
030704  005500                                  ADC     R0                      ;ADD CHAR CARRY TO OLD BCC
030706  006000                                  ROR     R0                      ;PUT BIT0 TO CARRY BIT
030710  103011                                  BCC     2$                      ;CARRY IS FEEDBACK BIT
030712  013700  030754                          MOV     XPOLY,R0                ;IF FEEDBACK = 1
030716  043700  030756                          BIC     CALBCC,R0               ;EXCLUSIVLY OR XPOLY TO CALBCC
030722  043737  030754  030756                  BIC     XPOLY,CALBCC
030730  050037  030756                          BIS     R0,CALBCC
030734  005337  001276                  2$:     DEC     $TMP0                   ;DEC SHIFT COUNT
030740  001352                                  BNE     1$                      ;BR IF NOT DONE
030742  013700  030756                          MOV     CALBCC,R0               ;PUT RESULT IN R0
030746  006000                                  ROR     R0                      ;SHIFT BIT0 TO CARRY
030750  012600                                  MOV     (SP)+,R0                ;RESTORE R0
030752  000205                                  RTS     R5                      ;$LPADR
030754  000000                          XPOLY:  0
030756  000000                          CALBCC: 0
        000200                          LRC8=200
        120001                          CRC16=120001
        102010                          CRC.CCITT=102010


030760                                  BCCLD:
                                                ;THIS SUBROUTINE LOADS THE OUT SILO WITH 2 SYNCS
                                                ;WITH SOM SET, AND ONE CHARACTER PASSED TO IT
                                                ;WITH THE SOM BIT CLEAR (ENABLE CRC)

030760  013637  001300                          MOV     @(SP)+,$TMP1            ;GET CHARACTER
030764  062746  000002                          ADD     #2,-(SP)                ;ADJUST STACK
030770  012737  000002  001276                  MOV     #2,$TMP0                ;SET FOR 2 SYNCS
030776  012761  000026  000004                  MOV     #26,4(R1)               ;LOAD PORT4
031004  104412                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031006  122114                                  122114                          ;LOAD SYNC REGISTER
031010  004737  030140                  1$:     JSR     PC,OUTRDY               ;WAIT FOR OUTRDY
031014  012761  000001  000004                  MOV     #1,4(R1)                ;LOAD PORT4
```

```
031022  104412                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031024  122111                                122111                  ;SET SOM
031026  012761  000026  000004                MOV     #26,4(R1)       ;LOAD PORT4
031034  104412                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031036  122110                                122110                  ;LOAD OUT DATA
031040  005337  001276                        DEC     $TMPO           ;ALL DONE?
031044  001361                                BNE     1$              ;BR IF NOT
031046  004737  030140                        JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
031052  013761  001300  000004                MOV     $TMP1,4(R1)     ;LOAD PORT4
031060  104412                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031062  122110                                122110                  ;LOAD OUT DATA
031064  004737  030006                        JSR     PC,OCOR         ;WAIT FOR OCOR
031070  000207                                RTS     PC


031072                                GETQO:
                                              ;THIS SUBROUTINE READS THE STATE OF THE TRANSMIT
                                              ;BCC LSB AND PUTS IT IN THE CARRY BIT

031072  104412                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031074  021364                                021364                  ;PORT4←LU-17
031076  106177  150772                        ROLB    @KMPO4          ;PUT QO IN CARRY
031102  000207                                RTS     PC              ;RETURN


031104                                GETQI:
                                              ;THIS SUBROUTINE READS THE STATE OF THE RECEIVE
                                              ;BCC LSB AND PUTS IT IN THE CARRY BIT

031104  104412                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031106  021364                                021364                  ;PORT4←LU-17
031110  106177  150760                        ROLB    @KMPO4          ;PUT QO IN CARRY
031114  106177  150754                        ROLB    @KMPO4          ;PUT QI IN CARRY
031120  000207                                RTS     PC              ;RETURN


031122                                SYNLD:
                                              ;THIS SUBROUTINE LOADS OUT SILO WITH
                                              ;2 SYNC CHARACTERS WITH SOM SET

031122  012737  000002  001276                MOV     #2,$TMPO        ;LOAD COUNTER FOR 2 SYNCS
031130  012761  000026  000004                MOV     #26,4(R1)       ;PORT4←26
031136  104412                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031140  122114                                122114                  ;LOAD SYNC REG
031142  004737  030140                1$:     JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
031146  012761  000001  000004                MOV     #1,4(R1)        ;LOAD PORT4
031154  104412                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031156  122111                                122111                  ;SET SOM
031160  012761  000026  000004                MOV     #26,4(R1)       ;PORT←26
031166  104412                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031170  122110                                122110                  ;LOAD OUT DATA WITH SYNC
031172  005337  001276                        DEC     $TMPO           ;DECREMENT COUNTER
031176  001361                                BNE     1$              ;BR IF NOT DONE
031200  000207                                RTS     PC              ;RETURN
```

```
031202                          SOM:
                                        ;THIS SUBROUTINE LOADS SOM AND OUT DATA WITH A
                                        ;GARBAGE CHARACTER (0)

031202  004737  030140                  JSR     PC,OUTRDY               ;WAIT FOR OUTRDY
031206  012761  000001  000004          MOV     #1,4(R1)                ;PORT4+1
031214  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031216  122111                          122111                          ;SET SOM
031220  005061  000004                  CLR     4(R1)                   ;CLEAR DATA CHAR
031224  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031226  122110                          122110                          ;LOAD GARBAGE CHARACTER
031230  000207                          RTS     PC                      ;RETURN


031232                          EOM:
                                        ;THIS SUBROUTINE LOADS EOM AND OUT DATA WITH A
                                        ;GARBAGE CHARACTER (2) TO ENABLE TRANSMISSION OF BCC

031232  004737  030140                  JSR     PC,OUTRDY               ;WAIT FOR OUTRDY
031236  012761  000002  000004          MOV     #2,4(R1)                ;PORT4+2
031244  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031246  122111                          122111                          ;SET EOM
031250  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031252  122110                          122110                          ;LOAD GARBAGE CHARACTER
031254  000207                          RTS     PC                      ;RETURN


031256                          MESLD:
                                        ;THIS SUBROUTINE LOADS SILO WITH MESSAGE
                                        ;THE FIRST ARUGUMENT IS THE ADDRESS OF THE MESSAGE
                                        ;THE SECOND ARGUMENT IS THE NUMBER OF CHARACTERS IN THE MESSAGE

031256  010046                          MOV     R0,-(SP)                ;SAVE R0
031260  012500                          MOV     (R5)+,R0                ;R0=MESSAGE POINTER
031262  012537  001276                  MOV     (R5)+,$TMP0             ;$TMP0=CHARACTER COUNT
031266  004737  030140          1$:     JSR     PC,OUTRDY               ;WAIT FOR OUT RDY
031272  112061  000004                  MOVB    (R0)+,4(R1)             ;LOAD PORT4 WITH CHARACTER
031276  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031300  122110                          122110                          ;LOAD OUT DATA SILO
031302  005337  001276                  DEC     $TMP0                   ;DEC CHAR COUNT
031306  001367                          BNE     1$                      ;BR IF NOT DONE
031310  004737  030006                  JSR     PC,OCOR                 ;WAIT FOR OCOR
031314  012600                          MOV     (SP)+,R0                ;RESTORE R0
031316  000205                          RTS     R5                      ;RETURN


031320                          CLRIO:
                                        ;THIS SUBROUTINE SETS IN CLR AND OUT CLR TO
                                        ;CLEAR THE TRANSMIT AND RECEIVE BCC REGISTERS

031320  012761  000200  000004          MOV     #BIT7,4(R1)             ;LOAD PORT4
031326  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031330  122112                          122112                          ;SET IN CLR!
031332  104412                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031334  122111                          122111                          ;SET OUT CLR!
031336  000207                          RTS     PC                      ;RETURN
```

```
031340                           STFFCL:
                                         ;THIS SUBROUTINE ADDS ANY NECESSSARY BIT STUFF CLOCK TICKS
                                         ;FIRST ARGUMENT IS CHAR, SECOND ARGUMENT IS SHIFT COUNT.

031340  010046                           MOV     R0,-(SP)        ;SAVE R0
031342  012500                           MOV     (R5)+,R0        ;PUT CHAR IN R0
031344  012537  001302                   MOV     (R5)+,$TMP2     ;PUT SHIFT COUNT IN $TMP2
031350  106000                   1$:     RORB    R0              ;LOOK AT NEXT BIT
031352  103403                           BCS     2$              ;BR IF A MARK
031354  005037  031536                   CLR     BITCON          ;IT WAS A SPACE, CLEAR 1'S COUNTER
031360  000412                           BR      3$              ;CONTINUE
031362  005237  031536           2$:     INC     BITCON          ;INC CONSECUTIVE 1'S COUNTER
031366  022737  000005  031536           CMP     #5,BITCON       ;IS IT 5 YET?
031374  001004                           BNE     3$              ;BR IF NO
031376  005037  031536                   CLR     BITCON          ;YES!  SO START AGAIN
031402  104413  000001                   DATACLK,        1       ;GIVE EXTRA TICK TO STUFF ZERO
031406  005337  001302           3$:     DEC     $TMP2           ;DEC SHIFT COUNT
031412  001356                           BNE     1$              ;BR IF NOT DONE
031414  012600                           MOV     (SP)+,R0        ;RESTORE R0
031416  000205                           RTS     R5              ;RETURN


031420                           STFFCK:
                                         ;THIS SUBROUTINE CHECKS TO SEE IF TRANSMITTER
                                         ;IS STUFFING ZEROS WHEN IT SHOULD. FIRST ARGUMENT
                                         ;IS THE CHARACTER, SECOND ARGUMENT IS SHIFT COUNT.

031420  010046                           MOV     R0,-(SP)        ;SAVE R0
031422  012500                           MOV     (R5)+,R0        ;PUT CHAR IN R0
031424  012537  001302                   MOV     (R5)+,$TMP2     ;PUT SHIFT COUNT IN $TMP2
031430  106000                   1$:     RORB    R0              ;SHIFT OUT NEXT BIT
031432  103403                           BCS     2$              ;BR IF IT IS A MARK
031434  005037  031536                   CLR     BITCON          ;IT WAS A SPACE, CLEAR 1'S COUNTER
031440  000416                           BR      3$              ;CONTINUE
031442  005237  031536           2$:     INC     BITCON          ;INC CONSECUTIVE I'S COUNTER
031446  022737  000005  031536           CMP     #5,BITCON       ;5 IN A ROW YET?
031454  001010                           BNE     3$              ;BR IF NO
031456  005037  031536                   CLR     BITCON          ;YES, SO START OVER
031462  104413  000001                   DATACLK,        1       ;EXTRA TICK TO STUFF ZERO
031466  004737  027754                   JSR     PC,GETSI        ;LOOK AT WINDOW
031472  103001                           BCC     3$              ;IS IT A ZERO, BR IF YES
031474  104030                           ERROR   30              ;NO, ERROR ZERO WAS NOT STUFFED
031476  005337  001302           3$:     DEC     $TMP2           ;DEC SHIFT COUNT
031502  001352                           BNE     1$              ;BR IF NOT DONE
031504  012600                           MOV     (SP)+,R0        ;RESTORE R0
031506  000205                           RTS     R5              ;RETURN


031510                           CTSDLY:
                                         ;THIS SUBROUTINE WASTES TIME UNTIL CTS SETS,
                                         ;BUT HOPEFULLY NOT SO LONG THAT THE SILO RUNS OUT

031510  010046                           MOV     R0,-(SP)        ;SAVE R0
031512  012700  000032                   MOV     #32,R0          ;LOAD R0 WITH COUNT
```

```
031516  027777  147522  147520  1$:     CMP     @STKS,@STKS             ;WASTE TIME
031524  005300                          DEC     R0                      ;DECREMENT COUNTER
031526  001373                          BNE     1$                      ;DO IT AGAIN IF NOT = 0
031530  012600                          MOV     (SP)+,R0                ;RESTORE R0
031532  000207                          RTS     PC                      ;RETURN


031534  000176                          FLAG:   ↑B<01111110>            ;FLAG CHARACTER
031536  000000                          BITCON: 0
031540     000     125     252  MESDAT: .BYTE   0,125,252,377
031543     377
031544     001     002     004  FLTDAT: .BYTE   1,2,4,10,20,40,100,200,376,375,373,367,357,337,277,177
031547     010     020     040
031552     100     200     376
031555     375     373     367
031560     357     337     277
031563     177
031564     100     140     160  STUFDT: .BYTE   100,140,160,170,3,300,174,176,177,1
031567     170     003     300
031572     174     176     177
031575     001
031576     363     347     317          .BYTE   363,347,317,200,0,377,377,377,200,37
031601     200     000     377
031604     377     377     200
031607     037
                                .EVEN
031610  046200  047111  020105  EM1:    .ASCIZ  <200>/LINE UNIT INITALIZATION TEST/
031646  046200  047111  020105  EM2:    .ASCIZ  <200>↑LINE UNIT REGISTER READ/ONLY TEST↑
031711     200  044514  042516  EM3:    .ASCIZ  <200>↑LINE UNIT REGISTER WRITE/READ TEST↑
031755     200  044514  042516  EM4:    .ASCIZ  <200>/LINE UNIT INTERNAL CLOCK FAILURE/
032017     200  051124  047101  EM5:    .ASCIZ  <200>/TRANSMITTER DATA ERROR/
032047     200  042522  042503  EM6:    .ASCIZ  <200>/RECEIVER TEST/
032066  051200  041505  044505  EM7:    .ASCIZ  <200>/RECEIVER DATA ERROR/
032113     200  047515  042504  EM10:   .ASCIZ  <200>/MODEM SIGNAL ERROR/
032137     200  051124  047101  EM11:   .ASCIZ  <200>/TRANSMITTER CRC ERROR/
032166  051200  041505  044505  EM12:   .ASCIZ  <200>/RECEIVER CRC ERROR/
032212  044600  020116  041502  EM13:   .ASCIZ  <200>/IN BCC MATCH ERROR (LU REG 12)/
032252  052200  040522  051516  EM14:   .ASCIZ  <200>/TRANSMITTER FAILED TO GO TO MARK STATE/
032322  041600  041101  042514  EM15:   .ASCIZ  <200>/CABLE DATA TEST/
032343     200  046106  043501  EM16:   .ASCIZ  <200>/FLAG ERROR/
032357     200  051124  047101  EM17:   .ASCIZ  <200>/TRANSMITTER FAILED TO STUFF A ZERO/
032423     200  053523  052111  EM20:   .ASCIZ  <200>/SWITCH PAC TEST/
032444  040600  047502  052122  EM21:   .ASCIZ  <200>/ABORT ERROR/
032461     200  051124  047101  EM22:   .ASCIZ  <200>/TRANSMITTER ERROR/
032504  044200  046101  020106  EM23:   .ASCIZ  <200>/HALF DUPLEX TEST/
032526  047600  052125  051040  EM24:   .ASCIZ  <200>/OUT READY NOT SET/
032551     200  047111  051040  EM25:   .ASCIZ  <200>/IN READY NOT SET/

032573     200  054105  042520  DH1:    .ASCIZ  <200>/EXPECTED  FOUND/
032614  042600  050130  041505  DH2:    .ASCIZ  <200>/EXPECTED  FOUND  LU-REGISTER/
032652  041600  040510  040522  DH3:    .ASCIZ  <200>/CHARACTER    BIT THAT FAILED/
032710  041600  051117  042522  DH4:    .ASCIZ  <200>/CORRECT CRC    BIT THAT FAILED/
032750  042600  050130  041505  DH5:    .ASCIZ  <200>/EXPECTED   FOUND   SHIFT/
033002  042600  050130  041505  DH6:    .ASCIZ  <200>/EXPECTED   FOUND   CHARACTER   SHIFT/
033050  041200  047514  045503  DH7:    .ASCIZ  <200>/BLOCK END NOT SET/
033073     200  052122  020123  DH10:   .ASCIZ  <200>/RTS DID NOT CLEAR/
```

```
                                       .EVEN

        033116  000002                 DT1:    2
        033120     003      007                .BYTE   3,7
        033122  001274                         $REG5
        033124     003      002                .BYTE   3,2
        033126  001272                         $REG4
        033130  000003                 DT2:    3
        033132     003      007                .BYTE   3,7
        033134  001274                         $REG5
        033136     003      010                .BYTE   3,10
        033140  001272                         $REG4
        033142     003      002                .BYTE   3,2
        033144  001266                         $REG2
        033146  000002                 DT3:    2
        033150     003      017                .BYTE   3,17
        033152  001274                         $REG5
        033154     002      002                .BYTE   2,2
        033156  001270                         $REG3
        033160  000002                 DT4:    2
        033162     006      021                .BYTE   6,21
        033164  030756                         CALBCC
        033166     002      002                .BYTE   2,2
        033170  001270                         $REG3
        033172  000003                 DT5:    3
        033174     001      011                .BYTE   1,11
        033176  001462                         ZERO
        033200     001      011                .BYTE   1,11
        033202  001464                         ONE
        033204     002      002                .BYTE   2,2
        033206  001262                         $REG0
        033210  000003                 DT6:    3
        033212     001      011                .BYTE   1,11
        033214  001464                         ONE
        033216     001      011                .BYTE   1,11
        033220  001462                         ZERO
        033222     002      002                .BYTE   2,2
        033224  001262                         $REG0
        033226  000004                 DT7:    4
        033230     001      011                .BYTE   1,11
        033232  001462                         ZERO
        033234     001      011                .BYTE   1,11
        033236  001464                         ONE
        033240     003      007                .BYTE   3,7
        033242  001274                         $REG5
        033244     002      001                .BYTE   2,1
        033246  001270                         $REG3
        033250  000004                 DT10:   4
        033252     001      011                .BYTE   1,11
        033254  001464                         ONE
        033256     001      011                .BYTE   1,11
        033260  001462                         ZERO
        033262     003      007                .BYTE   3,7
        033264  001274                         $REG5
        033266     002      001                .BYTE   2,1
        033270  001270                         $REG3
        033272  000002                  DT11:   2
        033274     003     007                  .BYTE   3,7
        033276  031534                          FLAG
        033300     002     002                  .BYTE   2,2
        033302  001270                          $REG3
        033304  000002                  DT12:   2
        033306     006     004                  .BYTE   6,4
        033310  030756                          CALBCC
        033312     006     002                  .BYTE   6,2
        033314  001302                          $TMP2


        033316                          CORMAX:
                000001                  .END
```

```
ABASE  = 000000          266      307
ACDW1  = 000000          266      309
ACDW2  = 000000          266      310
ACPUOP= 000000           266      281
ADDW0  = 000000          266      311
ADDW1  = 000000          266      312
ADDW10= 000000           266      321
ADDW11= 000000           266      322
ADDW12= 000000           266      323
ADDW13= 000000           266      324
ADDW14= 000000           266      325
ADDW15= 000000           266      326
ADDW2  = 000000          266      313
ADDW3  = 000000          266      314
ADDW4  = 000000          266      315
ADDW5  = 000000          266      316
ADDW6  = 000000          266      317
ADDW7  = 000000          266      318
ADDW8  = 000000          266      319
ADDW9  = 000000          266      320
ADEVCT= 000000           266      272
ADEVM  = 000000          266      308
ADRCNT   006057         1410*    1425*    1434#
ADVANC= 104420          1579#    4958     5041
AENV   = 000002            1#     266      277
AENVM  = 000000          266      278
AFATAL= 000000           266      269
AMADR1= 000000           266      294
AMADR2= 000000           266      298
AMADR3= 000000           266      301
AMADR4= 000000           266      304
AMAMS1= 000000           266      288
AMAMS2= 000000           266      296
AMAMS3= 000000           266      299
AMAMS4= 000000           266      302
AMSGAD= 000000           266      274
AMSGLG= 000000           266      275
AMSGTY= 000000           266      268
AMTYP1= 000000           266      289
AMTYP2= 000000           266      297
AMTYP3= 000000           266      300
AMTYP4= 000000           266      303
APASS  = 000000          266      271
APRIOR= 000000           266
APTCSU= 000040          1135     1240#
APTENV= 000001          1128     1196     1238#    1640
APTSIZ= 000200          1237#
APTSPO= 000100          1130     1198     1239#
APT.SI   013510          803     2214#
ASWREG= 000000           266      279
ATESTN= 000000           266      270
AUDONE   003354          840      861      900#
AUNIT  = 000000          266      273
AUSTRT   003126          839#
AUSWR  = 000000          266      280
AUTO.S   012110          801     1958#
```

```
                                                  L10
DZKCE    MACY11 27(1006)  01-JUN-77  10:03  PAGE 108                                                       PAGE:  0128
DZKCE.P11    12-MAY-77 12:23            CROSS REFERENCE TABLE -- USER SYMBOLS

AVECT1= 000000           266      305
AVECT2= 000000           266      306
BASE    013726          2225*    2236     2250*    2260#
BCCLD   030760          3649     3678     3726     3755     3803     3832     3880     3909     5265#
BINWRD  006412          1482*    1485*    1486     1523#
BITCON  031536          5200*    5401*    5403*    5404     5406*    5424*    5426*    5427     5429*    5454#
BIT0  = 000001           129#    1747     1748
BIT00 = 000001           119#     129
BIT01 = 000002           118#     128
BIT02 = 000004           117#     127
BIT03 = 000010           116#     126
BIT04 = 000020           115#     125
BIT05 = 000040           114#     124
BIT06 = 000100           113#     123
BIT07 = 000200           112#     122
BIT08 = 000400           111#     121
BIT09 = 001000           110#     120
BIT1  = 000002           128#     808     1741     1747     1748     2095
BIT10 = 002000           109#    2076
BIT11 = 004000           108#    1071     2668     2702     2747     2806     2859     2912     2965     3019     3079     3109     3139
                        3169     3201     3251     3290     3329     3368     3410     3495     3597     3643     3720     3797     3874
                        3949     4017     4090     4189     4255     4504     4777     4842     4884     4980
BIT12 = 010000           107#    2030     2093
BIT13 = 020000           106#    2033     2079     2097     2362     3541     3555     3570     3593     3616
BIT14 = 040000           105#    1053     2044     2046     2097     2108     3543     3595     4882     4978
BIT15 = 100000           104#     748
BIT2  = 000004           127#     808      949     1748
BIT3  = 000010           126#    2099     2106     3572
BIT4  = 000020           125#    1731     1754     1756     2783     3210     3260     3299     3338     3377     4843     5066     5108
                        5224
BIT5  = 000040           124#    2211     2365     2725     2837     2890     2943     2996     3454     4794
BIT6  = 000100           123#    1736     1737     2101     3118     3178
BIT7  = 000200           122#    1737     1856     2211     2755     3215     3557     3618     5384
BIT8  = 000400           121#    2088     2090     2103     2105     2111     2114     2171
BIT9  = 001000           120#    2085     2111     2114     2169     2171
BM      010631          1779#    2057
BPTVEC= 000014           136#
BRLVL   013442          2166     2176     2184     2197#
BRW     004360           954     1088#
CALBCC  030756          3668     3697     3745     3774     3822     3851     3899     3928     3986     4054     4115     4136     4284
                        4305     4335     4356     4432     4475     4534     4555     4604     4625     4701     4755     4896     4952
                        4992     5038     5240*    5241     5243*    5249     5250*    5251*    5254     5259#    5472
CHAR    030172          3202     3252     3291     3330     3369     3496     5116#
CHARSD  030304          5143#
CHRCNT  006410          1480*    1483     1487     1503*    1521#    1522
CKSWR   011212           779     1094     1586     1804#
CKSWR1  011266          1816#    1828
CKSWR2  011300          1819#
CKSWR3  011304          1821#
CKSWR4  011310          1822#    1830     1837
CKSWR5  011414          1805     1807     1811     1846#
CLKX    001452           341#
CLRIO   031320          3644     3673     3721     3750     3798     3827     3875     3904     5380#
CNERR   011027           873     1779#
CNT.MA  002302           355      628#     747      749      751     1884
CNVRT = 104417           870      979      981      983      985     1578#    1621     1623     1698     1819
```

```
CONERR  010762          868     1779#
CONN    010671         1779#    2035
CONTAB  003330          877      885#
CONVRT= 104416          820      876     1577#    1637     1980
CORMAX  033316         5472#
CR    = 000015           44#    1174     1184
CRC.CC= 102010         5262#
CRC16 = 120001         3646     3675     3723     3752     3800     3829     3877     3906     3954     4022     4109     4278     4329
                       4528     4598     4887     4983     5261#
CREAM   001502          354#     746*    1880*    1881     1883*    1887
CRLF  = 000200           45#    1145     1184
CSR     010363         1779#    1984
CSRMAP  012112         1960#
CTSDLY  031510         5440#
CYCLE   011460          955     1016     1871#
DATABP  006764         1610*    1613     1635     1638#
DATACL= 104413         1574#    2709     2754     2758     2819     2820     2834     2872     2873     2887     2925     2926     2940
                       2978     2979     2993     3037     3040     3083     3113     3143     3173     3204     3254     3293     3332
                       3371     3412     3413     3458     3459     3498     3651     3652     3680     3681     3728     3729     3757
                       3758     3805     3806     3834     3835     3882     3883     3911     3912     3968     3970     4036     4038
                       4102     4116     4137     4155     4162     4164     4199     4271     4285     4306     4336     4357     4375
                       4382     4384     4395     4521     4535     4556     4576     4605     4626     4644     4651     4653     4664
                       4786     4791     4797     4852     5407     5430
DATAHD  006752         1609*    1631     1634#
DDISP = 177570           51#     207#     236
DELAY = 104411         1572#
DEVTAB  003342          843      890#
DH1     032573          404      407      410      446      452      458      485     5472#
DH10    033073          467     5472#
DH2     032614          386      389      392      398      470      479     5472#
DH3     032652          401      449      464     5472#
DH4     032710          419     5472#
DH5     032750          413      416      425      428      482     5472#
DH6     033002          431      434      437      440     5472#
DH7     033050          461     5472#
DISPLA  001242          236#     765*     771*    1083*
DISPRE  000174          197#     771
DSWR  = 177570           50#     206#     235
DT1     033116          405      408      411      447      459      486     5472#
DT10    033250          435      441     5472#
DT11    033272          450     5472#
DT12    033304          453     5472#
DT2     033130          387      390      393      399      471      480     5472#
DT3     033146          402      465     5472#
DT4     033160          420     5472#
DT5     033172          414      417      483     5472#
DT6     033210          426      429     5472#
DT7     033226          432      438     5472#
DZDME = 000000          382
DZDMG = ****** U        382      487
EMTVEC= 000030          139#
EM1     031610          385     5472#
EM10    032113          409     5472#
EM11    032137          412      418      424      430      433     5472#
EM12    032166          415      427      436      439      451     5472#
EM13    032212          421      484     5472#
```

```
DZKCE    MACY11 27(1006)  01-JUN-77  10:03  PAGE 110                                                      PAGE:  0130
DZKCE.P11    12-MAY-77 12:23        CROSS REFERENCE TABLE -- USER SYMBOLS

EM14     032252            442    5472#
EM15     032322            445    5472#
EM16     032343            448    5472#
EM17     032357            454    5472#
EM2      031646            388    5472#
EM20     032423            457    5472#
EM21     032444            460     463    5472#
EM22     032461            466    5472#
EM23     032504            469    5472#
EM24     032526            472    5472#
EM25     032551            475    5472#
EM3      031711            391    5472#
EM4      031755            394    5472#
EM5      032017            397     400     481    5472#
EM6      032047            403     478    5472#
EM7      032066            406    5472#
EOM      031232           4099    4197    4264    4268    4513    4518    4902    4906    4910    4998    5348#
ERCT00   002304            630#
ERCT01   002310            633#
ERCT02   002314            636#
ERCT03   002320            639#
ERCT04   002324            642#
ERCT05   002330            645#
ERCT06   002334            648#
ERCT07   002340            651#
ERCT10   002344            654#
ERCT11   002350            657#
ERCT12   002354            660#
ERCT13   002360            663#
ERCT14   002364            666#
ERCT15   002370            669#
ERCT16   002374            672#
ERCT17   002400            675#
ERR      003244            852     865#    869
ERRMSG   006740           1608*   1626    1629#
ERRPC    003322            871     882#
ERRVEC=  000004            132#   1058    1059*   1061*   1064*
ERTAB0   007112           1623    1663#
EXIT   = 000205            159#
EXITER   007046           1649    1654#
FLAG     031534           5453#   5472
FLOAT    003156            845#    851
FLTDAT   031544           4890    4900    4904    4908    4915    4955    5457#
FY       003202            853#    857     862
GETQI    031104           3688    3692    3765    3769    3842    3846    3919    3923    4044    4048    5303#
GETQO    031072           3659    3663    3736    3740    3813    3817    3890    3894    3976    3980    5293#
GETSI    027754           2823    2827    2876    2880    2929    2933    2982    2986    3043    3047    4119    4123    4140
                          4144    4156    4165    4288    4292    4309    4313    4339    4343    4360    4364    4376    4385
                          4538    4542    4559    4563    4579    4583    4608    4612    4629    4633    4645    4654    5047#
                          5431
GNS    = ****** U         1562    1565    1566    1567    1568    1569    1570    1571    1572    1573    1574    1575    1576
                          1577    1578    1579
HALTS    006770           1593    1640#
HILIM    006052           1407*   1415    1431#
HT     = 000011             42#   1143    1184
INCHAR   011420           1822    1850#
```

```
DZKCE     MACY11 27(1006)  01-JUN-77  10:03  PAGE 111                                                          PAGE:  0131
DZKCE.P11     12-MAY-77 12:23          CROSS REFERENCE TABLE -- USER SYMBOLS

INIFLG  001506              359#    774     804     811*
INLP1   005766             1412#   1421
INPUT = 104415             1576#   1920    1971    1983    1991    2048    2056
INRDY   030614             3414    3460    3499    3507    3515    4200    4213    4218    4398    4412    4417    4441    4455
                           4460    4667    4681    4686    4707    4721    4735    4740    4798    4808    4816    4916    4932
                           4937    5003    5018    5023    5212#
INTTY   013456             2001    2020    2036    2205#
IOTVEC= 000020              137#
KMACTV  001470              348#    788     913*    914    1871    1885    1966*   2126*   2132*   2133*   2137    2162    2228*
                           2229
KMCM    011050              879    1779#
KMCR00  002100              543#
KMCR01  002110              548#
KMCR02  002120              553#
KMCR03  002130              558#
KMCR04  002140              563#
KMCR05  002150              568#
KMCR06  002160              573#
KMCR07  002170              578#
KMCR10  002200              583#
KMCR11  002210              588#
KMCR12  002220              593#
KMCR13  002230              598#
KMCR14  002240              603#
KMCR15  002250              608#
KMCR16  002260              613#
KMCR17  002270              618#
KMCSR   002066              526#    839*    854     860*    889    1019    1086    1701    1755    1889*   1898    1946    2278*
KMCSRH  002070              527#   1736*   1737*   1741*   1747*   1748*   1754*   1756*   1898*   1899*   1900
KMCTL   002072              528#   1900*   1901*   1902
KMNUM   001472              349#    742     996    1964*   1978*   2118*   2119    2127    2129
KMP04   002074              529#   1725*   1731    1768    1773    1902*   1903*   1904    5053    5066    5108    5224    5299*
                           5309*   5310*
KMP06   002076              530#   1742*   1904*   1905*
KMRLVL  002060              523#    989     990*   1907*   1908*   1909
KMRVEC  002056              522#    989*   1022    1890*   1891*   1907
KMS100  002102              544#
KMS101  002112              549#
KMS102  002122              554#
KMS103  002132              559#
KMS104  002142              564#
KMS105  002152              569#
KMS106  002162              574#
KMS107  002172              579#
KMS110  002202              584#
KMS111  002212              589#
KMS112  002222              594#
KMS113  002232              599#
KMS114  002242              604#
KMS115  002252              609#
KMS116  002262              614#
KMS117  002272              619#
KMS200  002104              545#
KMS201  002114              550#
KMS202  002124              555#
KMS203  002134              560#
```

C11

```
KMS204  002144            565#
KMS205  002154            570#
KMS206  002164            575#
KMS207  002174            580#
KMS210  002204            585#
KMS211  002214            590#
KMS212  002224            595#
KMS213  002234            600#
KMS214  002244            605#
KMS215  002254            610#
KMS216  002264            615#
KMS217  002274            620#
KMS300  002106            546#
KMS301  002116            551#
KMS302  002126            556#
KMS303  002136            561#
KMS304  002146            566#
KMS305  002156            571#
KMS306  002166            576#
KMS307  002176            581#
KMS310  002206            586#
KMS311  002216            591#
KMS312  002226            596#
KMS313  002236            601#
KMS314  002246            606#
KMS315  002256            611#
KMS316  002266            616#
KMS317  002276            621#
KMTLVL  002064            525#     991     992*   1911*   1912*
KMTVEC  002062            524#     991*   1909*   1910*   1911
KM.END  002300            623#    1962    2233
KM.MAP  002100            354      542#    746     813     823    1881    1883    1960    1965    2154    2231    2235
LF    = 000012             43#    1178    1184
LINE    010573           1779#    2049
LOBITS  006056           1409*    1433#
LOCK    001444            335#    1085*   1097    1099    1440*   1617    2384*   2400*   2427*   2443*   2470*   2493*   2529*
                         2549*    3640*   3672*   3717*   3749*   3794*   3826*   3871*   3903*
LOKFLG  001510            361#
LOLIM   006050           1406*    1417    1430#
LRC8  = 000200           5260#
LUTYPE= ****** U            1     5232
MASKX   001454            342#
MASTEK  010015           1619     1779#
MCSRX   007745            978     1779#
MDATA   011150           1501     1511    1796#
MEMLIM  001466            347#     937*
MEPASS  007620            977     1779#
MERRPC  010072           1622     1779#
MERRX   007772            984     1779#
MERR2   007645           1779#    2139
MERR3   007672            910     1779#
MESDAT  031540           3606     4091    4097    4190    4195    4256    4262    4266    4322    4397    4440    4505    4511
                         4516     4591    4666    4720    4778    4783    4788    4849    4986    4996    5002    5455#
MESLD   031256           3605     4096    4194    4261    4265    4510    4515    4782    4787    4848    4899    4903    4907
                         4995     5361#
MILK    001504            355#     747*    986    1879*   1884*   1888
```

```
                                                  D11
DZKCE    MACY11 27(1006)  01-JUN-77  10:03  PAGE 113                                              PAGE:  0133
DZKCE.P11     12-MAY-77 12:23             CROSS REFERENCE TABLE -- USER SYMBOLS

MLOCK    007716            951    1779#
MNEW     010017            905    1779#
MODU     010461           1779#   2019
MPASSX   007761            982    1779#
MPFAIL   007562           1697    1714    1779#
MR       007642            958    1779#   1936
MRESET=  004000            159#
MSTCLR=  104410           1571#   1703    2329    2355    2386    2429    2472    2531    2583    2607    2631    2667    2701
                          2746    2805    2858    2911    2964    3018    3078    3108    3138    3168    3199    3249    3288
                          3327    3366    3405    3448    3494    3540    3592    3642    3719    3796    3873    3948    4016
                          4086    4188    4251    4500    4776    4840    4881    4977
MTITLE   001000            205#    778
MTSTN    010003           1620    1779#   1921
MVECX    007753            980    1779#
NEXT     001442            334#   1439    1659    2276*   2302*   2327*   2353*   2383*   2426*   2469*   2528*   2581*   2605*
                          2629*   2665*   2699*   2744*   2803*   2856*   2909*   2962*   3016*   3076*   3106*   3136*   3166*
                          3197*   3247*   3286*   3325*   3364*   3403*   3446*   3492*   3538*   3590*   3639*   3716*   3793*
                          3870*   3946*   4014*   4084*   4186*   4249*   4498*   4774*   4838*   4879*   4975*
NITCH    030004           5053*   5054*   5055*   5056*   5058#
NOACT    010731            790    1779#   1873
NODEV    003240            841     863#
NUM      010323           1779#   1972
OCOR     030006           2708    2753    2816    2869    2922    2975    3036    3967    4035    4100    4198    4269    4519
                          4785    4790    4851    4911    4999    5061#   5067    5097    5139    5155    5196    5289    5375
OK       003220            855     858#    881
ONE      001464            346#   5472
OUTRDY   030140           2807    2812    2860    2865    2913    2918    2966    2971    3023    3027    3032    3055    4095
                          4193    4260    4509    4781    4847    5081    5090    5101#   5126    5135    5148    5152    5172
                          5181    5276    5285    5322    5338    5352    5369
PACT00   002302            629#
PACT01   002306            632#
PACT02   002312            635#
PACT03   002316            638#
PACT04   002322            641#
PACT05   002326            644#
PACT06   002332            647#
PACT07   002336            650#
PACT10   002342            653#
PACT11   002346            656#
PACT12   002352            659#
PACT13   002356            662#
PACT14   002362            665#
PACT15   002366            668#
PACT16   002372            671#
PACT17   002376            674#
PARBIT=  040000            159#
PERFOR=  004537            159#
PFTAB    007324           1698    1720#
PIRQ   = 177772             49#
PIRQVE=  000240            143#
POPRO  = 012600            154#   1653
POP1SP=  005726            152#
POP2SP=  022626            156#
PRIO     010422           1779#   2000
PRIRTY   013730           2221*   2227*   2240    2261#
PRO    = 000000             66#   2197    2198
```

```
                                                  E11
DZKCE   MACY11 27(1006)  01-JUN-77  10:03  PAGE 114                                      PAGE:  0134
DZKCE.P11    12-MAY-77 12:23         CROSS REFERENCE TABLE -- USER SYMBOLS

PR1   = 000040         67#
PR2   = 000100         68#
PR3   = 000140         69#
PR4   = 000200         70#   2199
PR5   = 000240         71#   2200
PR6   = 000300         72#   2201
PR7   = 000340         73#    178     180     182     184    2202
PS    = 177776         46#     47     739*    948*   2166*   2176*
PSW   = 177776         47#
PUSHRO= 010046        153#   1650
PUSH1S= 005746        151#
PUSH2S= 024646        155#
PWRVEC= 000024        138#   1676*   1677*   1686*   1692*   1711*   1712*
QV.FLG  001511        362#    745*    995*
RDCHR = 104402       1292    1565#
RDLIN = 104403       1365    1566#
RDOCT = 104404       1414    1567#
RESREG  006766       1636    1639#
RESVEC= 000010        133#
RESOS = 104407       1570#   1639
ROMCLK= 104412       1573#   1726    1729    1766    1771    2280    2305    2331    2357    2389    2391    2402    2404
                     2432    2434    2445    2447    2478    2480    2499    2501    2536    2538    2554    2556    2584
                     2608    2634    2643    2670    2672    2676    2704    2706    2711    2721    2749    2751    2756
                     2760    2770    2780    2809    2814    2835    2862    2867    2888    2915    2920    2941    2968
                     2973    2994    3025    3029    3034    3057    3084    3114    3144    3174    3206    3216    3218
                     3256    3265    3295    3304    3334    3343    3373    3382    3415    3455    3461    3500    3508
                     3516    3547    3550    3562    3565    3599    3603    3611    3957    3961    3965    3995    4025
                     4029    4033    4063    4201    4214    4219    4228    4399    4413    4418    4427    4442    4456
                     4461    4470    4668    4682    4687    4696    4708    4722    4736    4741    4750    4792    4799
                     4809    4817    4844    4853    4917    4933    4938    4947    5004    5019    5024    5033    5051
                     5064    5079    5083    5086    5092    5095    5106    5124    5128    5131    5137    5150    5153
                     5170    5174    5177    5183    5203    5205    5222    5274    5278    5281    5287    5297    5307
                     5320    5324    5327    5340    5343    5354    5356    5371    5385    5387
RUN     001500        352#    748*   1877*   1878*   1885
SAVACT  001474        350#    908    2137*   2229*
SAVNUM  001476        351#    742*    993*    996*   2130*   2223*
SAVPC   001460        344#    869*    884    1446*   1665
SAVSP   001456        343#
SAVOS = 104406       1569#   1598
SCHAR   030610       3406*   3449*   5164    5182    5187    5191*   5192    5198*   5208#
SCOP1 = 104405       1568#   2399    2412    2442    2455    2489    2510    2545    2563    3671    3700    3748    3777
                     3825    3854    3902    3931
SILOLD  030350       3411    3427    3457    3473    5159#
SIMBCC  030650       3654    3683    3731    3760    3808    3837    3885    3914    3971    4039    4111    4280    4331
                     4530    4600    4892    4988    5232#
SOFTSW  011452       1820    1859#
SOM     031202       4514    5334#
SPACNT= 006411       1481*   1505    1508*   1522#
STACK = 001200         37#    740     927    1660
STAT    001450        340#
STAT1   002050        515#   1892*   2362    3541    3543    3555    3570    3593    3595    3616    4882    4978
STAT2   002052        516#   1893*   2587    2611
STAT3   002054        517#   1894*
STFFCK  031420       5414#
STFFCL  031340       5188    5392#
STKLMT= 177774         48#
```

F11

```
DZKCE    MACY11 27(1006)  01-JUN-77  10:03  PAGE 115                                                              PAGE:  0135
DZKCE.P11     12-MAY-77 12:23          CROSS REFERENCE TABLE -- USER SYMBOLS

STRTSW  001446          339#    780*    783*    784     786     800*    806     808     903     949     956    1914    1937*
                       1967    2150
STUFDT  031564         5463#
STUFLG  030612         3407*   3450*   5185    5199*   5209#
SVOS    006100         1450#
SWFLG   011416          743*   1815*   1842*   1848.
SWR     001240          235#    764*    766     770*    780     908     913    1053    1071    1095    1587    1592    1648
                       1655    1657    1684    1704*   1744    1804    1841*
SWREG   000176          198#    759*    770    1804    1861
SW0    = 000001         101#
SW00   = 000001          91#    101     784     800    1967    2150
SW01   = 000002          90#    100     956    1914    1937
SW02   = 000004          89#     99
SW03   = 000010          88#     98     903
SW04   = 000020          87#     97
SW05   = 000040          86#     96
SW06   = 000100          85#     95    1744
SW07   = 000200          84#     94
SW08   = 000400          83#     93    1655
SW09   = 001000          82#     92    1095
SW1    = 000002         100#
SW10   = 002000          81#   1657
SW11   = 004000          80#
SW12   = 010000          79#   1587
SW13   = 020000          78#   1592
SW14   = 040000          77#
SW15   = 100000          76#
SW2    = 000004          99#
SW3    = 000010          98#
SW4    = 000020          97#
SW5    = 000040          96#
SW6    = 000100          95#
SW7    = 000200          94#
SW8    = 000400          93#
SW9    = 001000          92#
SYNC    030024         3081    3111    3141    3171    5071#
SYNLD   031122         3955    4023    4094    4192    4259    4508    4780    4846    4885    4886    4981    4982    5314#
TBITVE= 000014          134#
TEMP    011106         1488    1752*   1757*   1763*   1775*   1794#   2632*   2638*   2641*   2647*
TIMER = 104414         1575#   2674    3205    3255    3294    3333    3372    3549    3564    3601
TKVEC = 000060          141#
TLAST = 027434         1940    5042#
TPVEC = 000064          142#
TRAPVE= 000034          140#
TRTVEC= 000014          135#
TST1    013732         1047    1927    1945    2274#
TST10   014730         2469    2526#
TST11   015104         2528    2579#
TST12   015146         2581    2603#
TST13   015210         2605    2627#
TST14   015310         2629    2663#
TST15   015410         2665    2697#
TST16   015546         2699    2742#
TST17   015744         2744    2801#
TST2    014006         2276    2300#
TST20   016126         2803    2854#
```

```
DZKCE   MACY11 27(1006)  01-JUN-77  10:03  PAGE 116                                                                          PAGE:  0136
DZKCE.P11     12-MAY-77 12:23              CROSS REFERENCE TABLE -- USER SYMBOLS

TST21    016310           2856    2907#
TST22    016472           2909    2960#
TST23    016654           2962    3014#
TST24    017062           3016    3074#
TST25    017150           3076    3104#
TST26    017240           3106    3134#
TST27    017326           3136    3164#
TST3     014054           2302    2325#
TST30    017416           3166    3195#
TST31    017570           3197    3245#
TST32    017704           3247    3284#
TST33    020022           3286    3323#
TST34    020140           3325    3362#
TST35    020256           3364    3401#
TST36    020406           3403    3444#
TST37    020546           3446    3490#
TST4     014126           2327    2351#
TST40    020706           3492    3536#
TST41    021104           3538    3588#
TST42    021264           3590    3637#
TST43    021600           3639    3714#
TST44    022114           3716    3791#
TST45    022430           3793    3868#
TST46    022744           3870    3944#
TST47    023202           3946    4012#
TST5     014220           2353    2381#
TST50    023440           4014    4082#
TST51    023772           4084    4184#
TST52    024174           4186    4247#
TST53    025274           4249    4496#
TST54    026474           4498    4772#
TST55    026726           4774    4836#
TST56    027044           4838    4877#
TST57    027434           4879    4973#   5042
TST6     014360           2383    2424#
TST60 =  ****** U         4975
TST7     014520           2426    2467#
TTST     004146            952*    954*   1049#
TWOSYN=  010000            159#
TYPDAT   006754           1614    1632    1635#
TYPE  =  104401            778     790     812     867     872     878     905     910     951     958     977     978     980
                           982     984    1148    1299    1305    1310    1314    1319    1320    1322    1325    1329    1394
                          1396    1412    1419    1420    1471    1511    1562#   1615    1616    1619    1620    1622    1624
                          1628    1633    1697    1713    1818    1821    1873    1919    1936    1942    1979    1999    2013
                          2018    2027    2034    2041    2139
TYPMSG   006654           1612    1615#
VEC      010401           1779#   1992
VECMAP   013172           2138    2150#
VECTR    013724           2220*   2226*   2245    2251*   2259#
WHAT     005770           1405*   1413#
WHERE    006054           1408*   1422    1432#
WHICH    013164           1981    2146#
WRDCNT   006406           1479*   1512*   1520#
WRKO.F   006742           1627    1630#
XBX      006542           1588    1590    1592#
XCSR     004104            979    1017#
```

```
DZKCE   MACY11 27(1006)  01-JUN-77  10:03  PAGE 117                                                          PAGE:  0137
DZKCE.P11    12-MAY-77 12:23            CROSS REFERENCE TABLE -- USER SYMBOLS

XERR     004126             985    1026#
XHEAD    010077             812    1779#
XPASS    004120             983    1023#
XPOLY    030754            3646*   3675*   3723*   3752*   3800*   3829*   3877*   3906*   3954*   4022*   4109*   4278*   4329*
                           4528*   4598*   4887*   4983*   5248    5250    5258#
XSTATQ   011060             821    1779#
XTSTN    007120            1621    1666#
XVEC     004112             981    1020#
ZERO     001462             345#   5472
$APTHD   002034             497     503#
$ASTAT=  ****** U          1218    1233
$ATYC    004722            1189    1191#
$ATY1    004676            1187#
$ATY3    004704            1133    1188#
$ATY4    004714            1190#   1643
$AUTOB   001234             232#
$BASE    001372             307#   2225
$BDADR   001222             227#
$BDDAT   001226             229#
$CDW1    001376             309#   2222
$CDW2    001400             310#
$CHARC   004672            1150*   1160*   1167    1176*   1181#
$CKSWR=  ****** U          1565
$CMTAG   001200             215#
$CM1   = 000006             247#    248#    249#    250#    251#    252#    253#
$CM2   = 000014             247#    248#    249#    250#    251#    252#    253#
$CM3   = 000006             245#    247
$CM4   = 000005             253#    254#    255#    256#    257#    258#
$CNTLG   005534            1340#
$CNTLU   005527            1314    1339#
$COD   = ****** U             1
$CPUOP   001344             281#
$CRAP  = 177777               1#   2263#   2266    2269#   2289#   2292    2295#   2314#   2317    2320#   2340#   2343    2346#
                           2370#   2373    2376#   2413#   2416    2419#   2456#   2459    2462#   2515#   2518    2521#   2568#
                           2571    2574#   2592#   2595    2598#   2616#   2619    2622#   2651#   2654    2658#   2685#   2688
                           2692#   2730#   2733    2737#   2788#   2791    2796#   2841#   2844    2849#   2894#   2897    2902#
                           2947#   2950    2955#   3000#   3003    3009#   3063#   3066    3069#   3093#   3096    3099#   3123#
                           3126    3129#   3153#   3156    3159#   3183#   3186    3190#   3234#   3237    3240#   3273#   3276
                           3279#   3312#   3315    3318#   3351#   3354    3357#   3390#   3393    3396#   3431#   3434    3439#
                           3477#   3480    3485#   3524#   3527    3531#   3577#   3580    3583#   3625#   3628    3632#   3702#
                           3705    3709#   3779#   3782    3786#   3856#   3859    3863#   3933#   3936    3939#   4001#   4004
                           4007#   4069#   4072    4077#   4172#   4175    4179#   4231#   4234    4242#   4479#   4482    4491#
                           4759#   4762    4767#   4825#   4828    4831#   4862#   4865    4872#   4959#   4962    4968#
$CRLF    001313             260#   1149    1184    1319    1339    1399    1420    1471    1615    1616    1624    1919    1979
$DDW0    001402             311#   2230
$DDW1    001404             312#
$DDW10   001426             321#
$DDW11   001430             322#
$DDW12   001432             323#
$DDW13   001434             324#
$DDW14   001436             325#
$DDW15   001440             326#
$DDW2    001406             313#
$DDW3    001410             314#
$DDW4    001412             315#
$DDW5    001414             316#
```

```
DZKCE   MACY11 27(1006)  01-JUN-77  10:03  PAGE 118                                                    PAGE:  0138
DZKCE.P11     12-MAY-77 12:23            CROSS REFERENCE TABLE -- USER SYMBOLS

$DDW6    001416                 317#
$DDW7    001420                 318#
$DDW8    001422                 319#
$DDW9    001424                 320#
$DEVCT   001326                 272#
$DEVM    001374                 308#    2228
$DOAGN   004100                 994     1003     1008     1014#
$ENDAD   004070                 191      776     1010#    1646
$ENDCT   004054                1005#
$ENV     001336                 277#     757      793      795     1128     1196     1220     1640     1806
$ENVM    001337                 278#     798     1130     1135     1198
$EOP     003662                 973#    4975
$EOPCT   004046                1002#    1006
$ERFLG   001203                 218#     744*     976*    1038     1067     1069*    1090     1597*    1611     1625*    1699*
$ERMAX   001215                 224#    1090
$ERROR   006512                 181     1586#
$ERRPC   001216                 225#     753*     997*    1046*    1594     1596*    1700*
$ERRTB   001512                 379#    1607
$ERTTL   001212                 222#     988     1028     1654*    1896*
$ETABL   001336                 276#
$ETEND   001442                 329#     509
$FATAL   001320                 269#    1224*
$FFLG    005142                1187*    1190*    1218     1227*    1235#
$FILLC   001256                 243#    1153     1184
$FILLS   001255                 242#    1184
$GDADR   001220                 226#
$GDDAT   001224                 228#
$GET42   004060                1007#
$GTSWR=  ****** U              1564
$HD    = 000000                  11
$HIBTS   002034                 504#
$HIOCT   005722                1387*    1398#
$ICNT    001204                 219#    1075*    1076     1078*    1089
$ILLUP   007316                1676     1692     1716#
$INPUT   005724                1403#    1576
$INTAG   001235                 233#
$ITEMB   001214                 223#    1602*    1642
$LF      001314                 261#    1184     1329     1339     1399
$LFLG    005141                1228*    1234#
$LPADR   001206                 220#     755*     955*     959     1082*    1084     1089     1659*    1661     1935*    1945*    1947
$LPERR   001210                 221#
$MADR1   001350                 294#
$MADR2   001354                 298#
$MADR3   001360                 301#
$MADR4   001364                 304#
$MAIL    001316                 267#     505      509     1081     1128
$MAMS1   001346                 288#    2224
$MAMS2   001352                 296#
$MAMS3   001356                 299#
$MAMS4   001362                 302#
$MBADR   002036                 505#
$MFLG    005140                1188*    1194     1229*    1233#
$MNEW    005552                1343#    1821
$MSGAD   001332                 274#    1204*    1207
$MSGLG   001334                 275#    1209*
$MSGTY   001316                 268#    1202     1210*    1222     1226*
```

```
                                                 J11
DZKCE    MACY11 27(1006)  01-JUN-77  10:03  PAGE 119                                                        PAGE:  0139
DZKCE.P11     12-MAY-77 12:23            CROSS REFERENCE TABLE -- USER SYMBOLS

$MSWR     005541         1341#    1818
$MTYP1    001347          289#
$MTYP2    001353          297#
$MTYP3    001357          300#
$MTYP4    001363          303#
$MXCNT    004362         1079     1089#
$N      = 000057            1#    2263     2269     2271     2278#    2289     2295     2297     2304#    2314     2320     2322     2329
                         2330#    2340     2346     2348     2355     2356#    2370     2376     2378     2386     2387#    2413     2419
                         2421     2429     2430#    2456     2462     2464     2472     2473#    2515     2521     2523     2531     2532#
                         2568     2574     2576     2583     2584#    2592     2598     2600     2607     2608#    2616     2622     2624
                         2631     2632#    2651     2658     2660     2667     2668#    2685     2692     2694     2701     2702#    2730
                         2737     2739     2746     2747#    2788     2796     2798     2805     2806#    2841     2849     2851     2858
                         2859#    2894     2902     2904     2911     2912#    2947     2955     2957     2964     2965#    3000     3009
                         3011     3018     3019#    3063     3069     3071     3078     3079#    3093     3099     3101     3108     3109#
                         3123     3129     3131     3138     3139#    3153     3159     3161     3168     3169#    3183     3190     3192
                         3199     3200#    3234     3240     3242     3249     3250#    3273     3279     3281     3288     3289#    3312
                         3318     3320     3327     3328#    3351     3357     3359     3366     3367#    3390     3396     3398     3405
                         3406#    3431     3439     3441     3448     3449#    3477     3485     3487     3494     3495#    3524     3531
                         3533     3540     3541#    3577     3583     3585     3592     3593#    3625     3632     3634     3642     3643#
                         3702     3709     3711     3719     3720#    3779     3786     3788     3796     3797#    3856     3863     3865
                         3873     3874#    3933     3939     3941     3948     3949#    4001     4007     4009     4016     4017#    4069
                         4077     4079     4086     4087#    4172     4179     4181     4188     4189#    4231     4242     4244     4251
                         4252#    4479     4491     4493     4500     4501#    4759     4767     4769     4776     4777#    4825     4831
                         4833     4840     4841#    4862     4872     4874     4881     4882#    4959     4968     4970     4977     4978#
                         5042#
$NULL     001254          241#    1155     1184
$NWTST=   000000         2273#    2299#    2324#    2350#    2380#    2423#    2466#    2525#    2578#    2602#    2626#    2662#    2696#
                         2741#    2800#    2853#    2906#    2959#    3013#    3073#    3103#    3133#    3163#    3194#    3244#    3283#
                         3322#    3361#    3400#    3443#    3489#    3535#    3587#    3636#    3713#    3790#    3867#    3943#    4011#
                         4081#    4183#    4246#    4495#    4771#    4835#    4876#    4972#
$OVER     004334         1051     1054     1065     1077     1083#
$PASS     001324          271#     975*     987      999*    1000*    1017     1025     1073     1090     1895*
$PASTM    002042          507#
$PWRDN    007126          179      741     1676#    1711
$PWRMG    007312         1714#
$PWRUP    007200         1686     1692#
$QUES     001312          259#    1184     1322     1339     1396     1399     1419     1942     2014     2028     2042
$RDCHR    005144         1259#    1565
$RDDEC=   ****** U       1568
$RDLIN    005264         1287#    1566
$RDOCT    005564         1360#    1567
$RDSZ =   000007         1280#
$REGAD    001260          245#
$REG0     001262          247#    1455*    1460     5472
$REG1     001264          248#     875*     887     1454*    1461
$REG2     001266          249#    1453*    1462     5472
$REG3     001270          250#    1452*    1463     5472
$REG4     001272          251#    1451*    1464     5472
$REG5     001274          252#    1450*    1465     5472
$RTNAD    004102         1016#
$R2A  =   ****** U       1568
$S    =   000061            1#    2276     2278#    2302     2304#    2327     2330#    2353     2356#    2383     2387#    2426     2430#
                         2469     2473#    2528     2532#    2581     2584#    2605     2608#    2629     2632#    2665     2668#    2699
                         2702#    2744     2747#    2803     2806#    2856     2859#    2909     2912#    2962     2965#    3016     3019#
                         3076     3079#    3106     3109#    3136     3139#    3166     3169#    3197     3200#    3247     3250#    3286
                         3289#    3325     3328#    3364     3367#    3403     3406#    3446     3449#    3492     3495#    3538     3541#
```

K11

```
DZKCE   MACY11 27(1006)  01-JUN-77  10:03  PAGE 120                                                     PAGE:  0140
DZKCE.P11    12-MAY-77 12:23          CROSS REFERENCE TABLE -- USER SYMBOLS

                        3590     3593#    3639     3643#    3716     3720#    3793     3797#    3870     3874#    3946     3949#    4014
                        4017#    4084     4087#    4186     4189#    4249     4252#    4498     4501#    4774     4777#    4838     4841#
                        4879     4882#    4975     4978#
$SAVRE= ****** U        1568
$SAVR6  007322          1685*    1693     1694*    1695*    1718#
$SCOPE  004134           177     1045#    2194
$SETUP= 000000           998     1046     1248     1345
$SVLAD  004316          1062     1080#
$SVPC = 000040           189#     194
$SWR  = 164000             1#      11      258      259      970      998     1009     1015     1017     1039     1040     1041     1042
                        1053     1065     1067     1068     1069     1070     1071     1083     1089     1715     2275     2301     2326
                        2352     2382     2425     2468     2527     2580     2604     2628     2664     2698     2743     2802     2855
                        2908     2961     3015     3075     3105     3135     3165     3196     3246     3285     3324     3363     3402
                        3445     3491     3537     3589     3638     3715     3792     3869     3945     4013     4083     4185     4248
                        4497     4773     4837     4878     4974
$SWREG  001340           279#     759
$SWRMK= 000000          1042
$TESTN  001322           270#    1081*
$TIMES  001310           258#     998*    1070*    1076     1079*    1089
$TKB    001246           238#    1052     1246     1263     1269     1808     1810     1852     2207
$TKS    001244           237#    1050     1246     1261     1267     1850     2205     5446
$TMP0   001276           253#     814*    1781     2162*    2163*    5076*    5088*    5122*    5133*    5168*    5179*    5220*    5226*
                        5238*    5252*    5272*    5283*    5318*    5329*    5368*    5373*
$TMP1   001300           254#     815*    1783     4429*    4430*    4431     4472*    4473*    4474     4698*    4699*    4700     4752*
                        4753*    4754     4949*    4950*    4951     5035*    5036*    5037     5120*    5136     5146*    5149     5163*
                        5167*    5194*    5201*    5239*    5244*    5270*    5286
$TMP2   001302           255#     817*     838*     865      874*    1785     1975     1978     2065*    4216*    4217*    4415*    4416*
                        4431*    4432     4458*    4459*    4474*    4475     4684*    4685*    4700*    4701     4738*    4739*    4754*
                        4755     4935*    4936*    4951*    4952     5021*    5022*    5037*    5038     5398*    5408*    5421*    5434*
                        5472
$TMP3   001304           256#     818*    1787     1987     1990     1995     1998     2052     2055     2060     2063
$TMP4   001306           257#     819*    1789     1970*    1982*    2148     5104*    5110*
$TN   = 000060             1#      11     2273     2275#    2299     2301#    2324     2326#    2350     2352#    2380     2382#    2423
                        2425#    2466     2468#    2525     2527#    2578     2580#    2602     2604#    2626     2628#    2662     2664#
                        2696     2698#    2741     2743#    2800     2802#    2853     2855#    2906     2908#    2959     2961#    3013
                        3015#    3073     3075#    3103     3105#    3133     3135#    3163     3165#    3194     3196#    3244     3246#
                        3283     3285#    3322     3324#    3361     3363#    3400     3402#    3443     3445#    3489     3491#    3535
                        3537#    3587     3589#    3636     3638#    3713     3715#    3790     3792#    3867     3869#    3943     3945#
                        4011     4013#    4081     4083#    4183     4185#    4246     4248#    4495     4497#    4771     4773#    4835
                        4837#    4876     4878#    4972     4974#
$TPB    001252           240#    1173*    1184     1591*    1855*    2210*
$TPFLG  001257           244#    1122     1184
$TPS    001250           239#    1171     1184     1589     1853     2208
$TRAP   006414           183     1539#
$TRAP2  006436          1550#    1561
$TRP  = 000021          1554#    1563#    1565     1566#    1567#    1568#    1569#    1570#    1571#    1572#    1573#    1574#    1575#
                        1576#    1577#    1578#    1579#    1580#
$TRPAD  006450          1544     1561#
$TSTM   002040           506#
$TSTNM  001202           217#     754*    1038     1080*    1081     1083     1090     1668     1722     1924     1931     1933     2275*
                        2301*    2326*    2352*    2382*    2425*    2468*    2527*    2580*    2604*    2628*    2664*    2698*    2743*
                        2802*    2855*    2908*    2961*    3015*    3075*    3105*    3135*    3165*    3196*    3246*    3285*    3324*
                        3363*    3402*    3445*    3491*    3537*    3589*    3638*    3715*    3792*    3869*    3945*    4013*    4083*
                        4185*    4248*    4497*    4773*    4837*    4878*    4974*
$TTYIN  005520          1289     1290     1302     1320     1334     1338#
$TYPBN= ****** U        1563
```

```
$TYPDS= ****** U        1563
$TYPE   004414          1122#    1215     1554     1562
$TYPEC  004626          1152     1159     1166     1171#    1172
$TYPEX  004674          1177     1179     1182#
$TYPOC= ****** U        1563
$UNIT   001330           273#
$UNITM  002044           508#
$USWR   001342           280#
$VECT1  001366           305#    2226     2227
$VECT2  001370           306#
$XTSTR  004174          1048     1056#
$Y    = 000000             1#     510#
$$GET4= 000000          1009#
$40CAT= ****** U        1053
.     = 033316           171#     173      176#     189      190#     192#     194#     196#     200#     204#     214#     262      348#
                         349#     350#     351#     360#     487#     493      494#     496#     498#     541#     543#     544#     545#
                         546#     548#     549#     550#     551#     553#     554#     555#     556#     558#     559#     560#     561#
                         563#     564#     565#     566#     568#     569#     570#     571#     573#     574#     575#     576#     578#
                         579#     580#     581#     583#     584#     585#     586#     588#     589#     590#     591#     593#     594#
                         595#     596#     598#     599#     600#     601#     603#     604#     605#     606#     608#     609#     610#
                         611#     613#     614#     615#     616#     618#     619#     620#     621#     782      792      899#     912
                        1017     1089     1090     1184     1236#    1246     1338#    1339     1345#    1399     1688     1717     1760#
                        1795#    1797#    1851     1854     1875     1968     2009     2092     2096     2142     2173     2206     2209
                        2363     3542     3556     3571     3594     3617     4166     4386     4655
.ADVAN  006060          1439#    1579
.BEGIN  003464           927#
.CNVRT  006170          1472#    1578
.CONVR  006164          1471#    1577
.DATAC  007446          1574     1751#
.DELAY  007332          1572     1724#
.MSTCL  007362          1571     1735#
.RESOS  006132          1460#    1570
.ROMCL  007400          1573     1740#
.SAVOS  006072          1446#    1569
.SCOP1  004364          1094#    1568
.START  002402           201      739#     755     1838
.TIMER  007512          1575     1762#
.$ASTA= ****** U        1188     1191
.$X   = 002034           493#     498
```

M11

```
DZKCE    MACY11 27(1006)  01-JUN-77  10:03  PAGE 123                                                    PAGE:  0142
DZKCE.P11     12-MAY-77 12:23           CROSS REFERENCE TABLE -- MACRO NAMES

COMMEN    144#
ENDCOM    144#
ERROR      38#    2287    2312    2338    2368    2398    2411    2441    2454    2488    2509    2544    2562    2590    2614
         2640     2649    2683    2718    2728    2767    2777    2786    2825    2829    2839    2878    2882    2892    2931
         2935     2945    2984    2988    2998    3045    3049    3091    3121    3151    3181    3213    3225    3232    3263
         3271     3302    3310    3341    3349    3380    3388    3421    3467    3506    3514    3522    3560    3575    3623
         3661     3665    3690    3694    3738    3742    3767    3771    3815    3819    3844    3848    3892    3896    3921
         3925     3978    3982    4046    4050    4121    4125    4142    4146    4158    4167    4207    4226    4290    4294
         4311     4315    4341    4345    4362    4366    4378    4387    4405    4425    4434    4448    4468    4477    4540
         4544     4561    4565    4581    4585    4610    4614    4631    4635    4647    4656    4674    4694    4703    4714
         4728     4748    4757    4796    4805    4815    4823    4860    4923    4945    4954    5010    5031    5040    5112
         5228     5433
ESCAPE    144#
GETPRI    144#
GETSWR    144#
HLT       157#
KMEND       1#     960
KMFRNT      1#
MULT      144#
NEWTST    144#    2273    2299    2324    2350    2380    2423    2466    2525    2578    2602    2626    2662    2696    2741
         2800     2853    2906    2959    3013    3073    3103    3133    3163    3194    3244    3283    3322    3361    3400
         3443     3489    3535    3587    3636    3713    3790    3867    3943    4011    4081    4183    4246    4495    4771
         4835     4876    4972
POP       144#    1230    1231    1388    1704    1705    2253
PUSH      144#    1191    1193    1214    1362    1678    1684    2216
REPORT      1#     144#
SCOPE      39#    2274    2300    2325    2351    2381    2424    2467    2526    2579    2603    2627    2663    2697    2742
         2801     2854    2907    2960    3014    3074    3104    3134    3164    3195    3245    3284    3323    3362    3401
         3444     3490    3536    3588    3637    3714    3791    3868    3944    4012    4082    4184    4247    4496    4772
         4836     4877    4973
SETPRI    144#
SETTRA   1554#    1565    1566    1567    1568    1569    1570    1571    1572    1573    1574    1575    1576    1577    1578
         1579
SETUP     144#
SKIP      144#
SLASH     144#
SPACE     144#
STARS     144#     187     210     262     265     490     492     499     968    1035    1107    1186    1245    1251    1280
         1348     1533    1674    1690    2273    2299    2324    2350    2380    2423    2466    2525    2578    2602    2626
         2662     2696    2741    2800    2853    2906    2959    3013    3073    3103    3133    3163    3194    3244    3283
         3322     3361    3400    3443    3489    3535    3587    3636    3713    3790    3867    3943    4011    4081    4183
         4246     4495    4771    4835    4876    4972
SWRSU     144#
TRMTRP   1554#
TYPBIN    144#
TYPDEC    144#
TYPNAM    144#
TYPNUM    144#
TYPOCS    144#
TYPOCT    144#
TYPTXT    144#
$ABORT      1#
$AUTO       1#     824
$BCC        1#    3625    3702    3779    3856
$BINCR      1#    3933    4001
$BINWI      1#    3000
```

```
$BUFFE      1#    1791
$CDATA      1#    4862    4959
$CLOCK      1#    2616
$COMP       1#    3208    3220    3227    3258    3267    3297    3306    3336    3345    3375    3384    3502    3510    3518
          3552    3567    4221    4420    4463    4689    4743    4940    5026
$CRC        1#    3644    3673    3721    3750    3798    3827    3875    3904
$CRCSH      1#    3949    4017
$CYCLE      1#    1862
$EMPTY      1#    4759
$EOP        1#     960
$ERTBL      1#     381
$FINI       1#    5042
$FLAG       1#
$FLOAT      1#    2474    2494    2533    2550
$GETPA      1#
$HALF       1#    4825
$HEADE      1#      11
$INACT      1#    3063    3093    3123    3153
$INIT       1#
$LINE1      1#    2456    2515
$LU1        1#    2263    2289    2314    2340
$LU12       1#    2370
$LU17       1#    2413
$MARHI      1#
$MARK       1#    3477
$MATCH      1#    4210    4408    4451    4677    4731    4928    5014
$MOCK       1#
$MODEM      1#    3524
$MSG        1#    1779
$MULT       1#
$PASEN      1#     974
$PATTE      1#    3390    3431
$PFAIL      1#    1669
$QOQI       1#    5293    5303
$QUEST      1#    1971    1983    1991    2048    2056
$RAMCL      1#    1723
$RCLK       1#    1726    1729    1766    1771    2280    2305    2331    2357    2389    2391    2402    2404    2432    2434
          2445    2447    2478    2480    2499    2501    2536    2538    2554    2556    2584    2608    2634    2643    2670
          2672    2676    2704    2706    2711    2721    2749    2751    2756    2760    2770    2780    2809    2814    2835
          2862    2867    2888    2915    2920    2941    2968    2973    2994    3025    3029    3034    3057    3084    3114
          3144    3174    3206    3216    3218    3256    3265    3295    3304    3334    3343    3373    3382    3415    3455
          3461    3500    3508    3516    3547    3550    3562    3565    3599    3603    3610    3957    3961    3965    3995
          4025    4029    4033    4063    4201    4214    4219    4228    4399    4413    4418    4427    4442    4456    4461
          4470    4668    4682    4687    4696    4708    4722    4736    4741    4750    4792    4799    4809    4817    4844
          4853    4917    4933    4938    4947    5004    5019    5024    5033    5051    5064    5079    5083    5086    5092
          5095    5106    5124    5128    5131    5137    5150    5153    5170    5174    5177    5183    5203    5205    5222
          5274    5278    5281    5287    5297    5307    5320    5324    5327    5340    5343    5354    5356    5371    5385
          5387
$RCRC       1#    4172
$REC        1#    3183    3234    3273    3312    3351
$ROVAR      1#     331
$SCADD      1#    1046
$SCAD1      1#    1085
$SIMBC      1#    5232
$SINAC      1#
$SOFTC      1#    1799
```

```
DZKCE    MACY11 27(1006)  01-JUN-77  10:03  PAGE 125                                                    PAGE:  0144
DZKCE.P11     12-MAY-77 12:23          CROSS REFERENCE TABLE -- MACRO NAMES

$STUFF      1#
$SWPAC      1#    2568    2592
$TCHAR      1#    4090    4189    4255    4504
$TCRC       1#    4069    4231    4479
$TRANW      1#    4109    4278    4329    4528    4598
$TRAN1      1#    2651    2685    2730
$TSTN       1#    2271    2297    2322    2348    2378    2421    2464    2523    2576    2600    2624    2660    2694    2739
          2798    2851    2904    2957    3011    3071    3101    3131    3161    3192    3242    3281    3320    3359    3398
          3441    3487    3533    3585    3634    3711    3788    3865    3941    4009    4079    4181    4244    4493    4769
          4833    4874    4970
$UPADD      1#    1697
$VARIA      1#     203
$WINDO      1#    2788    2841    2894    2947
$XZ         1#    2263    2269    2289    2295    2314    2320    2340    2346    2370    2376    2413    2419    2456    2462
          2515    2521    2568    2574    2592    2598    2616    2622    2651    2658    2685    2692    2730    2737    2788
          2796    2841    2849    2894    2902    2947    2955    3000    3009    3063    3069    3093    3099    3123    3129
          3153    3159    3183    3190    3234    3240    3273    3279    3312    3318    3351    3357    3390    3396    3431
          3439    3477    3485    3524    3531    3577    3583    3625    3632    3702    3709    3779    3786    3856    3863
          3933    3939    4001    4007    4069    4077    4172    4179    4231    4242    4479    4491    4759    4767    4825
          4831    4862    4872    4959    4968
$ZEROS      1#
$$CMRE    208#     247     248     249     250     251     252
$$CMTM    208#     253     254     255     256     257
$$ESCA    144#
$$NEWT    144#    2273    2299    2324    2350    2380    2423    2466    2525    2578    2602    2626    2662    2696    2741
          2800    2853    2906    2959    3013    3073    3103    3133    3163    3194    3244    3283    3322    3361    3400
          3443    3489    3535    3587    3636    3713    3790    3867    3943    4011    4081    4183    4246    4495    4771
          4835    4876    4972
$$SCOP      1#    1029
$$SET    1554#    1565    1566    1567    1568    1569    1570    1571    1572    1573    1574    1575    1576    1577    1578
          1579
$$SKIP    144#
.EQUAT      1#      34
.HEADE      1#
.SETUP      1#
.$ACT1      1#     185
.$APTB      1#     263#
.$APTH      1#     488
.$APTY      1#    1184
.$CATC      1#
.$CMTA      1#     208
.$EOP       1#     966
.$ERRO      1#
.$ERRT      1#
.$POWE      1#    1672
.$RDOC      1#    1346
.$READ      1#    1243
.$SCOP      1#    1033
.$TRAP      1#    1531
.$TYPE      1#    1105
.$TYPO      1#

. ABS.   033316      000
```

C12

```
 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DZKCE0,DZKCE/SOL/CRF+DZKCE.MAC,DZKCE.P11/EQ:DZDME
 RUN-TIME: 30 27 2 SECONDS
 RUN-TIME RATIO: 2064/60=34.3
 CORE USED:  53K  (106 PAGES)
```