DZ11

8 LINE ASYNCH MUX TESTS

MD-11-DZDZA-C

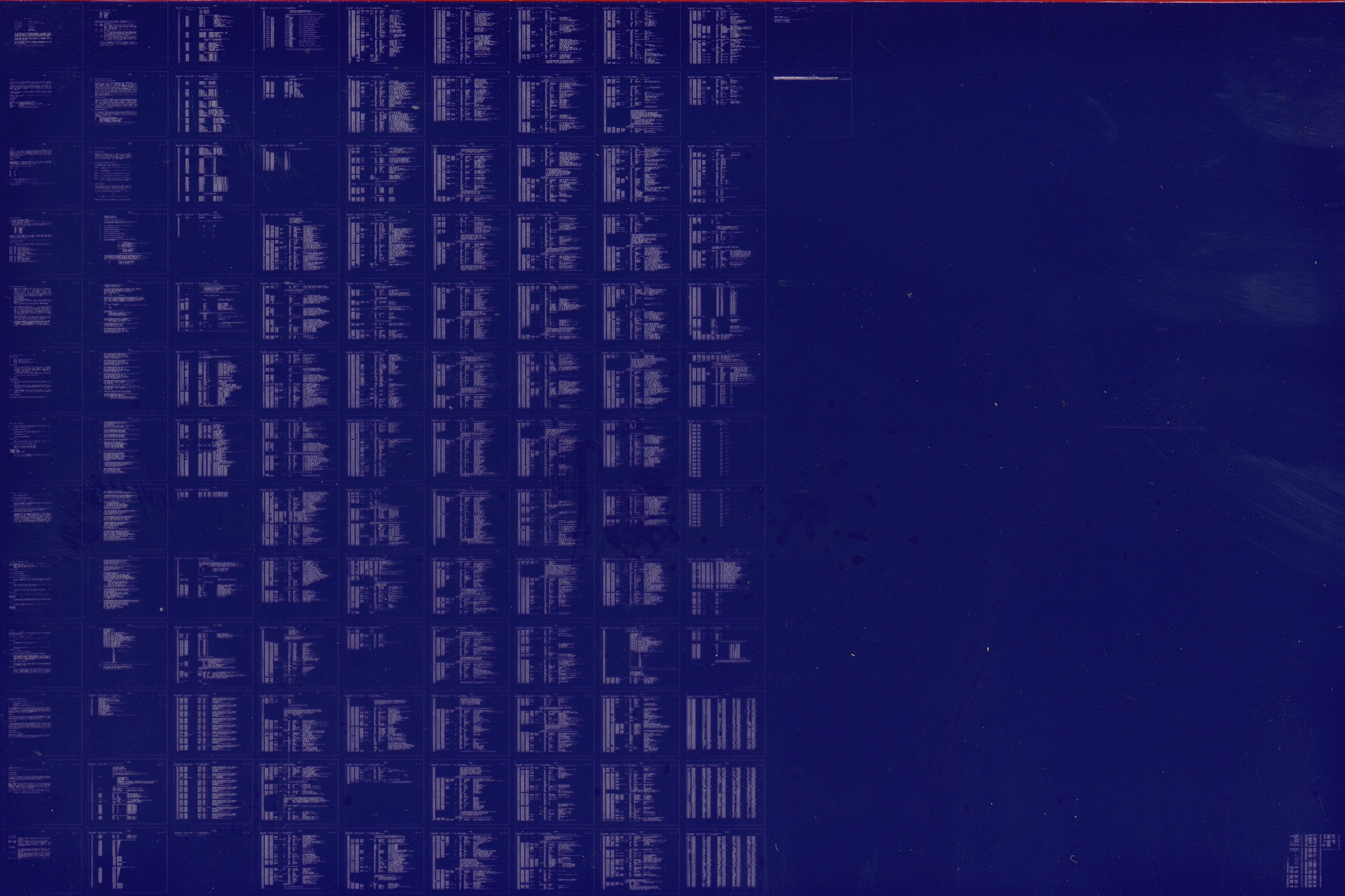
EP-DZDZA-C-DL-B

FICHE 1 OF 1

DEC 1976

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZDZA-C-D
PRODUCT NAME:	DZ11 8 LINE ASYNC MUX TESTS
DATE RELEASED:	DEC 1976
MAINTAINER:	DIAGNOSTICS
AUTHORS:	JOHN EGOLF JERRYL PAYNE

1. ABSTRACT

The function of the DZ11 diagnostics is to verify the option operates according to specifications. The diagnostics also verify that the DZ11 operates in its environment such as the system in which it is installed.

Parameters may be supplied to the program by either 'AUTO SIZING' or input from the user on the console by having SW00=1 at start time. Auto sizing will be done only the first time the program is started and SW07=0 and SW00=0 and SW03=0. Console input may be done at any start time if SW00=1.

Currently there is one standalone diagnostic (DZDZA), one system module for DEC X/11 (DZAA), and there are plans for an online overlay for DZITA (ITEP) - DZDZB.

DZDZA will test all parts of the DZ11 such as cables, dist pnl., and the interface module itself.

2. REQUIREMENTS

2.1 EQUIPMENT

```
Any PDP11 family CPU (WITH MINIMUM 8K MEMORY)
ASR 33 (or equilivalent for console)
DZ11            INTERFACE MODULE (M7819(EIA), M7814(20MA))
H327            Staggered turnaround connector.  (if        ↑B   PARITY
and BREAK are to be tested.
H325            Cable turnaround and dist pnl testing.
H315            This may be subsituted for H325.
```

2.2 STORAGE

Program will use all 8K of memory except where ABL and BOOTSTRAP LOADER reside. Location 1500 thru 2000 are especially to be noted and to be untouched by operator after parameters have been input from console (SW00=1); or after the 'AUTO SIZING' has been done. These locations may be t8changed if the user understands their meaning and different parameters are required.

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 leave or set SW00=1 for user
↑B input from console terminal.
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 or parameters were changed by SW00=1) and also the following:

```
'MAP OF DZ11 STATUS'
1500     160010
1502     000300
1504     000005
1506     000377
1510     017470
1512     000000
```

The above is only an example! This would indicate the status table starting at add. 1500 in the program. ↑BTHE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. For information of status table see section 8.4 for help.

The program will type "Running" and proceed to run the diagnostic.

4.1 CONTROL SWITCH SETTINGS

NOTE: If there is no real SWR (177570); SWR may be modified at Loc:176 or by hitting Control "G" (↑G) on console terminal.

```
SW 15   Set:  Halt on error
SW 14   Set:  Loop on current test
SW 13   Set:  Inhibit error print out
SW 12   Set:  Inhibit **ALL** type out/bell on error.
SW 11   Set:  Inhibit iterations.  (quick pass)
SW 10   Set:  Escape to next test
SW 09   Set:  Loop with current data
SW 08   Set:  Catch error and loop on it
SW 07   Set:  NO AUTO SIZE;  CLR-do AUTO SIZE.  If 1st start of  program
              after loading.
SW 06   Set:  Reselect DZ11's desired active
SW 05   Set:  Reserved
SW 04   Set:  SELECT DELAY PARAMETER
SW 03   Set:  Extra parameter input
SW 02   Set:  Lock on selected test
SW 01   Set:  Restart program at selected test
SW 00   Set:  Get users parameters from console
```

```
4.1.2   SWITCH REGISTER RESTRICTIONS

        SW 06    RESELECT DZ11'S DESIRED ACTIVE.  please note that a  message  is
                 typed  out  for  setting  the  switch  register  equal to DZ11's
                 active.   this  means  if  the  system  has  four  DZ11s;  bits
                 00,01,02,03  will  be  set  in  loc  'DZACTV'  from  the  switch
                 register.      Using     this     switch(SW06)      alters     that
                 location;therefore  if four DZ11s are in the system ***DO NOT***
             ↑B    set switches greater than SW 03 in the up position.  This  would
                 be a fatal error.  do not select more active DZ11s than has been
                 given information about in parameter input (SW00=1)
METHOD: A:       Load address 200
        B:       Start with SW 06=1
        C:       Program will type message
        D:       Set the BINARY number of  DZ11s  desired  active  EXAMPLE:   1=1
                 DZ11;   3=2  DZ11;   7=3 DZ11;  17=4 DZ11 37=5 DZ11 etc/aa PRESS
                 CONTINUE.
        E:       Number (IF VALID) will be in data lights (excluding 11/05)
        F:       Set with any other switch settings desired.  PRESS CONTINUE.

        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 that is not in the order of sequence the  reason  being  is
                 that  the  program  has  to  clear  areas and set up parameters.
                 Note:  if running multiple DZ11's;  the DZ11 you  desire  to  be
                 under test must be selected by the use of SW06 before locking on
                 the test.  In other words;  each time the  program  is  started;
                 the  fit↑Brst DZ11 will be selected to be under test unless SW06 is
                 used to select only one.

        SW 09    LOOP ON CURRENT DATA:   this  switch  will  only  work  if  call
                 'SCOP1'  is in that test.  The reason being that most tests deal
                 with blocks of different data to be sent or received all at once
                 thus in block data;  one pattern cann't be singled out.

        SW 04    SELECT DELAY PARAMETER: THIS SWITCH SHOULD BE USED  WITH  CARE
                 AS TOO SHORT A DELAY WILL CAUSE VALID TESTS TO FAIL ON CERTAIN
                 PROCESSORS.  IT IS RECCOMMENDED THAT THIS SWITCH  ONLY BE
                 IN CONJUNCTION W↑BITH SCOPE LOOPS, E.G. SW 14,9,4,1 SET; SW 9,4,
                 2,1 SET.  THE SHORTEST PARAMETER IS 1; THE LONGEST ACCEPTED IS
                 177776.
```

4.1.3 SWITCH REGISTER PRIORITYS

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. SW 09 (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,↑BO)
2. SW 14
3. SW 11

4.2 STARTING ADDRESS

SA 200 - Address 200 is for normal execution of the diagnostic. This will do the major testing necessary for verification of hardware.

SA 210 - CABLE/ECHO - Terminal Tests. Starting at address 210 will give the user the option to verify the EIA cables at the dist pnl or verify a true link to any DEC supported EIA terminal supported by the DZ11.

NOTE: If address 000042 is non-zero the program assumes it is under
↑B ACT11 or XXDP control and will act accordingly after *ALL* available DZ11'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 four will be printed and program will begin running the diagnostic.

5.1 NORMAL START OF DIAGNOSTIC

On the first start of the diagnostic at address 200; if auto sizing is not used or whenever SW00=1; the following questions are asked and must be answered.

"1ST CSR ADDRESS (160000:163700): "

You must type in the first DZ11 CSR in the system you wish testing to begin at. RANGE: 160000:163700

"1ST VECTOR ADDRESS (300:770): "

You must type in the vector of the first DZ11 in the system under test. RANGE 300:770

"BR LEVEL (4:6): "

type in the priority level of the DZ11 that the above information has been given about. RANGE 4 or 5 or 6.↑B

"TYPE "A" FOR EIA MODULE OR "B" FOR 20MA (A:B): "

Type "A" if running a DZ11-A,B,E (EIA).
Type "B" if running a DZ11-C,D,F (20MA).
Typing a <CR> defaults to EIA MODULES.

```
"MAINTAINCE MODE
 [EXTERNAL  <H325>       (E)]
 [INTERNAL  <DZCSR03=1>(I)]
 [STAGGERED <H327>       (S)]
```

Type "E" or "I" or "S" depending on which mode you wish to run in. If running "EXTERNAL"; all selected lines must be terminated by a H325 test connector.

```
"# OF DZ11'S <IN OCTAL> (1:20):  "

        Type total number of DZ11's to be tested in the  system.   RANGE
        is 1 thru 20 in octal.

    ********* IF SW03=1 THEN *********
        If SW03=1 the following will be printed.

"LINES ACTIVE BY BIT <IN OCTAL> (001:377):"

        Each bit represents a line and any combination of lines  may  be
        selected  (HOWEVER  IN STAGGERED MODE TWO ADJACENT LINES MUST BE
        SELECTED (0-1, 2-3, 4-5, 6-7))..

"DEFAULT BAUD RATE <IN OCTAL> (00:17):  "

        This gives the user a chance to change  the  default  baud  rate
        used  in  APP.   90%  of  the  test.  Normal operation is a "17"
        (19.2k) or "16" (9.6k).  "00"(50 baud)- Not advised.
    *************************************

        It is important to note that all DZ11's in the  system  must  be
        CONTIGIOUS  for  both  ADDRESS  and VECTORS.  also all the EXTRA
        PARAMETERS other than CSR and VECTORS are given to the  EXISTING
        DZ11's in the system.  If not all DZ11's are same priority or if
↑B        the mode of operation is different for each DZ11;  THIS MUST  BE
        "PATCHED"  INTO THE CORRECT STATUS MAP ENTRY which is printed at
        start time.  An alternative is to  put  SW00=1  at  start  time;
        answer  questions about DZ11 under test and INDICATE ONLY 1 DZ11
        in the system.  IF THE STATUS MAP IS TO BE "PATCHED" IT MUST  BE
        DONE AFTER THE QUESTIONS ARE ANSWERED OR AFTER THE AUTO SIZE.
```

5.2 HOW TO RUN THE "CABLE/ECHO" TESTS.

↑B Normal starting for the first time would be: LOAD ADDRESS 210; START WITH THE SWR EQUAL TO 213.
NOTE: SW00=1 ASKS FOR "VECTOR" AND "CSR"
SW01=1 ASKS FOR "WHICH TEST ECHO OR CABLE", "BAUD RATE", "LINE" UNDER TEST. Program will print out:

```
"VECTOR ADDRESS-"
```

You type vector with a <CR>.

```
"CONTROL REGISTER ADDRESS-"
```

You type in DZCSR under test.

```
"WHICH TEST ?  ECHO OR CABLE (E OR C)"
```

Lets do the CABLE TEST first. **THIS TEST IS ONLY TO BE DONE ON THE EIA VERSION OF THE DZ11 NOT THE 20MA VERSION". Type "C" <CR>

```
"BAUD RATE- "
```

type either 50, 110, 135, 150, 300, 600, 1200 1800, 2000, 2400, 3600, 4800, 7200, 9600 followed by <CR>

```
"LINE:  "
```

You type the line which has the H325 test connector. (Type either 0, 1, 2, 3, 4, 5, 6, 7) Program will then print:

```
"CABLE TEST"
```

and if everything is working; the following will be printed:

```
"PASS DONE."
"PASS DONE."
  etc.
```

to change lines: HIT ANY PRINTING KEY↑B ON YOUR CONSOLE TERMINAL WHILE THE PROGRAM IS RUNNING and the following will be printed:

```
"LINE:  "
```

Now change the H325 test connector to another line and type the new line. Program will then print:

```
"CABLE TEST"
"PASS DONE."
"PASS DONE."
```

Continue this operation until all lines are tested.

5.3 ECHO TEST

If program has already been started at 210 and the vector and address have been typed ↑Bin; just load address 210 and start with SWR equal to 212. program will print:

"WHICH TEST ? ECHO OR CABLE (E OR C)"

Now type an "E" to do the ECHO TEST. program will print:

"BAUD RATE-"

Type BAUD RATE at which the terminal is set that is connected to the DZ11 dist pnl. program will print:

LINE: "

Type the line the terminal is connected to at the dist pnl then the program will print:

"TERMINAL ECHO TEST"

*** AT THIS POINT THE MESSAGE:

"THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789"

SHOULD BE PRINTED ON THE TERMINAL CONNECTED TO THE DZ11. IF THIS MESSAGE IS DESIRED TO BE CONTINIOUSLY OUTPUT; SET THE SWR TO 377 (SWR=377) WHILE IT IS BEING OUTPUT OR WHEN PROGRAM IS STARTED AT 210. WHEN THIS MESSAGE IS DONE AND THE SWR IS NOT EQUAL TO 377; THE CONSOLE WILL PRINT:

"TYPE A CHAR. ON DZ11 TERMINAL"

any printable char hit on DZ11 terminal should be echoed back on the terminal. **IF YOU HIT CNTRL C (↑C) ON THE DZ11↑B TERMINAL THE PROGRAM WILL PRINT:

"PASS DONE."

on the console terminal and the "QUICK BROWN FOX" will be printed on DZ11 terminal again and the echo test will be running. TO CHANGE LINES; do like cable test. HIT PRINTABLE KEY ON CONSOLE TERMINAL. And change the line on which the terminal is connected. And enter the new line to the program.

5.4 PROGRAM AND/OR OPERATOR ACTION

↑B 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 FUNCTIONING of the test CAN BE INTERPETED.

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 to the the error message which is to give the operator an indication of the error.

6.2 ERROR RECOVERY

If for some reason the DZ11 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 cont↑Brol of cpu. If this should happen; look in location 'TSTNO' (address 1216)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 DZ11 was doing at the time of the error.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

See section 4. (PLEASE)
Status table should be verified regardless 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

Parameter must be input from user OR APT if "AUTO SIZING" is not used.

8. MISCELLANEOUS

8.1 EXECUTION TIME

All DZ11 device diagnostics will give an 'END PASS' message (providing no errors and sw12=0) within 2 min. This is assuming SW11=2 (DELETE ITERATIONS) is set to give the fastest possible execution. The actual execution time depends greatly on the PDP11 CPU configuration.

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 DZ11'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 DZDZA-C CSR:  160010 VEC:  300 PASSES:  000001 ERRORS:  000000
```

NOTE: The numbers for CSR and VEC are not necessarily the values for the device. They are only f↑Bor this example.

8.4 KEY LOCATIONS

$LPADR (1126)	Contains the address where program will return when iteration count is reached or if loop on test is asserted.
NEXT (1360)	Contains the address of the next test to be peformed.
$TSTNM (1122)	Contains the number of the test now being peformed.
RUN (1406)	The bit in 'RUN' always points one past the DZ11 currently being tested↑B. EXAMPLE: (RUN) 1304/0000000001000000 Means that DZ11 no.05 is the DZ11 now running.
STATUS MAP (1500)-(2000)	These locations contain the information needed to test up to 16 (decimal) DZ11s sequentialy. they contain the CSR,VECTOR and STATUS concerning the configuration of each DZ11.
DZACTV (1404)	Each bit set in this location indicates that the associated DZ11 will be tested in turn. EXAMPLE: (DZACTV) 1300/0000000000011111 means that DZ11 no. 00,01,02,03,04 will be tested. EXAMPLE: (DZACTV) 1300/0000000000010001 Means that DZ11 no. 00,04 will be tested.
$BASE (1310)	Contains the receiver csr of the current DZ11 under test.

8.4A MORE ON THAT 'STATUS TABLE' (1500-2000)

```
'MAP OF DZ 1 STATUS'
1500    160010
1502  ↑B  000300
1504    000005
1506    000377
1510    017470
1512    000000
```

The above information will be repeated for each of up to 8 DZ11's in the system(these will follow under this table). EXPLANATION:

1500	160010	This is the system control register for the 1st DZ11 in the system.
1502	000300	This is vector 'A' for the first DZ11 in the system.
1504	000005	This represents the bus interrupt priority level of the DZ11. BIT15 of this location indicates eit↑Bher EIA or 20MA. if BIT15=0 module should be eia; if bit15=1 module should be 20ma.
1506	000377	This is the binary representation of what lines are to be tested.
1510	017470	This is the parameter location used is most of the tests. It indicated parameters of: RX ON, SPEED SELECT 17 (19.2K BAUD) EIGHT BITS PER CHAR, AND TWO STOP BITS. The user may alter the stop bits and the speed, but the remainding parameters should be left alone.
1512	000000	This location will contain either all zeros indicating that internal loop was selected as mode of operation or it will contain 10000 indicating that "staggered mode" was selected or it will contain 000200 indicating that "external" was the mode selected.

The above is repeated for each DZ11 in the system. The table is filled by AUTO SIZING or by the manual parameter input program as described previously. Also if desired by ust↑Ber; the locations may be altered by hand (toggled in) to suit the specific configuration.

8.5 *** METHOD OF AUTO SIZING ***

8.5.1 FINDING THE CONTROL STATUS REGISTER.

The program will start at address 160000 and start 'REFERENCEING' the address in the pointer. If a NON-EX MEMORY TRAP occures, the pointer (holding 160000) is updated by 10 and the above is repeated until address 163700 is reached. If a 'SLAVE SYNC RESPONSE'↑B was issued by the DZ11 (or any other device) (no nxm trap), "MASTER SCAN ENABLE" is attempted to be set and the "TCR" bit for line 7 is set. "TRDY" is then tested to be set and both "TCR07" AND "MASTER SCAN ENABLE" are tested to be still set. If all of this worked; then a "DEVICE CLEAR" is issued testing that the bit can be read back and that after some time it self clears. If all of the above worked; this device is assumed to be a DZ11. If any of the above failed; updating of the pointer is done and the sequence is repeated.
NOTE: If the program does not find your DZ11; something is wrong and AUTO SIZING should not be done.

8.5.2 FINDING THE VECTOR

The vector area (address 300-776) is filled with the instruction IOT and '.+2' (next address). Bit14 and Bit5 (TX INTERUPT ENABLE AND MSTSCAN ENABLE) are set into the DZCSR. "TCR07" is then set. a delay is made and if no interupt occures (because of a bad DZ11) the program assumes vector address 300 and the problem should be fixed in the diagnostic. Once the problem is fixed; the program should be re-setup again to get
↑B correct vector. If an interupt occurred; the address to which the DZ11 interupted to is picked up and reported as the vector. NOTE: if the vector reported is not the vector set up by you; there is a problem and AUTO SIZING should not be done.

8.5.3 PARAMETER ASSUMPTIONS.

Since too much hardware would need to be turned on to SIZE the rest of the parameters; the program must assume the remaining variations. The result if not to your specific configuration may be altered by hand (toggle in) if desired. In this way 95% of the parameter setup was↑B done by the program and 5% by you.
THEREFORE:

```
1)      BUS PRIORITY IS SET TO LEVEL5.
2)      ALL EIGHT LINES ARE ASSUMED TO BE TESTED.
3)      DEFAULT BAUD RATE IS SET TO 17 (19.2 K).
4)      MODE OF OPERATION IS "INTERNAL MODE".
5)      MODULE IS ASSUMED TO BE "EIA" VERSION.
        SET BIT 15 IN PRIORITY ENTRY OF MAP IF YOU HAVE A 20MA MODULE.
```

In all adjustments please refer to section 8.4a for greater detail.

9.0 RUNNING THE DZ11 DIAGNOSTIC UNDER APT

9.1.1 THE APT INTERFACE

DZDZA has been redesigned to be compatible with the APT-Automated Product Test system. It can be run as a standalone diagnostic or in either of the APT modes. Certain variables in the original APT module were reassigned to the areas set aside for APT interfacing. These new variables generally begin with a dollar sign ($), e.g., $DEVM, $BASE.

9.1.2 SETTING UP THE DIAGNOSTIC USING APT

The diagnostic uses several variables in the region subtitled 'APT Mailbox-Etable'. These variables are:

$SWREG -	used if a software switch register is desired while under↑B apt
$VECT1 -	used to specify the interrupt level and the first vector address
$BASE -	used to indicate bottom address of DZ11 under test
$DEVM -	a bit map representing which DZ11's will be tested
$CDW1 -	used to indicate which lines to run on all DZ11's
$DDW0 -	each of the $DDW words describes the parameters (LPR) for a particular DZ11, going up to 16 DZ11's

9.1.3 RUNNING UNDER APT

The user should be familiar with the APT system. The APT timing parameters for the DZ11 diagnostic were b↑Based on an 11/40 processor. It may be necessary to add a few more seconds if the diagnostic is out on an 11/05 processor.

All of the variables mentioned in section 9.1.2 should be set up prior to running the diagnostic under APT.

NOTE

Be sure $BASE points to the first DZ11 before running

Based on these values, the diagnostic will set up the status table. The user is then free to monitor under APT as normal.

```
2          COPYRIGHT (C) 1976
           DIGITAL EQUIPMENT CORP.
           MAYNARD, MASS. 01754

           PROGRAM BY JERRYL PAYNE,JOHN EGOLF

           THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
           PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.

23         INITIAL ADDRESS OF THE STACK POINTER *** 1120 ***

28         MISCELLANEOUS DEFINITIONS

40         GENERAL PURPOSE REGISTER DEFINITIONS

52         PRIORITY LEVEL DEFINITIONS

62         "SWITCH REGISTER" SWITCH DEFINITIONS

90         DATA BIT DEFINITIONS (BIT00 TO BIT15)

118        BASIC "CPU" TRAP VECTOR ADDRESSES

354        THIS TABLE CONTAINS VARIOUS COMMON STORAGE L↑BOCATIONS
           USED IN THE PROGRAM.

424                                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

432                                MEM.TYPE BYTE   --  (HIGH BYTE)
                                          900 NSEC CORE=001
                                          300 NSEC BIPOLAR=002
                                          500 NSEC MOS=003

437                                  MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE

475        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 I↑BN 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:

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

```
1088    INCREMENT THE PASS NUMBER ($PASS)
        IF THERES A MONITOR GO TO IT
        IF THERE ISN'T JUMP TO CYCLE

1149    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

1225    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:
        ↑B1) USING A TRAP INSTRUCTION
                TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
        OR
                TYPE
                MESADR

1927    ROUTINE USED TO "AUTO SIZE" THE DZ11
        CSR AND VECTOR.
        NOTE:   THE CSR MAY BE ANY WHERE IN THE FLOATING
                ADDRESS RANGE (160000:163700)
                AND THE VECTOR MAY BE ANY WHERE IN THE
                FLOATING VECTOR RANGE (300:770)

2044    ************************ TEST 1 ******************************
        THIS TEST PROVES THE SLAVE SYNC RESPONSE
        DURING A READ OR WRITE TO THE FOLLOWING ADDRESS:
                DZCSR, DZRBUF, DZTCR, DZMSR

2087    ************************ TEST 2 ******************************
↑B      THIS TEST PROVES THAT BIT "DCLR"
        CAN BE SET AND THAT IT WILL CLEAR
        BY ITSELF AFTER A PERIOD OF TIME.

2117    ************************ TEST 3 ******************************
        TEST TO VERIFY THAT BIT "MAINT" CAN
        BE SET. THEN VERIFY THAT BIT "MAINT" CAN
        BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
        VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
        CLEARED BY A "DEVICE CLEAR"
        2149    *********************** TEST 4 ******************************
                TEST TO VERIFY THAT BIT "MSENAB" CAN
                BE SET. THEN VERIFY THAT BIT "MSENAB" CAN
                BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                CLEARED BY A "DEVICE CLEAR"

        2181    *********************** TEST 5 ******************************
                TEST TO VERIFY THAT BIT "SILOEN" CAN
                BE SET. THEN VERIFY THAT BIT "SILOEN" CAN
                BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                CLEARED BY A "DEVICE CLEAR"

        2213    *********************** TEST 6 ******************************
                TEST TO VERIFY THAT BIT "RIE" CAN
                BE SET. THEN VERIFY THAT BIT "RIE" CAN
                BE CLEARED (WR↑BITTEN TO A ZERO). AND FINALLY
                VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                CLEARED BY A "DEVICE CLEAR"

        2245    *********************** TEST 7 ******************************
                TEST TO VERIFY THAT BIT "TIE" CAN
                BE SET. THEN VERIFY THAT BIT "TIE" CAN
                BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                CLEARED BY A "DEVICE CLEAR"

        2277    *********************** TEST 10 *****************************
                THIS TESTS THAT ALL OF THE FOLLOWING
                BITS CAN BE: SET, CLEARED, CLEARED BY "DEVICE CLEAR "
                BITS TESTED ARE:
                 TCR0, TCR1, TCR2, TCR3, TCR4, TCR5, TCR6, TCR7

↑B      2319    *********************** TEST 11 *****************************
                THIS TESTS THAT ALL OF THE FOLLOWING
                BITS CAN BE: SET, CLEARED, CLEARED BY "RESET INSTR *NOT* DEVICE CLEAR "
                BITS TESTED ARE:

        2323     DTR0, DTR1, DTR2, DTR3, DTR4, DTR5, DTR6, DTR7
                THIS TEST IS NOT DONE IF MODULE IS 20MA VERSION

        2371    *********************** TEST 12 *****************************
                THIS TEST PERFORMS RESET TESTING &
                TESTING OF WRITE ONLY OR READ ONLY BIT
                 TEST BITS "RDONE, BIT11, BIT10, BIT9, BIT8, BIT2, BIT1
                            BIT0, SILOAL" ARE READ ONLY AND THAT TRDY IS
                            ZERO UNTIL A LINE IS SELECTED AND MSENAB IS SET.
```

```
2408      *********************** TEST 13 ******************************
          THIS TEST PERFORMS RESET TESTING AND
          TESTING OF READ ONLY AND WRITE ONLY BITS
           IN REGISTER DZCSR
          VERIFY THAT "TIE", "SILOEN", "RIE", "MSENAB", "MAINT"
          ARE THE ONLY R/W BITS IN THE DZCSR.
          THEN SET "DCLR" AND VERIFY THEY ARE CLEARED

2437      *********************** TEST 14 ******************************
          THIS TEST PERFORMS RESET TESTING AND
          TESTING OF READ ONLY REGISTER DZR↑BBUF
          AND TESTING OF WRITE ONLY REGISTER DZLPR

2463      *********************** TEST 15 ******************************
          THIS TEST PERFORMS RESET TESTING AND
          TESTING OF READ ONLY REGISTER DZMSR
          AND TESTING OF WRITE ONLY REGISTER DZTDR

2489      *********************** TEST 16 ******************************
          VERIFY THAT IF WE ARE IN "STAGGERED" MODE

2491      THAT SETTING "DTR" FOR A LINE WILL
          BRING UP "RING" AND "CARRIER" FOR THE
          ASSOCIATED LINE IN WHICH WE ARE STAGGERED!
           LINE0 DTR= LINE1 RING AND CARRIER
           LINE1 DTR= LINE0 RING AND CARRIER
           LINE2 DTR= LINE3 RING AND CARRIER
           LINE3 DTR= LINE 4↑B RING AND CARRIER
                           ETC...

2546      *********************** TEST 17 ******************************

2547      TEST TO VERIFY THAT IF IN "EXTERNAL"
          MODE; SETTING DTR FOR SELECTED LINES
          WILL BRING UP "CARRIER" AND "RING"
          FOR THAT SAME LINE. NOTE: IF YOU HAVE
          SELECTED MODE AS "EXTERNAL"; THE H325 TEST CONNECTER
          MUST BE USED ON ALL SPECIFIED LINES.
          LINES MAY BE SPECIFIED BY SWR03=1
          AND SWR00=1 AT START TIME OR ALTERING
          STATUS MAP.

2593      *********************** TEST 20 ******************************
           THIS TEST VERIFIES THAT TRDY IS SET WHEN A LINE
           IS READY TO BE LOADED, AND THAT THE LINE SPECI-
           FIED IN BITS 8-10 OF DZCSR CORRESPOND
           TO THE LINE SELECTED IN DZTCR

2627      *********************** TEST 21 ******************************
          TEST TO TRANSMIT ONE CHAR AND
          RECEIVE ONE CHAR ON ONE LINE
          AT A TIME. THE CHAR IS "252" AND
          ALL SELECTED LINES WILL BE TURNED ON
          ONE AT A TIME. THIS IS THE FIRST TIME ANY
```

```
              DATA IS CHECKED IN THE RECEIVER.
              USING SWITCH NINE WITH THIS TEST CREATES A TIGHT SCOPE LOOP
              WHICH TRANSMITS A STEADY STREAM OF CHARACTERS.
↑B
      2711    *********************** TEST 22 ******************************
               THIS TEST PROVES THAT THE TRANSMITTER TRANSMITS
              CHARACTERS (FLAG MODE)AND THE RECEIVER RECEIVES (FLAG MODE)
              (ONE LINE AT A TIME   BASED UPON VALID LINES)

      2715    THIS IS THE FIRST TIME THAT ALL DATA IS CHECKED

      2792    *********************** TEST 23 ******************************
              THIS TEST WILL PROVE THAT:
               1) THE TRANSMITTER "BREAK BIT" WORKS
               2) THE RECEIVER CAN FLAG "FRAMING ERRORS"
               3) THE RECEIVER CAN FLAG "PARITY ERRORS"
              ONLY ONE LINE AT A TIME WILL BE EXERCISED.
              THIS TEST WILL NOT BE  EXERCISED UNLESS
              CONNECTE↑BD BY EXTERNAL PLUG.

      2859    *********************** TEST 24 ******************************
               THIS TEST VERIFIES THAT THE DEVICE DOES NOT INTERRUPT
              WHILE THE PROCESSOR STATUS IS SET EXACTLY
              TO WHAT THE DZ11 PRIORITY IS SET TO.
              DEFAULT PRIORITY IS AT  5 (240).

      2927    *********************** TEST 25 ******************************
               THIS TEST VERIFIES THAT THE DEVICE DOES INTERRUPT
              WHILE THE PROCESSOR STATUS IS SET TO EXACTLY
              ONE LEVEL LOWER THAN THE DZ11. DZ11 PRIORITY
              DEFAULT TO LEVEL 5 MINUS ONE LEVEL IS LEVEL 4.

      3001    *********************** TEST 26 ******************************
              THIS TEST VERIFIES THAT THE RECEIVER WILL
              INTERRUPT BEFORE THE TRANSMITTER EVEN
              THOUGH THE TRANSMITTER WAS ENABLED
              FIRST.  SET PS TO LEVEL 7;
              GET RDONE AND TRDY TO SET;
              SET TX IE AND RX IE;
              CLEAR PS AND EXPECT RX TO INTERRUPT FIRST

      3111    *********************** TEST 27 ******************************
              THIS TEST VERIFIES OVERRUN AND SILO ALARM
              ONE LINE AT A TIME  - BASED UPON VALID LINES
              AS EACH OF THE FIRST 16 CHARS ARE SENT; SILO ALARM IS
              TESTED TO BE CLEARED. ON THE 16TH CHAR THE PROGRAM THEN
              EXPECTS SILO ALARM TO SET. THEN THE ENTIRE
              SILO IS FILLED AND AN OVERRUN IS EXPECTED ON THE 65TH
↑B            CHAR PULLED OUT OUT THE SILO.
              USING SWITCH NINE FOR THIS TEST SENDS 20. CHARACTERS
              ON DZ LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
              USED TO SCOPE SILO ALARM PULSES, ETC.
```

```
3246        *********************** TEST 30 ******************************
            THIS TEST THAT "SILO ENABLE" WILL INHIBIT
            RECEIVER INTERRUPTS AND THAT ON THE
            16TH CHAR THAT "SILO ALARM" WILL CAUSE AN
            INTERRUPT WITH "RIE" SET.
            THIS WILL DO ALL SELECTED LINES ONE AT A TIME.

3331        *************↑B********** TEST 31 ******************************
            THIS TEST RUNS ALL LINES FULL BORE
            BASED UPON QUALIFIED LINES
            ..THIS IS AN INTERRUPT TEST ON THE RECEIVER AND
            TRANSMITTER

3475        *********************** TEST 32 ******************************
            DZ11 RELATIVE TIMING TEST.
            EACH SELECTED LINE WILL IN TURN RUN 16. CHARS
            AT ALL BAUD RATES AND THEN THE HIGHEST BAUD
            WITH ALL CHAR LENGTHS. EACH NEW PARAMETER SHOULD
            DECREASE IN TIME FROM THE PREVIOUS PARAMETERS SELECTED.
            THE TIME IS CHECKED AGAINST THE LAST PARAMETER USED
             AND A LOWER TIME IS EXPECTED ON THE CURRENT PARAMETER.
            PARAMETERS ARE:
              EIGHT BITS/PER/CHAR - TWO STOP BITS AT
                    50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000
                    2400, 3600, 4800, 7200, 9600 BAUD.
              19.2 K BAUD - TWO STOP BITS AT
                    SEVEN, SIX, FIVE BITS/PER/CHAR.
            AFTER EACH LINE HAS FINISHED ALL THE ABOVE PARAMETERS
            THE NEXT SELECTED LINE IS THE TESTED.

3572        *********************** TEST 33 ******************************
             THIS TEST VERIFIES THAT EVEN PARITY WORKS
             FOR ALL ODD LINES SELECTED AND THAT ODD PARITY WORKS FOR ALL
             EVEN LINES SELECTED.
            THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
            THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
            THE UARTS. THI↑BS TEST WILL NOT BE DONE UNLESS
            YOU ARE IN "STAGGERED" MODE.
            40(8) CHARS ARE USED FOR THIS TEST.
            ALL SELECTED LINES WILL BE ENABLED
            AT THE SAME TIME!

3671        *********************** TEST 34 ******************************
            THIS TEST VERIFIES THAT ODD PARITY WORKS FOR ALL ODD LINES
             SELECTED AND THAT EVEN PARITY WORKS FOR ALL EVEN LINES SELECTED
            THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
            THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
            THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
            YOU ARE IN "STAGGERED" MODE.
            40(8) CHARS ARE USED FOR THIS TEST.
            ALL SELECTED LINES WILL BE ENABLED
            AT THE SAM↑BE TIME!
```

```
3855    STARTING PROCEDURE
        LOAD PROGRAM
        LOAD ADDRESS 000210
        PRESS START
        PROGRAM WILL TYPE DZ11 ECHO/CABLE TEST
        PROGRAM WILL TYPE  WHICH TEST- ECHO OR CABLE
        TYPE IN  E  OR  C   RESPECTIVELY
        PROGRAM WILL TYPE "VECTOR ADDRESS-"
        TYPE IN THE ADDRESS OF THE RECEIVER INTERRUPT VECTOR
        FOR THE DZ11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
        PROGRAM WILL TYPE "CONTROL REGISTER ADDRESS-"
        TYPE IN THE ADDRESS OF THE SYSTEM CONTROL REGISTER
        FOR THE DZ11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
        PROGRAM WILL TYPE "LINE NUMBER-"
        TYPE IN THE LINE NUMBER TO BE TESTED (IN OCTAL)
        ,FOLLOWED BY <CARRIAGE RETURN>
        PROGRAM WILL TYPE "BAUD RATE-"
        TYPE IN THE BAUD RATE OF THE DZ11 TERMINAL
        ,FOLLOWED BY <CARRIAGE RETURN>
        THE FOLLOWING BAUD RATES ARE ACCEPTED IN DECIMAL
                        50
                        75
                        110
                        135      (ROUNDED OFF   134.5)
                        150
                        300
                        600
                        1200
                        1800
                        2000
                        2400
                        3600
                        4800
                        7200
                        9600
        ALL OTHERS ARE REJECTED

3892    PROGRAM WILL TYPE "ECHO"  OR  "CABLE TEST" TO INDICATE THAT TESTING HAS STARTE

401872  TEST TO VERIFY THAT SETTING DTR FOR A GIVEN LINE
        WILL BRING UP "CO" AND "RING" FOR THE SAME LINE
        THE DIST PNL MUST HAVE JUMPER FROM DTR TO RQST TO SEND
        IN ORDER FOR THIS TEST TO WORK!
```

```
                                .TITLE  MD-11-DZDZA-C
                                ;*COPYRIGHT (C) 1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY JERRYL PAYNE,JOHN EGOLF
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
                                ;*
        000001                  $TN=1
                                        ;STARTING PROCEDURE
                                        ;LOAD PROGRAM
                                        ;LOAD ADDRESS 000200
                                        ;PRESS START
                                        ;PROGRAM WILL TYPE "MAINDEC-11-DZDZAC/<200>/EIGHT LINE ASYNC MUX TESTS"
                                        ;PROGRAM WILL TYPE "RUNNING" TO INDICATE THAT TESTING HAS STARTED
                                        ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
                                        ;AND THEN RESUME TESTING

                                .SBTTL  BASIC DEF↑BINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1120 ***
        001120                  $TACK=  1120
                                .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
↑B              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 DEFINITIO↑BNS
        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↑B         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             ↑B;;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=572↑B6    ;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

                                ;DZ11 CONTROL AND STATUS REGISTER DEFINITIONS
                                ;(DZCSR)        BIT DEFINITIONS
                                ;------------------------------

        000010          MAINT = BIT3     ;MAINTENANCE MODE ENABLE
        000020          DCLR=BIT4        ;DEVICE CLEAR
        000040          MSENAB=BIT5      ;MASTER SCAN ENABLE
        000100          RIE=BIT6         ;RECEIVER INTERRUPT ENABLE
        000200          RDONE=BIT7       ;RECEIVER DONE
        010000          SILOEN= BIT12    ;SILO ALARM ENABLE
        020000          SILOAL = BIT13   ;SILO ALARM
        040000          TIE=BIT14        ;TRANSMITTER INTERRUPT ENABLE
        100000          TRDY=BIT15       ;TRANSMITTER READY

                                ;DZCSR WORD DEFINITIONS
                                ;---------------------
        000000          TL0=0            ;TRANSMIT LINE 0
        000400          TL1=BIT8         ;TRANSMIT LINE 1
        001000          TL2=BIT9         ;TRANSMIT LINE 2
        001400          TL3=BIT9!BIT8    ;TRANSMIT LINE 3
        002000          TL4=BIT10        ;TRANSMIT LINE 4
        002400          TL5=BIT10!BIT8   ;TRANSMIT LINE 5
        003000          TL6=BIT10!BIT9   ;TRANSMIT LINE 6
        003400          TL7↑B=BIT10!BIT9!BIT8 ;TRANSMIT LINE 7
```

```
                                        ;DZRBUF BIT DEFINITIONS
                                        ;----------------------

        010000          PARER=BIT12     ;PARITY ERROR
        020000          FRMERR=BIT13    ;FRAME ERROR
        040000          OVRRUN=BIT14    ;OVERRUN ERROR
        100000          DVALID=BIT15    ;DATA VALID

                                        ;DZRBUF WORD DEFINITIONS
                                        ;-----------------------

        000000          RL0=0           ;RECEIVER LINE 0
        000400          RL1=BIT8        ;RECEIVER LINE 1
        001000          RL2=BI↑BT9      ;RECEIVER LINE 2
        001400          RL3=BIT9!BIT8   ;RECEIVER LINE 3
        002000          RL4=BIT10       ;RECEIVER LINE 4
        002400          RL5=BIT10!BIT8  ;RECEIVER LINE 5
        003000          RL6=BIT10!BIT9  ;RECEIVER LINE 6
        003400          RL7=BIT10!BIT9!BIT8 ;RECEIVER LINE 7

                                        ;DZLPR WORD DEFINITIONS
                                        ;----------------------

        000000          LP0=0           ;LINE PARAMETER 0
        000001          LP1=BIT0        ;LINE PARAMETER 1
        000002          LP2=BIT1        ;LINE PARAMETER 2
        000003          LP3=BIT1!BIT0   ;LINE PARAMETER 3
        000004          LP4=BIT2        ;LINE PARAMETER 4
        000005          LP5=BIT2!BIT0   ;LINE PARAMETER 5
        000006          LP6=BIT2!BIT1   ;LINE PARAMETER 6
        000007          LP7=BIT2!BIT1!BIT0      ;LINE PARAMETER 7

        000000          FIVE=0          ;FIVE BITS/CHAR,1 STOP BIT
        000010          SIX=BIT3        ;SIX BITS/CHAR,1 STOP BIT
        000020          SEVEN=BIT4      ;SEVEN BITS/CHAR,1 STOP BIT
        000030          EIGHT=BIT4!BIT3 ;EIGHT BITS/CHAR,1 STOP BIT
        000040          FIVES=BIT5      ;FIVE BITS/CHAR,2 STOP BITS
        000050          SIXS=BIT5!BIT3  ;SIX BITS/CHAR,2 STOP BITS
        000060          SEVENS=BIT5!BIT4        ;SEVEN BITS/CHAR, 2 STOP BITS
        000070          EIGHTS=BIT5!BIT4!BIT3   ;EIGHT BITS/CHAR, 2 STOP BITS

        000100          PARITY=BIT6             ;PARITY ENABLE↑BD
        000200          ODDPAR=BIT7             ;ODD PARITY ENABLED
        000000          ONESTOP=0               ;ONE STOP BIT ENABLED
        000040          TWOSTOP=BIT5            ;TWO STOP BITS ENABLED
        000000          EVEPAR=0                ;EVEN PARITY ENABLED
        010000          RCVON=BIT12             ;ENABLE RECEIVER (RECEIVER ON)

        000000          S50=0                   ;SPEED 50 BAUD
        000400          S75=BIT8                ;SPEED 75 BAUD
        001000          S110=BIT9               ;SPEED 110 BAUD
        001400          S134=BIT9!BIT8          ;SPEED 134.5 BAUD
        002000          S150=BIT10              ;SPEED 150 BAUD
        002400          S300=BIT10!BIT8         ;SPEED 300 BAUD
        003000          S600=BIT10!BIT9         ;SPEED 600 BAUD
        003400          S1200=BIT10!BIT↑B9!BIT8 ;SPEED 1200 BAUD
```

```
 004000          S1800=BIT11                     ;SPEED 1800 BAUD
 004400          S2000=BIT11!BIT8                ;SPEED 2000 BAUD
 005000          S2400=BIT11!BIT9                ;SPEED 2400 BAUD
 005400          S3600=BIT11!BIT9!BIT8           ;SPEED 3600 BAUD
 006000          S4800=BIT11!BIT10               ;SPEED 4800 BAUD
 006400          S7200=BIT11!BIT10!BIT8          ;SPEED 7200 BAUD
 007000          S9600=BIT11!BIT10!BIT9          ;SPEED 9600 BAUD
 007400          S19200=BIT11!BIT10!BIT9!BIT8        ;SPEED 19200 BAUD

                          ;DZTCR BIT DEFINITIONS
                          ;----------------------
 000001          TCR0=BIT0                       ;TCR0
 000002          TCR1=BIT1                       ;TCR1
 000004          TCR2=BIT2                       ;TCR2
 000010          TCR3=BIT3                       ;TCR3
 000020          TCR4=BIT4                       ;TCR4
 000040          TCR5=BIT5                       ;TCR5
 000100          TCR6=BIT6                       ;TCR6
 000200          TCR7=BIT7                       ;TCR7
 000400          DTR0=BIT8                       ;DTR0
 001000          DTR1=BIT9                       ;DTR1
 002000          DTR2=BIT10                      ;DTR2
 004000          DTR3=BIT11                      ;DTR3
 010000          DTR4=BIT12                      ;DTR4
 020000          DTR5=BIT13                      ;DTR5
 040000          DTR6=BIT14                      ;DTR6
 100000          DTR7=BIT15                      ;DTR7

                          ;DZMSR BIT DEFINI↑BTIONS
                          ;----------------------
 000001          RING0=BIT0                      ;RING INDICATED ON LINE 0
 000002          RING1=BIT1                      ;RING INDICATED ON LINE 1
 000004          RING2=BIT2                      ;RING INDICATED ON LINE 2
 000010          RING3=BIT3                      ;RING INDICATED ON LINE 3
 000020          RING4=BIT4                      ;RING INDICATED ON LINE 4
 000040          RING5=BIT5                      ;RING INDICATED ON LINE 5
 000100          RING6=BIT6                      ;RING INDICATED ON LINE 6
 000200          RING7=BIT7                      ;RING INDICATED ON LINE 7
 000400          CO0=BIT8                        ;CARRIER PRESENT ON LINE 0
 001000          CO1=BIT9                        ;CARRIER PRESENT ON LINE 1
 002000          CO2=BIT10                       ;CARRIER PR↑BESENT ON LINE 2
 004000          CO3=BIT11                       ;CARRIER PRESENT ON LINE 3
 010000          CO4=BIT12                       ;CARRIER PRESENT ON LINE 4
 020000          CO5=BIT13                       ;CARRIER PRESENT ON LINE 5
 040000          CO6=BIT14                       ;CARRIER PRESENT ON LINE 6
 100000          CO7=BIT15                       ;CARRIER PRESENT ON LINE 7

                          ;DZTDR BIT DEFINITIONS
                          ;----------------------
 000400          BRK0=BIT8                       ;BREAK FOR LINE 0
 001000          BRK1=BIT9                       ;BREAK FOR LINE 1
 002000          BRK2=BIT10                      ;BREAK FOR LINE 2
 004000          BRK3=BIT11                      ;BREAK FOR LINE 3
 010000          BRK4=BIT12                      ;BREAK FOR LINE 4
 020000          BRK5=BIT13                      ;BREAK FOR LINE 5
```

```
            040000              BRK6=BIT14              ;BREAK FOR LINE 6
            100000              BRK7=BIT15              ;BREAK FOR LINE 7


                                ;TABLE OF LOOP AROUND FUNCTIONS (H325)
                                ;
                                ;       -----------------
                                ;       I               ↑
                                ;       V               ↑
                                ;       REC             TRANS
                                ;       DATA            DATA
                                ;
                                ;       -----------------
                                ;       I               ↑
                                ;       V               ↑
                                ;       CO              RTS
                                ;
                                ;
                                ;       -----------------
                                ;       I               ↑↑B
                                ;       V               ↑
                                ;       RING            DTR
```

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

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

         000010                    .=10
000010   010650                                 SET.PS                          ;FAKE "MTPS" INSTRUCTION TRAP
000012   000340                                 PR7                             ;MAKE SURE PS IS PRIORITY 7

         000020                    .=20
000020   004654                                 .SCOPE                          ;SCOPE LOOP HANDLER
000022   000340                                 PR7                             ;HANDLE AT PRIORITY 7
000024   007530                                 $PWRDN                          ;POWER FAIL HANDLER
000026   000340                                 340                             ;SERVICE AT PRIORITY LEVEL 7
000030   006620                                 $ERROR                          ;ERROR HANDLER
000032   000340                                 340                             ;SERVICE AT PRIORITY LEVEL 7
000034   006512                                 .TRPSRV                         ;GENERAL HANDLER DISPATCH SERVICE
000036   000340                                 340                             ;SERVICE AT PRIORITY LEVEL 7
                                   .SBTTL  ACT11 HOOKS

                                   ;;**************************************************************
                                   ;HOOKS REQUIRED BY ACT11
         000040                                 $SVPC=.                         ;SAVE PC
         000046                                 .=46
000046   004610                                 $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   0000↑B00                               DISPREG:0                               ;SOFTWARE DISPLAY REGISTER FOR SWITCHLESS 11S
000176   000000                    SWREG:       0                               ;SOFTARE SWITCH REGISTER FOR SWITCHLESS 11S
         000200                    .=200
000200   000137   002150                        JMP     .START                  ;GO TO START OF PROGRAM
         000210                    .=210
000210   000137   023142                        JMP     XSTART                  ;GOTO CABLE TEST/ECHO TEST


         001000                    .=1000
001000   005200   040515   047111  MTITLE: .ASCIZ  <200><12>/MAINDEC-11-DZDZAC/<200>/EIGHT LINE ASYNC MUX TESTS/<200>
```

```
                                .SBTTL  COMMON TAGS

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

                001120                  .=1120
        001120                  $CMTAG:                                 ;;START OF COMMON TAGS
        001120  000000                  .WORD   0
        001122     000          $TSTNM: .BYTE   0                       ;;CONTAINS THE TEST NUMBER
        001123     000          $ERFLG: .BYTE   0                       ;;CONTAINS ERROR FLAG
        001124  000000          $ICNT:  .WORD   0                       ;;CONTAINS SUBTEST ITERATION COUNT
        001126  000000          $LPADR: .WORD   0                       ;;CONTAINS SCOPE LOOP ADDRESS
        001130  000000          $LPERR: .WORD   0                       ;;CONTAINS SCOPE RETURN FOR ERRORS
        001132  000000          $ERTTL: .WORD   0                       ;;CONTAINS TOTAL ERRORS DETECTED
        001134     000          $ITEMB: .BYTE   0                       ;;CONTAINS ITEM CONTROL BYTE
        001135     001          $ERMAX: .BYTE   1                       ;;CONTAINS MAX. ERRORS PER TEST
        001136  000000          $ERRPC: .WORD   0                       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
        001140  000000          $GDADR: .WORD   0                       ;;CONTAINS ADDRESS OF 'GOOD' DATA
        001142  000000          $BDADR: .WORD   0                       ;;CONTAINS ADDRESS OF 'BAD' DATA
        001144  000000          $GDDAT: .WORD   0                       ;;CONTAINS 'GOOD' DATA
        001146  000000          $BDDAT: .WORD   0                       ;;CONTAINS 'BAD' DATA
        001150  000000                  .WORD   0                       ;;RESERVED--NOT TO BE U↑BSED
        001152  000000                  .WORD   0
        001154     000          $AUTOB: .BYTE   0                       ;;AUTOMATIC MODE INDICATOR
        001155     000          $INTAG: .BYTE   0                       ;;INTERRUPT MODE INDICATOR
        001156  000000                  .WORD   0
        001160  177570          SWR:    .WORD   DSWR                    ;;ADDRESS OF SWITCH REGISTER
        001162  177570          DISPLAY: .WORD  DDISP                   ;;ADDRESS OF DISPLAY REGISTER
        001164  177560          $TKS:   177560                          ;;TTY KBD STATUS
        001166  177562          $TKB:   177562                          ;;TTY KBD BUFFER
        001170  177564          $TPS:   177564                          ;;TTY PRINTER STATUS REG. ADDRESS
        001172  177566          $TPB:   177566                          ;;TTY PRINTER BUFFER REG. ADDRESS
        001174     000          $NULL:↑B          .BYTE   0
                                                                                ;;CONTAINS NULL CHARACTER FOR FILLS
        001175     002          $FILLS: .BYTE   2                       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
        001176     012          $FILLC: .BYTE   12                      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
        001177     000          $TPFLG: .BYTE   0                       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
        001200  000000          $REGAD: .WORD   0                       ;;CONTAINS THE ADDRESS FROM
                                                                        ;;WHICH  ($REG0) WAS OBTAINED
        001202  000000          $REG0:  .WORD   0                       ;;CONTAINS (($REGAD)+0)
        001204  000000          $REG1:  .WORD   0                       ;;CONTAINS (($REGAD)+2)
        001206  000000          $REG2:  .WORD   0                       ;;CONTAINS (($REGAD)+4)
        001210  000000          $REG3:  .WORD   0                       ;;CONTAINS (($REGAD)+6)
        001212  000000          $REG4:  .WORD   0                       ;;CONTAINS (($REGAD)+10)
        001214  000000          $REG5:  .WORD   0                       ;;CONTAINS (($REGAD)+12)
        001216  000000          $TMP0:  .WORD   0                       ;;USER DEFINED
        001220  000000          $TMP1:  .WORD   0                       ;;USER DEFINED
        001222  000000          $TMP2:  .WORD   0                       ;;USER DEFINED
        001224  000000          $TMP3:  .WORD   0                       ;;USER DEFINED
        001226  000000          $TIMES: 0                               ;;MAX. NUMBER OF ITERATIONS
        001230     077          $QUES:  .ASCII  /?/                     ;;QUESTION MARK
        001231     015          $CRLF:  .ASCII  <15>                    ;;CARRIAGE RETURN
        001232  000012          $LF:    .ASCIZ  <12>                    ;;LINE FEED
                                ;;*******↑B*************************************************************
                                .SBTTL  APT MAILBOX-ETABLE
```

```
                          ;;*******************************************************************
                          .EVEN
001234                    $MAIL:                              ;;APT MAILBOX
001234   000000           $MSGTY:  .WORD    AMSGTY            ;;MESSAGE TYPE CODE
001236   000000           $FATAL:  .WORD    AFATAL            ;;FATAL ERROR NUMBER
001240   000000           $TESTN:  .WORD    ATESTN            ;;TEST NUMBER
001242   000000           $PASS:   .WORD    APASS             ;;PASS COUNT
001244   000000           $DEVCT:  .WORD    ADEVC↑BT                ;;DEVICE COUNT
001246   000000           $UNIT:   .WORD    AUNIT             ;;I/O UNIT NUMBER
001250   000000           $MSGAD:  .WORD    AMSGAD            ;;MESSAGE ADDRESS
001252   000000           $MSGLG:  .WORD    AMSGLG            ;;MESSAGE LENGTH
001254                    $ETABLE:                            ;;APT ENVIRONMENT TABLE
001254      000           $ENV:    .BYTE    AENV              ;;ENVIRONMENT BYTE
001255      000           $ENVM:   .BYTE    AENVM             ;;ENVIRONMENT MODE BITS
001256   000000           $SWREG:  .WORD    ASWREG            ;;APT SWITCH REGISTER
001260   000000           $USWR:   .WORD    AUSWR             ;;USER SWITCHES
001262   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
001264      000           $MAMS1:  .BYTE    AMAMS1            ;;HIGH ADDRESS,M.S. BYTE
001265      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
001266   000000           $MADR1:  .WORD    AMADR1            ;;HIGH ADDRESS,BLK#1
                          ;*                                    MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
43↑B8          001270      000            $MAMS2:  .BYTE    AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
001271      000           $MTYP2:  .BYTE    AMTYP2            ;;MEM.TYPE,BLK#2
001272   000000           $MADR2:  .WORD    AMADR2            ;;MEM.LAST ADDRESS,BLK#2
001274      000           $MAMS3:  .BYTE    AMAMS3            ;;HIGH ADDRESS,M.S.BYTE
001275      000           $MTYP3:  .BYTE    AMTYP3            ;;MEM.TYPE,BLK#3
001276   000000           $MADR3:  .WORD    AMADR3            ;;MEM.LAST ADDRESS,BLK#3
001300      000           $MAMS4:  .BYTE    AMAMS4            ;;HIGH ADDRESS,M.S.BYTE
001301      000           $MTYP4:  .BYTE    AMTYP4            ;;MEM.TYPE,BLK#4
001302   000000           $MADR4:  .WORD    AMADR4            ;;MEM.LAST ADDRESS,BLK#4
001304   000000           $VECT1:  .WORD    AVEC↑BT1                ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001306   000000           $VECT2:  .WORD    AVECT2            ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001310   160010           $BASE:   .WORD    ABASE             ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001312   000000           $DEVM:   .WORD    ADEVM             ;;DEVICE MAP
001314   000000           $CDW1:   .WORD    ACDW1             ;;CONTROLLER DESCRIPTION WORD#1
001316   000000           $CDW2:   .WORD    ACDW2             ;;CONTROLLER DESCRIPTION WORD#2
001320   000000           $DDW0:   .WORD    ADDW0             ;;DEVICE DESCRIPTOR WORD#0
001322   000000           $DDW1:   .WORD    ADDW1             ;;DEVICE DESCRIPTOR WORD#1
001324   000000           $DDW2:   .WORD    ADDW2             ;;DEVICE DESCRIPTOR WORD#2
001326   000000           $DDW3:   .WORD    ADDW3             ;;DEVICE DESCRIPTOR WORD#3
001330   000000           $DDW4:   .WORD    ADDW4             ;;DEVICE DESCRIPTOR WORD#4
001332   000000           $DDW5:   .WORD    ADDW5             ;;DEVICE DESCRIPTOR WORD#5
001334   000000           $DDW6:   .WORD    ADDW6             ;;DEVICE DESCRIPTOR WORD#6
001336   000000           $DDW7:   .WORD    ADDW7             ;;DEVICE DESCRIPTOR WORD#7
001340   000000           $DDW8:   .WORD    ADDW8             ;;DEVICE DESCRIPTOR WORD#8
001342   000000           $DDW9:   .WORD    ADDW9             ;;DEVICE DESCRIPTOR WORD#9
```

I03

```
   463  001344  000000                     $DDW10: .WORD    ADDW10  ;;DEVICE DESCRIP↑BTOR WORD#10
   464  001346  000000                     $DDW11: .WORD    ADDW11  ;;DEVICE DESCRIPTOR WORD#11
   465  001350  000000                     $DDW12: .WORD    ADDW12  ;;DEVICE DESCRIPTOR WORD#12
   466  001352  000000                     $DDW13: .WORD    ADDW13  ;;DEVICE DESCRIPTOR WORD#13
   467  001354  000000                     $DDW14: .WORD    ADDW14  ;;DEVICE DESCRIPTOR WORD#14
   468  001356  000000                     $DDW15: .WORD    ADDW15  ;;DEVICE DESCRIPTOR WORD#15
   469
   470
   471  001360                             $ETEND:
   472
```

```
                                   .SBTTL  ERROR POINTER TABLE

                                   ;*THIS TABLE CONTAINS THE INFORMATION FOR↑B 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


        001360                     $ERRTB:

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

        001360  000000             NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
        001362  000000             LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT DATA

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

        001364  000377             LINE:   377                     ;DEFAULT ALL EIGHT LINES RUNNING
        001366  017470             PAR:    17470                   ;PARAMETERS: 8 BITS/CHAR,2 STOP BITS,19200 BAUD,NO PARIT
        001370  000000             MODE:   0                       ;DEFAULT MAINTENANCE MODE
        001372  000000             SAVLIN: 0                       ;LINE NUMBER
        001374  000000             XMTLIN: 0                       ;TRANSMISSION LINE NUMBER
        001376  000000             XMTCNT: 0                       ;COU↑BNT OF WORDS IN A TRANSMISSION PATTERN
        001400  000000             REGIST: 0                       ;DEVICE ADDRESS STORAGE LOCATION
        001402  000000             SAVPC:  0                       ;PROGRAM COUNTER STORAGE
        001404  000001             DZACTV: .BLKW   1               ;*DZ11'S SELECTED ACTIVE.
        001406  000001             RUN:    1                       ;*POINTER ONE PAST RUNNING DEVICE.
        001410  000001             DZNUM:  .BLKB 1                 ;*OCTAL NUMBER OF DZ11'S.
        001411     001             SAVNUM: .BYTE 1                 ;*WORKABLE NUMBER.
                                   .EVEN
        001412  001500             ACTIVE: DZ.MAP                  ;TABLE POINTER.
```

```


                                              ;↑BPROGRAM CONTROL FLAGS
                                              ;----------------------

001414      000                         EIAFLG: .BYTE   0                       ;0=EIA  100000=20MA
001415      000                         INIFLG: .BYTE   0                       ;PROGRAM INITIALIZATION FLAG
001416      000                         HDRFLG: .BYTE   0                       ;PROGRAM INITIALIZATION FLAG FOR HEADER MAP
001417      000                         MNTFLG: .BYTE   0                       ;MAINTENANCE BIT SET FLAG
001420      000                         DONFLG: .BYTE   0                       ;TRANSMISSION COMPLETION FLAG
         001422                         .EVEN
                                        ;DATA VARIABLES
001422   000000                         TD0:    .WORD   0
001424   000000                         TD1:    .WORD   0
001426   000000                         TD2:    .WORD   0
001430   000000                         TD3:    .WORD   0
001432   000000                         TD4:    .WORD   0
001434   000000                         TD5:    .WORD   0
001436   000000                         TD6:    .WORD   0
001440   000000                         TD7:    .WORD   0
001442   000000                         TR0:    .WORD   0
001444   000000                         TR1:    .WORD   0
001446   000000                         TR2:    .WORD   0
001450   000000                         TR3:    .WORD   0
001452   000000                         TR4:    .WORD   0
001454   000000                         TR5:    .WORD   0
001456   000000                         TR6:    .WORD   0
001460   000000                         TR7:    .WORD   0
001462                                  STOP:
                                        .SBTTL  APT PARAMETER BLOCK

                                        ;;************************************************************
                                        ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                          ↑B            ;;************************************************************
         001462                                 .$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   001462                                 $APTHDR ;;POINT TO APT HEADER BLOCK
         001462                                 .=.$X   ;;RESET LOCATION COUNTER
                                        ;;************************************************************
                                        ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                        ;INTERFACE SPEC.

001462                                  $APTHD:
00146↑B2                000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001464   001234                         $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001466   000132                         $TSTM:  .WORD   90.     ;;RUN TIM OF LONGEST TEST
001470   000137                         $PASTM: .WORD   95.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001472   000137                         $UNITM: .WORD   95.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001474   000052                                 .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
                                                ;DZ11 STATUS TABLE AND ADDRESS ASSIGNMENTS
                                                ;----------------------------------------

         001500                         .=1500
001500                                  DZ.MAP:

```

```
001500  000001                  DZCR0:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 0
001502  000001                  DZVC0:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 0
001504  000001                  DZLV0:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001506  000001                  LINE0:  .BLKW   1       ;ALL LINES SELECTED
001510  000001                  PAR0:   .BLKW   1       ;PARAMETERS
001512  000001                  MANT0:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001514  000001                  DZCR1:  .BLKW   1       ;CONTROL STATUS REG↑BISTER FOR DZ11 NUMBER 1
001516  000001                  DZVC1:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 1
001520  000001                  DZLV1:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001522  000001                  LINE1:  .BLKW   1       ;ALL LINES SELECTED
001524  000001                  PAR1:   .BLKW   1       ;PARAMETERS
001526  000001                  MANT1:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001530  000001                  DZCR2:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 2
001532  000001                  DZVC2:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 2
001534  000001                  DZLV2:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001536  000001   ↑B             LINE2:  .BLKW   1       ;ALL LINES SELECTED
001540  000001                  PAR2:   .BLKW   1       ;PARAMETERS
001542  000001                  MANT2:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001544  000001                  DZCR3:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 3
001546  000001                  DZVC3:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 3
001550  000001                  DZLV3:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001552  000001                  LINE3:  .BLKW   1       ;ALL LINES SELECTED
001554  000001                  PAR3:   .BLKW   1       ;PARAMETERS
001556  000001                  MANT3:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001560  000001                  DZCR4:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 4
001562  000001                  DZVC4:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 4
001564  000001                  DZLV4:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001566  000001                  LINE4:  .BLKW   1       ;ALL LINES SELECTED
001570  000001                  PAR4:   .BLKW   1       ;PARAMETERS
001572  000001                  MANT4:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001574  000001                  DZCR5:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 5
001576  000001                  DZVC5:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 5
001600  000001                  DZLV5:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
↑B        001602  000001                LINE5:  .BLKW   1       ;ALL LINES SELECTED
001604  000001                  PAR5:   .BLKW   1       ;PARAMETERS
001606  000001                  MANT5:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001610  000001                  DZCR6:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 6
001612  000001                  DZVC6:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 6
001614  000001                  DZLV6:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001616  000001                  LINE6:  .BLKW   1       ;ALL LINES SELECTED
001620  000001                  PAR6:   .BLKW   1       ;PARAMETERS
001622  000001                  MANT6:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

001624  00000↑B1                        DZCR7:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 7
001626  000001                  DZVC7:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 7
001630  000001                  DZLV7:  .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
001632  000001                  LINE7:  .BLKW   1       ;ALL LINES SELECTED
001634  000001                  PAR7:   .BLKW   1       ;PARAMETERS
001636  000001                  MANT7:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

```

```
 001640  000001            DZCR10: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 10
 001642  000001            DZVC10: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 10
 001644  000001            DZLV10: .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
 001646  000001            LINE10: .BLKW   1       ;ALL LINES SELECTED
 001650  000001            PAR10:  .BLKW   1       ;PARAMETERS
 001652  000001            MANT10: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

 001654  000001            DZCR11: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 11
 001656  000001            DZVC11: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 11
 001660  000001            DZLV11: .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
 001662  000001            LINE11: .BLKW   1       ;ALL LINES SELECTED
6↑B35   001664  000001            PAR11:  .BLKW   1       ;PARAMETERS
 001666  000001            MANT11: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

 001670  000001            DZCR12: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 12
 001672  000001            DZVC12: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 12
 001674  000001            DZLV12: .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
 001676  000001            LINE12: .BLKW   1       ;ALL LINES SELECTED
 001700  000001            PAR12:  .BLKW   1       ;PARAMETERS
 001702  000001            MANT12: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

 001704  000001            DZCR13: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ↑B11 NUMBER 13
 001706  000001            DZVC13: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 13
 001710  000001            DZLV13: .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
 001712  000001            LINE13: .BLKW   1       ;ALL LINES SELECTED
 001714  000001            PAR13:  .BLKW   1       ;PARAMETERS
 001716  000001            MANT13: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

 001720  000001            DZCR14: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 14
 001722  000001            DZVC14: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 14
 001724  000001            DZLV14: .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
 001726  000001            LINE14: .BLKW   1       ;ALL LINES SELECTED
 001730  000001            PAR14:  .BLKW   1       ;PARAMETERS
 001732  000001            MANT14: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

 001734  000001            DZCR15: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 15
 001736  000001            DZVC15: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 15
 001740  000001            DZLV15: .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
 001742  000001            LINE15: .BLKW   1       ;ALL LINES SELECTED
 001744  000001            PAR15:  .BLKW   1       ;PARAMETERS
 001746  000001            MANT15: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

 001750  000001            DZCR↑B16:       .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 16
 001752  000001            DZVC16: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 16     ;
 001754  000001            DZLV16: .BLKW   1       ;PRIORITY LEVEL AND EIA FLAG SELECTOR
 001756  000001            LINE16: .BLKW   1       ;ALL LINES SELECTED
 001760  000001            PAR16:  .BLKW   1       ;PARAMETERS
 001762  000001            MANT16: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

 001764  000001            DZCR17: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZ11 NUMBER 17
 001766  000001            DZVC17: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZ11 NUMBER 17
 001770  000001            DZLV17: .BLKW   1       ;PRIORITY LEVEL A↑BND EIA FLAG SELECTOR
 001772  000001            LINE17: .BLKW   1       ;ALL LINES SELECTED
 001774  000001            PAR17:  .BLKW   1       ;PARAMETERS
 001776  000001            MANT17: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
```

```
MD-11-DZDZA-C   MACY11 27(1006)  21-OCT-76  13:09  PAGE 16                                                    PAGE:  0039
DZDZAC.P11    21-OCT-76 13:07            APT PARAMETER BLOCK

   680  002000  177777                   DZ.END: 177777
```

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

                          ;;*******************************************************************
                          ;-------------------------------------------------------------------
   002002                 .TRPTAB:
             104400       ADVANCE=TRAP+0          ;CALL TO ADVANCE TO NEXT TEST( OR SCOPE THIS ONE)
   002002    006606               .ADVANCE
             104401       SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
   002004    005120               .SCOP1
             104402       TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
   002006    005144               .TYPE
             104403       INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
   002010    005712               .INSTR
             104404      †BINSTER=TRAP+4          ;CALL TO INPUT ERROR HANDLER
   002012    006016               .INSTER
             104405       PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
   002014    006036               .PARAM
             104406       SETFLG=TRAP+6           ;CALL TO  SET FLAG ROUTINE
   002016    010362               .SETFLG
             104407       SAVOS=TRAP+7            ;CALL TO REGISTER SAVE ROUTINE
   002020    006236               .SAVOS
             104410       RESOS=TRAP+10           ;CALL TO REGISTER RESTORE ROUTINE
   002022    006276               .RESOS
             104411       CONVRT=TRAP+11          ;CALL TO DATA OUTPUT ROUTINE
   002024    006330               .CONVRT
             104412       CNVRT=TRAP+12           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
†B 002026    006334               .CNVRT
             104413       DEVICE.CLR=TRAP+13              ;CALL TO ISSUE A DEVICE CLEAR
   002030    006534               .DEVICE.CLR
             104414       DELAY=TRAP+14           ;CALL TO DELAY FOR FAST CPU'S
   002032    006566               .DELAY
             104415       PARMD=TRAP+15           ;CONVERT DECIMAL STRING TO OCTAL
   002034    024654               .PARMD
             104416       PAWCH=TRAP+16           ;SET FLAG    ECHO OR  CABLE
   002036    025050               .PAWCH
             104417       DCLASM=TRAP+17          ;CLEAR DEVICE, SET MAINT. BIT IF I MODE
   002040    006554               .DCLASM

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

```
 723                                             ;DZ11 VECTOR AND REGISTER INDIRECT POINTERS
 724                                             ;WORKING AREA
 725
 726   002042  160040                   DZCSR:   160040   ;R/W
 727   002044  160041                   HDZCSR:  160041   ;R/W
 728   002046  160042                   DZRBUF:  160042   ;READ ONLY
 729   002050  160043                   HDZRBUF:          160043  ;READ ONLY
 730   002052  160042                   DZLPR:   160042   ;WRITE ONLY
 731   002054  160043                   HDZLPR:  160043   ;WRITE ONLY
 732   002056  160044                   DZTCR:   160044   ;R/W
 733   002060  160045                   HDZTCR:  160045   ;↑BR/W
 734   002062  160046                   DZMSR:   160046   ;READ ONLY
 735   002064  160047                   HDZMSR:  160047   ;READ ONLY
 736   002066  160046                   DZTDR:   160046   ;WRITE ONLY
 737   002070  160047                   HDZTDR:  160047   ;WRITE ONLY
 738                                            ;DEFAULT DZ VECTORS
 739   002072  000300                   DZRIV:   300      ;REC INTR VECTOR
 740   002074  000302                   DZRIS:   302      ;REC INTR STATUS
 741   002076  000304                   DZTIV:   304      ;XMIT INTR VECTOR
 742   002100  000306                   DZTIS:   306      ;XMIT INTR STATUS
 743
 744
```

```
745
746                                                  ;TIME TABLE FOR RELATIVE TIMING TESTS
7↑B47                                                         ;-----------------------------------
748
749  002102                              TMTBL:
750  002102  000000                      T50:     0
751  002104  000000                      T75:     0
752  002106  000000                      T110:    0
753  002110  000000                      T134:    0
754  002112  000000                      T150:    0
755  002114  000000                      T300:    0
756  002116  000000                      T600:    0
757  002120  000000                      T1200:   0
758  002122  000000                      T1800:   0
759  002124  000000                      T2000:   0
760  002126  000000                      T2400:   0
761  002130  000000                      T3600:   0
762  002132  000000                      T4800:   0
763  002134  000000                      T7200:   0
764  002136  000000                      T9600:   0
765  002140  000000                      TEIGHT:0
766  002142  000000                      TSEVEN:  0
767  002144  000000                      TSIX:    0
768  002146  000000                      TFIVE:   0
```

```

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

002150                                 .START:
002150  000005                                 RESET                     ;CLEAR THE WORLD. START NEW ENVIRONMENT
002152  012706  001120                         MOV     #STACK,SP         ;SET UP STACK
002156  106427  000340                         MTPS    #PR7              ;LOC↑BK OUT INTERRUPTS
002162  012737  007530  000024                 MOV     #$PWRDN,@#24      ;SET UP POWER FAIL VECTOR
002170  113737  001410  001411                 MOVB    DZNUM,SAVNUM      ;SAVE NUMBER OF DEVICES IN SYSTEM.
002176  005037  001242                         CLR     $PASS             ;CLEAR PASS COUNT
002202  105037  001123                         CLRB    $ERFLG            ;CLEAR ERROR FLAG
002206  012737  001500  001412                 MOV     #DZ.MAP,ACTIVE    ;GET MAP POINTER.
002214  012737  000001  001406                 MOV     #1,RUN            ;POINT POINTER TO FIRST DEVICE.
002222  005037  001132                         CLR     $ERTTL            ;CLEAR ERROR COUNT
002226  005037  001136                         CLR     $ERRPC            ;CLEAR LAST ERROR POINTER
002232  005037  001122                         CLR     $TSTNM            ;S↑BET UP FOR TEST 1
002236  012737  002150  001126                 MOV     #.START,$LPADR    ;SET UP FOR POWER FAIL BEFORE
                                                                         ;TESTING STARTS
                                               ;SET UP FOR SMALL 11 SWITCH REGISTER COMPATIBILITY
002244  013746  000006                         MOV     6,-(SP)           ;SAVE BUS ERROR PS
002250  013746  000004                         MOV     4,-(SP)           ;SAVE BUS ERROR PC
002254  012737  002274  000004                 MOV     #20$,4            ;SET UP TO TRAP TO THIS ROUTINE
002262  022777  177777  176670                 CMP     #-1,@SWR          ;CAN 177570 BE REFERENCED?
002270  001402                                 BEQ     22$               ;IF SO AND IT IS -1, TREAT LIKE SWITCHLESS
002272  000407                                 BR      21$               ;IF YES,SKIP AROUND THE SETUP
002274  022626                         20$:    POP2SP                    ;REMOVE THE TRAP FROM THE STACK
002276  012737  000176  001160         22$:    MOV     #SWREG,SWR        ;IF NO,TRAP COMES HERE.POINT TO SOFTWARE SWR
002304  012737  000174  001162                 MOV     #DISPREG,DISPLAY  ;POINT TO SOFTWARE DISPLAY REGISTER
002312  012637  000004                 21$:    MOV     (SP)+,4           ;RESTORE THE BUS ERROR VECTOR
002316  012637  000006                         MOV     (SP)+,6
002322  005737  000042                         TST     42                ;WORKING UNDER A MONITOR ?
002326  001402                                 BEQ     31$               ;NO
002330  000137  004126                         JMP     63$               ;IF YES,SKIP THE TERMINAL INTERROGATION
002334  105737  001415                 31$:    TSTB    INIFLG            ;HAVE WE ALREADY BEEN HERE TODAY?
        002340  001004                                 BNE     29$               ;IF SO, SKIP PRINTING THE TITLE
002342  104402  001000                         TYPE    ,MTITLE           ;PRINT THE DIAGNOSTIC'S TITLE
002346  105337  001415                         DECB    INIFLG            ;SET THE ONCE ONLY FLAG
002352  105737  001255                 29$:    TSTB    $ENVM             ;DETERMINE WHETHER APT SIZING SHOULD BE DONE
002356  100004                                 BPL     30$               ;IF NOT, GO CHECK FOR AUTO-SIZING
002360  004737  011310                         JSR     PC,SETAPT         ;OTHERWISE, GO DO APT SIZING FROM ETABLE
002364  000137  004152                         JMP     16$               ;GO PRINT DZ STATUS TABLE
002370  032777  000001  176562         30$:    BIT     #SW00,@SWR        ;RESELECT ?
002376  001011                                 BNE     32$               ;IF YES, GO SET UP THE↑B INFORMATION
002400  122737  000377  001415                 CMPB    #377,INIFLG       ;ON 1ST START; MUST ANSWER QUESTION
002406  001003                                 BNE     .+10              ;IF NOT ANSWERING QUESTIONS
002410  105777  176544                         TSTB    @SWR              ;ARE U AUTO SIZING?
002414  100402                                 BMI     32$               ;NO AUTO SIZE! NO SW00=1 ON 1ST START!
002416  000137  003104                         JMP     73$               ;IF NO, SKIP THE INTERROGATION
002422  012700  001500                 32$:    MOV     #DZ.MAP,R0        ;POINT TO THE BEGINNING OF THE MAP TABLE
002426  105037  001416                         CLRB    HDRFLG            ;MAKE SURE A MAP GETS PRINTED
002432  005020                         65$:    CLR     (R0)+             ;CLEAR A TABLE LOCATION
```

```
002434  020027  002000                  CMP     R0,#DZ.END          ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
002440  001374                          BNE     65$                 ;IF NOT ,CLEAR THE NEXT LOCATION IN THE TABLE
002442  105337  001415                  DECB    INIFLG              ;INSURE NO AUTO SIZING IF QUESTIONS ANSWERED!

                                ;THE FOLLOWING ARE PARAMETERS USED TO FILL IN THE MAP
                                ;TABLE AND SET UP THE DIAGNOSTIC.

                                        ;GET THE BASE ADDRESS OF THE DZ11'S

002446                          33$:
002446  104403                          INSTR                       ;CALL THE STRING INPUT ROUTINE
002450  0033↑B24                                66$                         ;POINTER TO MESSAGE TO BE PRINTED
002452  104405                          PARAM                       ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
002454  160000                          160000                      ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002456  163770                          163770                      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002460  001500                          DZCR0                       ;POINTER TO MAP LOCATION TO BE FILLED
002462     007                          .BYTE   7                   ;MASK OF INVALID BITS FOR THIS PARAMETER
002463     001                          .BYTE   1                   ;NUMBER OF PARAMETERS TO STORE
002464  013737  001500  001310          MOV     DZCR0,$BASE         ;COPY BASE ADDRESS TO ETABLE

                                        ;GET THE BASE VECTOR ADDRESS

↑B002472                                34$:
002472  104403                          INSTR                       ;CALL THE STRING INPUT ROUTINE
002474  003370                          67$                         ;POINTER TO MESSAGE TO BE PRINTED
002476  104405                          PARAM                       ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
002500  000300                          300                         ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002502  000776                          776                         ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002504  001502                          DZVC0                       ;POINTER TO MAP LOCATION TO BE FILLED
002506     003                          .BYTE   3                   ;MASK OF INVALID BITS FOR THIS PARAMETER
002507     001                          .BYTE   1                   ;NUMBER OF PARAMETERS TO STORE
002510  013737  001502  001304          MOV     DZVC0,$VECT1        ;COPY VECTOR TO ETABLE

                                        ;GET THE BUS REQUEST LEVEL

002516  104403                          INSTR                       ;CALL THE STRING INPUT ROUTINE
002520  003431                          68$                         ;POINTER TO MESSAGE TO BE PRINTED
002522  104405                          PARAM                       ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
002524  000004                          4                           ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002526  000007                          7                           ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002530  001504                          DZLV0                       ;POINTER TO MAP LOCATION TO BE FILLED
002532     000                          .BYTE   0                   ;MASK OF INVALID BITS FOR THIS PARAMETER
002533     001                          .BYTE   1                   ;NUMBER OF PARAMETERS TO STOR↑BE
002534  113737  001504  001305          MOVB    DZLV0,$VECT1+1      ;GET BUS REQUEST LEVEL INTO ETABLE
002542  106337  001305                  ASLB    $VECT1+1            ;ALIGN THE BITS PROPERLY
002546  106337  001305                  ASLB    $VECT1+1            ;ALIGN THE BITS PROPERLY
002552  106337  001305                  ASLB    $VECT1+1            ;ALIGN THE BITS PROPERLY
002556  106337  001305                  ASLB    $VECT1+1            ;ALIGN THE BITS PROPERLY
002562  106337  001305                  ASLB    $VECT1+1            ;ALIGN THE BITS PROPERLY

                                        ;FIND OUT IF MODULE IS EIA OR 20 MA.

002566  104402  004012                  TYPE    ,74$                ;PRINT EIA MESSAGE
002572  005037  001220                  CLR     $TMP1               ;USE $TMP1
002576  1↑B05777          176362        80$:    TSTB    @$TKS               ;IS KEYBOARD DONE?
002602  100375                          BPL     80$                 ;IF NOT, WAIT FOR IT
```

```
002604  017746  176356                 MOV     @$TKB,-(SP)     ;IF YES, PUT CHARACETR ON STACK
002610  042716  000240                 BIC     #240,(SP)       ;STRIP DOWN CHARACTER
002614  122726  000015                 CMPB    #15,(SP)+       ;IS IT ?
002620  001414                         BEQ     81$             ;IF SO, GET OUT
002622  014677  176344                 MOV     -(SP),@$TPB     ;IF NOT, PRINT CHARACTER
002626  042737  100000  001504         BIC     #BIT15,DZLV0    ;CLEAR EIA FLAG
002634  122726  000102                 CMPB    #102,(SP)+      ;IS IT A B?
002640  001356                         BNE     80$             ;IF NOT, GO BACK FOR INPUT
002642  052737  100000  001504         BIS     #BIT15,DZLV0    ;IF SO, SET FLAG
002650  000752                         BR      80$             ;GET MORE INPUT
002652                          81$:

                                       ;GET THE MODE OF OPERATION (E,I,S)

002652  104403                         INSTR                   ;CALL THE STRING INPUT ROUTINE
002654  003642                         72$                     ;POINTER TO THE MESSAGE TO BE PRINTED
002656  104406                         SETFLG                  ;CALL THE MAINTENANCE FLAG SETUP ROUTINE
002660  001512                         MANT0                   ;THIS IS THE FLAG BEING SETUP

                                       ;GET THE NUMBER OF DZ11'S RUNNING↑B

002662  104403                         INSTR                   ;CALL THE STRING INPUT ROUTINE
002664  003600                         71$                     ;POINTER TO MESSAGE TO BE PRINTED
002666  104405                         PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
002670  000001                         1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002672  000020                         16.                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002674  001220                         $TMP1                   ;POINTER TO MAP LOCATION TO BE FILLED
002676     000                         .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
002677     001                         .BYTE   1               ;NUMBER OF PARAMETERS TO STORE

002700  012737  000377  001506         MOV     #377,LINE0      ;SET↑B UP DEFAULT LINES
002706  012737  017470  001510         MOV     #17470,PAR0     ;SET UP DEFAULT LPR PARAMETER
                                                               ;RECEIVER ON; 19.2 KBAUD; 2STOP BITS; 8 BIT/CHAR
002714  032777  000010  176236         BIT     #SW03,@SWR      ;DO YOU WANT PARAMETERS?
002722  001402                         BEQ     40$             ;IF NO, SKIP THE PARAMETER CALL
002724  004737  003134                 JSR     PC,23$          ;GET PARAMETERS
002730  012737  000001  001312  40$:   MOV     #1,$DEVM        ;INITIALIZE ACTIVE DEVICE SELECTION PARAMETER
002736  113737  001220  001410         MOVB    $TMP1,DZNUM     ;COPY THE NUMBER OF DEVICES
002744  113737  001220  001411         MOVB    $TMP1,SAVNUM    ;COPY A BACKUP NUMBER
002752  005337  001220          62$:   DEC     $TMP1           ;$TMP1 CONTAINS THE COUNT OF UNINITIALIZED
002756  001404                         BEQ     61$             ; SELECTED DEVICES
002760  000261                         SEC                     ;SET A BIT FLAG TO INDICATE AN ACTIVE DEVICE
002762  006137  001312                 ROL     $DEVM           ;POINT TO THE NEXT DEVICE
002766  000771                         BR      62$             ;GO DO THIS PROCEDURE AGAIN
002770  013737  001312  001222  61$:   MOV     $DEVM,$TMP2     ;# OF TIMES
002776  013737  001312  001404         MOV     $DEVM,DZACTV    ;COPY THE ACTIVE DEVICE PARAMETER
003004  012700  001500                 MOV     #DZCR0,R0       ;SET A POINTER TO THE SPECIFIED INFORMATION
003010  012701  001514                 MOV     #DZCR1,R1       ;POINT R1 TO T↑BHE REST OF THE MAP TABLE
003014  012702  001320                 MOV     #$DDW0,R2       ;POINT TO ETABLE'S DEVICE DESCRIPTOR WORDS
003020  000241                         CLC                     ;INITIALIZE THE "C" BIT FOR A ROTATION
003022  006037  001222                 ROR     $TMP2           ;SKIP MAPPING SETUP FOR DEVICE 0- IT'S DONE
003026  006237  001222          64$:   ASR     $TMP2           ;ISOLATE A SELECTION FLAG IN THE "C" BIT
003032  103404                         BCS     41$             ;IS THIS DEVICE SELECTED? IF YES, GO LOAD TABLE
003034  012711  177777                 MOV     #-1,(R1)        ;TERMINATE THE LIST
003040  000137  004126                 JMP     63$             ;GO TO THE NEXT BLOCK
003044  012011                  41$:   MOV     (R0)+,(R1)      ;ADDRESS
```

```
003046  062721  000010                  ADD     #10,(R1)+           ;POINT TO THE NEXT DZ11 ADDRESS VALUE
003052  012011                          MOV     (R0)+,(R1)          ;VECTOR
003054  062721  000010                  ADD     #10,(R1)+           ;POINT TO THE NEXT VECTOR VALUE
003060  012021                          MOV     (R0)+,(R1)+         ;LEVEL
003062  012021                          MOV     (R0)+,(R1)+         ;LINES
003064  016012  177774                  MOV     -4(R0),(R2)         ;GET THE EIA FLAG FROM THE PRIORITY WORD
003070  042712  077777                  BIC     #77777,(R2)         ;ISOLATE THAT FLAG
003074  051022                          BIS     (R0),(R2)+          ;ADD PARAMETERS TO DEVICE DESCRIPTOR WORD
003076  012021                          MOV     (R0)+,(R1)+         ;PARAMETERS
003100  012021                          MOV     (R0)+,(R1)+         ;MAINTENANCE MODE
003102  000751                          BR      64$
003104  032777  000010  176046  73$:    BIT     #SW03,@SWR          ;ASK PARAMETERS ?
003112  001002                          BNE     42$                 ;IF NO, GO DO AUTO SIZING
003114  000137  004126                  JMP     63$                 ;GO SET UP FOR AUTO SIZING
003120  004737  003134          42$:    JSR     PC,23$              ;GO ASK PARAMETERS
003124  105337  001415                  DECB    INIFLG              ;INSURE NO AUTO SIZE IF QUESTIONS ANSWERED
003130  000137  004152                  JMP     16$                 ;GO TO THE NEXT BLOCK

                                        ;GET THE ACTIVE LINES PARAMETER

0031!B34                                23$:
003134  104403                          INSTR                       ;CALL THE STRING INPUT ROUTINE
003136  003454                          69$                         ;POINTER TO MESSAGE TO BE PRINTED
003140  104405                          PARAM                       ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
003142  000001                          1                           ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
003144  000377                          377                         ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
003146  001506                          LINE0                       ;POINTER TO MAP LOCATION TO BE FILLED
003150     000                          .BYTE   0                   ;MASK OF INVALID BITS FOR THIS PARAMETER
003151     001                          .BYTE   1                   ;NUMBER OF PARAMETERS TO STORE
003152  105037  001416                  CLRB    HDRFLG              ;MAKE SURE THE CHA!BNGES ARE PRINTED

                                ;THIS SEGMENT CHECKS TO MAKE SURE THE LINE PARAMETER JUST ENTERED
                                ;IS LEGITIMATE IN STAGGERED MODE OPERATION IF THAT MODE WAS SELECTED

003156  005737  001512                  TST     MANT0               ;IS STAGGERED THE MODE OF OPERATION?
003162  100021                          BPL     26$                 ;IF NOT, SKIP THIS SEGMENT
003164  013703  001506                  MOV     LINE0,R3            ;GET A SCRATCH COPY OF THE ACTIVE LINES
003170  006003                  24$:    ROR     R3                  ;GET A LINE SELECTION BIT(EVEN NUMBER LINE)
003172  103410                          BCS     25$                 ;IF IT IS SELECTED, CHECK TO SEE IF THE NEXT IS TOO
003174  001414                          BEQ     26$                 ;IF ALL HAVE BEEN CHECKED, CONTINUE PROCESSING
003176  006203                          ASR     R3                  ;IF IT IS 0,CHECK TO SEE IF THE NEXT IS TOO
003200  103373                          BCC     24$                 ;IF THIS ONE'S 0 TOO, GO CHECK THE NEXT PAIR
003202  104402  001230          27$:    TYPE    ,$QUES              ;THIS IS AN INCORRECT PARAMETER
003206  104402  010306                  TYPE    ,MBADLN             ;LET THE USER KNOW ABOUT IT
003212  000750                          BR      23$                 ;GO GET THE CORRECT PARAMETER
003214  001772                  25$:    BEQ     27$                 ;IF ANOTHER FLAG ISN'T SET, THERE'S AN ERROR
003216  006203                          ASR     R3                  ;GET THE NEXT FLAG
003220  103370                          BCC     27$                 ;IF IT ISN'T SET, THERE'S AN ERROR
003222  000241                          CLC                         ;INITIALI!BZE THE "C" BIT FOR TESTING OF THE NEXT PAIR
003224  000761                          BR      24$                 ;GO TEST THE NEXT PAIR OF FLAGS

                                        ;GET THE LINE PARAMETER REGISTER ARGUMENT

003226                          26$:
003226  104403                          INSTR                       ;CALL THE STRING INPUT ROUTINE
003230  003530                          70$                         ;POINTER TO MESSAGE TO BE PRINTED
```

```
003232  104405                          PARAM                           ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
003234  000000                          0                               ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
003236  000017                          17              ↑B              ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
003240  001510                          PARO                            ;POINTER TO MAP LOCATION TO BE FILLED
003242     000                          .BYTE   0                       ;MASK OF INVALID BITS FOR THIS PARAMETER
003243     001                          .BYTE   1                       ;NUMBER OF PARAMETERS TO STORE
003244  012702  001506                  MOV     #LINEO,R2               ;POINT TO THE LINE SELECTION PARAMETER
003250  012703  001510                  MOV     #PARO,R3                ;POINT TO THE CHOSEN PARAMETERS
003254  011304                          MOV     (R3),R4                 ;USE BAUD RATE AS AN INDEX IN DELAY TABLE
003256  006304                          ASL     R4                      ;ALIGN INDEX ON WORD BOUNDARY
003260  016437  030130  006604          MOV     DLYTBL(R4),DLYCNT       ;SET THE DELAY COUNT FOR THIS BAUD RATE
003266  000313                          SWAB    (R3)                    ;PLACE IN HIGH BYTE
003270  052713  010070                  BIS     #10070,(R3)             ;PLACE EXTRA PARAMETERS INTO LOC
003274  011262  000014          28$:    MOV     (R2),14(R2)             ;LOAD THE LINES
003300  011363  000014                  MOV     (R3),14(R3)             ;LOAD THE PARAMETERS
003304  062702  000014                  ADD     #14,R2                  ;POINT TO THE NEXT SET
003310  062703  000014                  ADD     #14,R3                  ; .. OF BOTH PARAMETERS
003314  020327  001774                  CMP     R3,#PAR17               ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
003320  001365                          BNE     28$                     ;IF NOT, GO LOAD SOME MORE PARAMETEPS
003322  000207                          RTS     PC                      ;RETURN TO CALLING BLOCK↑B
003324  030600  052123  041440  66$:    .ASCIZ  <200>/1ST CSR ADDRESS (160000:163700):  /
003370  030600  052123  053040  67$:    .ASCIZ  <200>/1ST VECTOR ADDRESS (300:770):  /
003431     200  051102  046040  68$:    .ASCIZ  <200>/BR LEVEL (4:6):  /
003454  046200  047111  051505  69$:    .ASCIZ  <200>/LINES ACTIVE BY BIT <IN OCTAL>(001:377):  /
003530  042200  043105  052501  70$:    .ASCIZ  <200>/DEFAULT BAUD RATE <IN OCTAL>(00:17):  /
003600  021600  047440  020106  71$:    .ASCIZ  <200>/# OF DZ11'S <IN OCTAL> (1:20):  /
003642  046600  044501  052116  72$:    .ASCII  <200>/MAINTENANCE MODE/
003663     200  055440  054105          ↑B.ASCII        <200>/ [EXTERNAL  <H325>     (E)]/
003717     200  055440  047111          .ASCII  <200>/ [INTERNAL  <DZCSR03=1>(I)]/
003753     200  055440  052123          .ASCIZ  <200>/ [STAGGERED <H327>     (S)]: /
004012  052200  050131  020105  74$:    .ASCIZ  <200>/TYPE "A" FOR EIA MODULE OR "B" FOR 20 MA (A:B): /
004074  042600  052116  051105  75$:    .ASCIZ  <200>/ENTER DELAY PARAMETER: /
        004126                          .EVEN
004126                          63$:
004126  122737  000377  001415          CMPB    #377,INIFLG             ;ONLY DO AUTO SIZE ON 1ST START
004134  001006                          BNE     16$
004136  032777  000200  175014          BIT     #BIT7,@SWR              ;BIT7=1??
004144  001002                          BNE     16$                     ;BR IF NO AUTO SIZE
004146  004737  011462                  JSR     PC,AUTO.SIZE            ;GO DO THE AUTO SIZE
004152  105737  001416          16$:    TSTB    HDRFLG                  ;HAS THE TABLE BEEN TYPED YET?
004156  001021                          BNE     1$                      ;IF SO, DON'T TYPE IT AGAIN
004160  105337  001416                  DECB    HDRFLG                  ;INDICATE THAT THE TABLE WILL BE TYPED
004164  104402  010261                  TYPE    ,XHEAD                  ;TYPE MAP HEADER
004170  012700  001500                  MOV     #DZ.MAP,R0              ;SET POINTER
004174  010037  001220          5$:     MOV     R0,$TMP1                ;POINT TO THE MAP LOCATION
004200  012037  001222                  MOV     (R0)+,$TMP2             ;SET DATA
004204  022737  177777  001222          CMP     #-1,$TMP2               ;END OF LIST?
↑B              004212  001403          BEQ     1$                      ;BR IF YES
004214  104411                  17$:    CONVRT                          ;CALL THE OCTAL TO ASCII CONVERSION ROUTINE
004216  010350                          XSTATQ                          ;CONVERT THE DATA AT THIS ADDRESS
004220  000765                          BR      5$                      ;GO PRINT THE NEXT PARAMETER
004222  005737  000042          1$:     TST     @#42                    ;IS PROGRAM RUNNING UNDER MONITOR
004226  001026                          BNE     3$                      ;YES
004230  032777  000100  174722          BIT     #SW06,@SWR              ;DESELECT SPECIFIC DEVICES??
004236  001422                          BEQ     3$                      ;BR IF NO.
004240  104402  010202                  TYPE    ,MNEW                   ;TYPE THE MESSAGE.
```

```
004244  005000                                    CLR     R0              ;ZERO DATA DISPLAY
004246  000000                                    HALT                    ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
004250  027737  174704  001312                    CMP     @SWR,$DEVM      ;IS THE NUMBER VALID?
004256  101404                                    BLOS    2$              ;BR IF NUMBER IS OK.
004260  104402  010054                            TYPE    ,MERR3          ;TELL USER OF INVALID NUMBER.
004264  000000                            9$:     HALT                    ;STOP EVERY THING.
004266  000776                                    BR      9$              ;RESTART THE PROGRAM AGAIN.
004270  017737  174664  001404            2$:     MOV     @SWR,DZACTV     ;GET NEW DEVICE PATTERN
004276  013700  001404                            MOV     DZACTV,R0       ;SHOW THE USER WHAT HE SELECTED.
004302  000000                                    HALT                    ;CONTINUE DYNAMIC SWITCHES.
004304  032777  000020  174646            3$:     BIT     #SW04,@SWR      ;CHECK TO SEE IF DELAY COUNT CHANGES
004312  001407                                    BEQ     18$             ;IF NOT, GO CLEAR VECTOR AREA
004314  104403                                    INSTR                   ;CALL THE STRING INPUT ROUTINE
004316  004074                                    75$                     ;POINTER TO MESSAGE TO BE PRINTED
004320  104405                                    PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
004322  000001                                    1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
004324  177777                                    177777                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
004326  006604                                    DLYCNT                  ;POINTER TO MAP LOCATIO↑BN TO BE FILLED
004330     000                                    .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
004331     001                                    .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
004332  012700  000300                    18$:    MOV     #300,R0         ;PREPARE TO CLEAR THE FLOATING
004336  012701  000302                            MOV     #302,R1         ;VECTOR AREA. 300-776
004342  010120                            4$:     MOV     R1,(R0)+        ;START PUTTING "PC+2 - HALT"
004344  005021                                    CLR     (R1)+           ;IN VECTOR AREA.
004346  022021                                    CMP     (R0)+,(R1)+     ;POP POINTERS
004350  022700  001000                            CMP     #1000,R0        ;ALL DONE??
004354  001372                                    BNE     4$              ;BR IF NO.

                                          ;TEST START AND RESTART
                                          ;-----↑B-----------------

004356  012706  001120                    .BEGIN: MOV     #STACK,SP       ;SET UP STACK
004362  106427  000340                            MTPS    #PR7            ;LOCK OUT INTERRUPTS
004366  005737  000042                            TST     @#42            ;IS PROGRAM UNDER MONITOR CONTROL
004372  001015                                    BNE     2$              ;BR IF YES
004374  032777  000004  174556                    BIT     #BIT2,@SWR      ;CHECK FOR LOCK ON TEST
004402  001406                                    BEQ     1$              ;BR IF NO LOCK DESIRED.
004404  104402  010100                            TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
004410  012737  000240  004672                    MOV     #NOP,TTST       ;ADJUST SCOPE ROUTINE.
004416  000403                                    BR      2$              ;CONTINUE ALONG.
004420  013737  005114  004672            1$:     MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
004426  012737  010752  001126            2$:     MOV     #CYCLE,$LPADR   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
004434  104402  007771                            TYPE    ,MR             ;TYPE "RUNNING"
004440  000177  174462                            JMP     @$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

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

    004444                        $EOP:
    004444 000004                         SCOPE
    004446 005037 001136                  CLR     $ERRPC              ;CLEAR LAST ERROR PC
    004452 105037 001123                  CLRB    $ERFLG              ;CLEAR ERROR FLAG
    004456 104402 007745                  TYPE    ,MEPASS             ;TYPE END PASS
    004462 104402 010127                  TYPE    ,MCSRX              ;TYPE CSR
    004466 104412 004624                  CNVRT   ,XCSR               ;SHOW IT
    004472 104402 010135                  TYPE    ,MVECX              ;TYPE VECTOR
    004476 104412 004632                  CNVRT   ,XVEC               ;SHOW IT
    004502 005237 001242          ↑B      INC     $PASS               ;RAISE PASS COUNT
    004506 104402 010143                  TYPE    ,MPASSX             ;TYPE PASSES
    004512 104412 004640                  CNVRT   ,XPASS              ;SHOW IT
    004516 005337 001242                  DEC     $PASS               ;RESTORE PASS COUNT
    004522 104402 010154                  TYPE    ,MERRX              ;TYPE ERRORS
    004526 104412 004646                  CNVRT   ,XERR               ;SHOW IT
    004532 105337 001411                  DECB    SAVNUM              ;ARE ALL DEVICES TESTED?
    004536 001030                         BNE     $DOAGN              ;BR IF NO.
    004540 113737 001410 001411           MOVB    DZNUM,SAVNUM        ;RESTORE THE COUNT
    004546 005037 001226                  CLR     $TIMES              ;;ZERO THE NUMBER OF ITERATIONS
    004552 005237 001242                  INC     $PASS               ;;INCREMENT THE PASS NUMBER
    004556 042737 100000 .001242          BIC     #100000,$PASS       ;;DON'T ALLOW A NEG. NUMBER
    004564 005327                         DEC     (PC)+               ;;LOOP?
    004566 000001                 $EOPCT: .WORD   1
    004570 003013                         BGT     $DOAGN              ;;YES
    004572 012737                         MOV     (PC)+,@(PC)+        ;;RESTORE COUNTER
    004574 000001                 $ENDCT: .WORD   1
    004576 004566                         $EOPCT
    004600 013700 000042          $GET42: MOV     @#42,R0             ;;GET MONITOR ADDRESS
    004604 001405                         BEQ     $DOAGN              ;;BRANCH IF NO MONITOR
    004606 000005                         RESET                       ;;CLEAR THE WORLD
    004610 004710                 $ENDAD: JSR     PC,(R0)             ;;GO TO MONITOR
    004612 000240                         NOP                         ;;SAVE ROOM
↑B  004614 000240                         NOP                         ;;FOR
    004616 000240                         NOP                         ;;ACT11
    004620                        $DOAGN:
    004620 000137                         JMP     @(PC)+              ;;RETURN
    004622 010752                 $RTNAD: .WORD   CYCLE

    004624 000001                 XCSR:   1
    004626    006    002                  .BYTE   6,2
    004630 002042                         DZCSR
    004632 000001                 XVEC:   1
    004634    003    002                  .BYTE   3,2
    004636 002072                         DZRIV
```

```
004640  000001                      XPASS:  1
004642     006     002                      .BYTE   6,2
004644  001242                              $PASS
004646↑B           000001                   XERR:   1
004650     006     002                      .BYTE   6,2
004652  001132                              $ERRTTL

                                            ;SCOPE LOOP AND ITERATION 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

004654                              $SCOPE:
004654  004737  007242              .SCOPE: JSR     PC,SERV.G              ;FIND OUT IF <↑G> WAS HIT
004660  005037  001136                      CLR     $ERRPC                 ;CLEAR LAST ERROR PC.
004664  022716  012220                      CMP     #TST1+2,(SP)           ;IS THIS THE SCOPE AT THE BEGINNING OF TST1?
004670  001413                              BEQ     $XTSTR                 ;IF SO, DON'T LOOP ON IT
004672  000406                      TTST:   BR      1$                     ;GOTO 1$    (IF LOCK SW02=1; THIS LOC =240)
004674  105777  174264                      TSTB    @$TKS                  ;KEYBOARD DONE?
004700  100067                              BPL     $OVER                  ;BR IF NO. (LOCK: HIT KE↑BY TO GOTO NEXT TEST)
004702  017766  174260  177776              MOV     @$TKB,-2(SP)           ;CLEAR DONE BIT
004710  032777  040000  174242      1$:     BIT     #BIT14,@SWR            ;;LOOP ON PRESENT TEST?
004716  001060                              BNE     $OVER                  ;;YES IF SW14=1
                                    ;#####START OF CODE FOR THE XOR TESTER#####
004720  000416                      $XTSTR: BR      6$                     ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                           ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
004722  013746  000004                      MOV     @#ERRVEC,-(SP)         ;;SAVE THE CONTENTS OF THE ERROR VECTOR
004726  012737  004746  000004              MOV     #5$,@#ERRVEC           ;;SET FOR TIMEOUT
004734  005737  177060                      TST     @#177060               ;;TIME OUT ON XOR?
↑B      004740  012637  000004              MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
004744  000436                              BR      $SVLAD                 ;;GO TO THE NEXT TEST
004746  022626                      5$:     CMP     (SP)+,(SP)+            ;;CLEAR THE STACK AFTER A TIME OUT
004750  012637  000004                      MOV     (SP)+,@#ERRVEC         ;;RESTORE THE ERROR VECTOR
004754  000441                              BR      $OVER                  ;;LOOP ON THE PRESENT TEST
004756                              6$:;#####END OF CODE FOR THE XOR TESTER#####
004756  105737  001123              2$:     TSTB    $ERFLG                 ;;HAS AN ERROR OCCURRED?
004762  001404                              BEQ     3$                     ;;BR IF NO
004764  105037  001123              4$:     CLRB    $ERFLG                 ;;ZERO THE ERROR FLAG
004770  005037  001226                      CLR     $TIMES                 ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
004774  032777  004000  174156      3$:     BIT     #BIT11,@SWR            ;;INHIBIT ITERATIONS?
005002  001011                              BNE     1$                     ;;BR IF YES
005004  005737  001242                      TST     $PASS                  ;;IF FIRST PASS OF PROGRAM
005010  001406                              BEQ     1$                     ;;      INHIBIT ITERATIONS
005012  005237  001124                      INC     $ICNT                  ;;INCREMENT ITERATION COUNT
005016  023737  001226  001124              CMP     $TIMES,$ICNT           ;;CHECK THE NUMBER OF ITERATIONS MADE
005024  002015                              BGE     $OVER                  ;;BR IF MORE ITERATION REQUIRED
```

```
        005026  012737  000001  001124  1$:     MOV     #1,$ICNT          ;;REINITIALIZE THE ITERATION COUNTER
005034  013737  005116  001226                  MOV     $MXCNT,$TIMES     ;;SET NUMBER OF ITERATIONS TO DO
005042  105237  001122                  $SVLAD: INCB    $TSTNM            ;;COUNT TEST NUMBERS
005046  113737  001122  001240                  MOVB    $TSTNM,$TESTN     ;;SET TEST NUMBER IN APT MAILBOX
005054  011637  001126                          MOV     (SP),$LPADR       ;;SAVE SCOPE LOOP ADDRESS
005060  013777  001122  174074          $OVER:  MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER
005066  013716  001126                          MOV     $LPADR,(SP)       ;;FUDGE RETURN ADDRESS
005072  105037  001417                  3$:     CLRB    MNTFLG            ;CLEAR THE MAINTENANCE BIT SE↑BTTER AFTER EACH TEST
005076  005737  001370                          TST     MODE              ;HAS THE MODE BEEN CHANGED?
005102  001003                                  BNE     4$                ;IF NOT INTERNAL , GO DO A TEST
005104  112737  000010  001417                  MOVB    #MAINT,MNTFLG     ;IF INTERNAL MODE NOW, SET THE MAINTENANCE BIT
005112  000002                          4$:     RTI                       ;GO DO THE TEST
005114  000406                          BRW:    406
005116  000005                          $MXCNT: 5                         ;;MAX. NUMBER OF ITERATIONS

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

005120  032777  001000  174032          .SCOP1: BIT     #SW09,@SWR        ;IS SW09=1(SET)?
005126  001405                                  BEQ     1$                ;BR IF NOT SET.
005130  005737  001362                          TST     LOCK              ;IS THER A TIGHT LOOP SPECIFIED?
005134  001402                                  BEQ     1$                ;IF NO, RETURN
005136  013716  001362                          MOV     LOCK,(SP)         ;IF YES, GOTO THE ADDRESS IN LOCK.
005142  000002                          1$:     RTI                       ;GO BACK.

005144  032777  010000  174006          .TYPE:  BIT     #SW12,@SWR        ;INHIBIT ALL PRINTOUT??
005152  001403                                  BEQ     1$                ;IF NOT, GO TYPE
005154  062716  000002                          ADD     #2,(SP)           ;SKIP OVER MESSAGE POINTER
005160  000002                                  RTI                       ;RETURN TO WHERE PROCEDURE WAS INVOKED
005162                                  1$:
                                        .SBTTL  TYPE ROUTINE

                                        ;;*********************************↑B************************
                                        ;;*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
                                                ;;*

005162  105737  001177                  $TYPE:  TSTB    $TPFLG            ;;IS THERE A TERMINAL?
005166  100002                                  BPL     1$                ;;BR IF YES
005170  000000                                  HALT                      ;;HALT HERE IF NO TERMINAL
005172  000430                                  BR      3$                ;;LEAVE
005174  010046                          1$:     MOV     R0,-(SP)          ;;SAVE R0
005176  017600  000002                          MOV     @2(SP),R0         ;;GET ADDRESS OF ASCIZ STRING
005202  122737  000001  001254                  CMPB    #APTENV,$ENV      ;;RUNNING IN APT MODE
005210  001011                                  BNE     62$               ;;NO,GO CHECK FOR APT CONSOLE
005212  132737  000100  001255                  BITB    #APTSPOOL,$ENVM   ;;SPOOL MESSAGE TO APT
```

N04

```
005220  001405                            BEQ     62$                  ;;NO,GO CHECK FOR CONSOLE
005222  010037  005232                    MOV     R0,61$               ;;SETUP MESSAGE ADDRESS FOR APT
005226  004737  005452                    JSR     PC,$ATY3             ;;SPOOL MESSAGE TO APT
005232  000000                    61$:    .WORD   0                    ;;MESSAGE ADDRESS
005234  132737  000040  001255    62$:    BITB    #APTCSUP,$ENVM       ;;APT CONSOLE SUPPRESSED
005242  001003                            BNE     60$                  ;;YES,SKIP TYPE OUT
005244  112046                    2$:     MOVB    (R0)+,-(SP)          ;;PUSH CHARACTER TO BE TYPED ONTO STACK
005246  001005                            BNE     4$                   ;;BR IF IT ISN'T THE TERMINATOR
005250  005726                            TST     (SP)+                ;;IF ↑BTERMINATOR POP IT OFF THE STACK
005252  012600                    60$:    MOV     (SP)+,R0             ;;RESTORE R0
005254  062716  000002            3$:     ADD     #2,(SP)              ;;ADJUST RETURN PC
005260  000002                            RTI                          ;;RETURN
005262  122716  000011            4$:     CMPB    #HT,(SP)             ;;BRANCH IF <HT>
005266  001430                            BEQ     8$
005270  122716  000200                    CMPB    #CRLF,(SP)           ;;BRANCH IF NOT <CRLF>
005274  001006                            BNE     5$
005276  005726                            TST     (SP)+                ;;POP  <CR><LF> EQUIV
005300  104402                            TYPE                         ;;TYPE A CR AND LF
005302  001231                            $CRLF
005304  105037  005440                    CLRB    $CHARCNT             ;;CLEAR CHARACTER COUNT
005310  000755                            BR      2$                   ;;GET N↑BEXT CHARACTER
005312  004737  005374            5$:     JSR     PC,$TYPEC            ;;GO TYPE THIS CHARACTER
005316  123726  001176            6$:     CMPB    $FILLC,(SP)+         ;;IS IT TIME FOR FILLER CHARS.?
005322  001350                            BNE     2$                   ;;IF NO GO GET NEXT CHAR.
005324  013746  001174                    MOV     $NULL,-(SP)          ;;GET # OF FILLER CHARS. NEEDED
                                                                       ;;AND THE NULL CHAR.
005330  105366  000001            7$:     DECB    1(SP)                ;;DOES A NULL NEED TO BE TYPED?
005334  002770                            BLT     6$                   ;;BR IF NO--GO POP THE NULL OFF OF STACK
005336  004737  005374                    JSR     PC,$TYPEC            ;;GO TYPE A NULL
005342  105337  005440                    DECB    $CHARCNT             ;;DO NOT COUNT AS A COUNT
005346  000770                            BR      7$                   ;;LOOP

                                  ;HORIZONTAL TAB PROCESSOR

005350  112716  000040            8$:     MOVB    #' ,(SP)             ;;REPLACE TAB WITH SPACE
005354  004737  005374            9$:     JSR     PC,$TYPEC            ;;TYPE A SPACE
005360  132737  000007  005440            BITB    #7,$CHARCNT          ;;BRANCH IF NOT AT
005366  001372                            BNE     9$                   ;;TAB STOP
005370  005726                            TST     (SP)+                ;;POP SPACE OFF STACK
005372  000724                            BR      2$                   ;;GET NEXT CHARACTER
005374  105777  173570            $TYPEC: TSTB    @$TPS                ;;WAIT UNTIL PRINTER IS READY
005400  100375                            BPL     $TYPEC
005402  116677  000002  173562            MOVB    2(SP),@$TPB          ;;LOAD CHAR TO BE TYPE↑BD INTO DATA REG.
005410  122766  000015  000002            CMPB    #CR,2(SP)            ;;IS CHARACTER A CARRIAGE RETURN?
005416  001003                            BNE     1$                   ;;BRANCH IF NO
005420  105037  005440                    CLRB    $CHARCNT             ;;YES--CLEAR CHARACTER COUNT
005424  000406                            BR      $TYPEX               ;;EXIT
005426  122766  000012  000002    1$:     CMPB    #LF,2(SP)            ;;IS CHARACTER A LINE FEED?
005434  001402                            BEQ     $TYPEX               ;;BRANCH IF YES
005436  105227                            INCB    (PC)+                ;;COUNT THE CHARACTER
005440  000000                    $CHARCNT:.WORD  0                    ;;CHARACTER COUNT STORAGE
005442  000207                    $TYPEX: RTS     PC

                                  .SBTTL  APT COMMUNICATIONS ROUTINE

↑B                                ;;************************************************************
```

```
 005444  112737  000001  005710  $ATY1:  MOVB    #1,$FFLG           ;;TO REPORT FATAL ERROR
 005452  112737  000001  005706  $ATY3:  MOVB    #1,$MFLG           ;;TO TYPE A MESSAGE
 005460  000403                          BR      $ATYC
 005462  112737  000001  005710  $ATY4:  MOVB    #1,$FFLG           ;;TO ONLY REPORT FATAL ERROR
 005470                          $ATYC:
 005470  010046                          MOV     R0,-(SP)           ;;PUSH R0 ON STACK
 005472  010146                          MOV     R1,-(SP)           ;;PUSH R1 ON STACK
 005474  105737  005706                  TSTB    $MFLG              ;;SHOULD TYPE A MESSAGE?
 005500  001450                          BEQ     5$                 ;;IF NOT:  BR
 005502  122737  000001  001254          CMPB    #APTENV,$ENV       ;;OPERATING UNDER APT?
 005510  001031                          BNE     3$                 ;;IF NOT:  BR
 005512  132737  000100  001255          BITB    #APTSPOOL,$ENVM    ;;SHOULD SPOOL MESSAGES?
 005520  001425                          BEQ     3$                 ;;IF NOT:  BR
 005522  017600  000004                  MOV     @4(SP),R0          ;;GET MESSAGE ADDR.
 005526  062766  000002  000004          ADD     #2,4(SP)                   ;;BUMP RETURN ADDR.
 005534  005737  001234          1$:     TST     $MSGTYPE           ;;SEE IF DONE W/ LAST XMISSION?
 005540  001375                          BNE     1$                 ;;IF NOT:  WAIT
 005542  010037  001250                  MOV ↑B  R0,$MSGAD          ;;PUT ADDR IN MAILBOX
 005546  105720                  2$:     TSTB    (R0)+              ;;FIND END OF MESSAGE
 005550  001376                          BNE     2$
 005552  163700  001250                  SUB     $MSGAD,R0          ;;SUB START OF MESSAGE
 005556  006200                          ASR     R0                 ;;GET MESSAGE LNGTH IN WORDS
 005560  010037  001252                  MOV     R0,$MSGLGT         ;;PUT LENGTH IN MAILBOX
 005564  012737  000004  001234          MOV     #4,$MSGTYPE        ;;TELL APT TO TAKE MSG.
 005572  000413                          BR      5$
 005574  017637  000004  005620  3$:     MOV     @4(SP),4$          ;;PUT MSG ADDR IN JSR LINKAGE
 005602  062766  000002  000004          ADD     #2,4(SP)                   ;;BUMP RETURN ADDRESS
 005610  013746  177776                  MOV     177776,-(SP) ↑B            ;;PUSH 177776 ON STACK
 005614  004737  005162                  JSR     PC,$TYPE           ;;CALL TYPE MACRO
 005620  000000                  4$:     .WORD   0
 005622                          5$:
 005622  105737  005710          10$:    TSTB    $FFLG              ;;SHOULD REPORT FATAL ERROR?
 005626  001416                          BEQ     12$                ;;IF NOT:  BR
 005630  005737  001254                  TST     $ENV               ;;RUNNING UNDER APT?
 005634  001413                          BEQ     12$                ;;IF NOT:  BR
 005636  005737  001234          11$:    TST     $MSGTYPE           ;;FINISHED LAST MESSAGE?
 005642  001375                          BNE     11$                ;;IF NOT:  WAIT
 005644  017637  000004  001236          MOV     @4(SP),$FATAL      ;;GET ERROR #
 005652  062766  000002  000004          ADD     #2,4(SP)                   ;;BUMP RETURN ADDR.
 005660  005237  001234                  INC     $MSGTYPE           ;;TELL APT TO TAKE ERROR
 005664  105037  005710          12$:    CLRB    $FFLG              ;;CLEAR FATAL FLAG
 005670  105037  005707                  CLRB    $LFLG              ;;CLEAR LOG FLAG
 005674  105037  005706                  CLRB    $MFLG              ;;CLEAR MESSAGE FLAG
 005700  012601                          MOV     (SP)+,R1           ;;POP STACK INTO R1
 005702  012600                          MOV     (SP)+,R0           ;;POP STACK INTO R0
 005704  000207                          RTS     PC                 ;;RETURN
 005706     000                  $MFLG:  .BYTE   0                  ;;MESSG. FLAG
 005707     000                  $LFLG:  .BYTE   0                  ;;LOG FLAG
 005710     000                  $FFLG:  .BYTE   0                  ;;FATAL FLAG
         005712                          .EVEN
         000200                  APTSIZE=200
 ↑B              000001                  APTENV=001
         000100                  APTSPOOL=100
         000040                  APTCSUP=040


                                         ;STRING INPUT ROUTINE
```

```
                                                    ;---------------------------

  005712  010346                           .INSTR:  MOV     R3,-(SP)             ;SAVE R3 ON STACK
  005714  010446                                    MOV     R4,-(SP)             ;SAVE R4 ON STACK
  005716  017637  000004  005734                    MOV     @4(SP),.MSG          ;GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
  005724  062766  000002  000004                    ADD     #2,4(SP)             ;POINT TO INSTRUCTION AFTER ADDRESS POINTER
  005732  104402                           .INST1↑B:        TYPE                 ;PRINT THE MESSAGE
  005734  000000                           .MSG:    0                            ;MESSAGE IS POINTED TO FROM HERE
  005736  012704  010502                            MOV     #INBUF,R4            ;POINT R4 TO THE INPUT BUFFER
  005742  012703  000007                            MOV     #7,R3                ;SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
  005746  105777  173212                   1$:      TSTB    @$TKS                ;HAS A CHARACTER BEEN RECEIVED?
  005752  100375                                    BPL     1$                   ;IF NO, KEEP WAITING FOR IT
  005754  117714  173206                            MOVB    @$TKB,(R4)           ;IF YES, SAVE IT IN THE INPUT BUFFER
  005760  142714  000200                            BICB    #200,(R4)            ;KEEP ONLY THE 7-BIT ASCII INFORMATION
  005764  122427  000015                            CMPB    (R4)+,#15            ;IS THIS CHARACTER A LINE FEED?
  005770  001417                                    BEQ     INSTR2               ;IF SO, TERMINATE THE INPUT SEQUENCE
  005772  105777  173172                   2$:      TSTB    @$TPS                ;IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
  005776  100375                                    BPL     2$                   ;IF WE CAN'T, WAIT UNTIL WE CAN
  006000  017777  173162  173164                    MOV     @$TKB,@$TPB          ;ECHO THE CHARACTER BACK
  006006  005303                                    DEC     R3                   ;REDUCE THE NUMBER OF CHARACTERS RECEIVED
  006010  001356                                    BNE     1$                   ;IF WE DON'T HAVE 7, GO GET SOME MORE
  006012  012604                                    MOV     (SP)+,R4             ;IF WE HAVE 7, RESTORE R4
  006014  012603                                    MOV     (SP)+,R3             ;RESTORE R3
  006016  010346                           .INSTE:  MOV     R3,-(SP)             ;SAVE R3 ON THE STA↑BCK
  006020  010446                                    MOV     R4,-(SP)             ;SAVE R4 ON THE STACK
  006022  104402  001230                            TYPE    ,$QUES               ;PRINT A QUESTION MARK... WHAT'S GOING ON?
  006026  000741                                    BR      .INST1               ;GO PRINT THE MESSAGE AGAIN
  006030  012604                           INSTR2:  MOV     (SP)+,R4             ;RESTORE R4
  006032  012603                                    MOV     (SP)+,R3             ;RESTORE R3
  006034  000002                                    RTI                          ;RETURN TO THE MAIN PROCEDURE

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

  006036  010546                           .PARAM:  MOV     R5,-(SP)             ;SAVE R5 ON THE STACK
  006040  010446                                    MOV     R4,-(SP)             ;SAVE R4 ON THE STACK
  006042  016605  00000↑B4                          MOV        4(SP),R5              ;GET THE SETUP INFORMATION POINTER
  006046  012537  006226                            MOV     (R5)+,LOLIM          ;SET THE LOW LIMIT FOR THE INPUT
  006052  012537  006230                            MOV     (R5)+,HILIM          ;SET THE HIGH LIMIT FOR THE INPUT
  006056  012537  006232                            MOV     (R5)+,DEVADR         ;SAVE THE ADDRESS WHERE THE RESULT WILL BE STORED
  006062  112537  006234                            MOVB    (R5)+,LOBITS         ;GET THE MASK OF THE INCORRECT BITS
  006066  112537  006235                            MOVB    (R5)+,ADRCNT         ;GET THE COUNT OF ITEMS TO BE STORED
  006072  010566  000004                            MOV     R5,4(SP)             ;POINT TO WHERE MAIN LINE PROGRAM WILL RESUME
  006076  005005                           PARAM1:  CLR     R5                   ;INITIALIZE THE ASCII TO OCTAL RESULT WORD
  006100  012704  010502                            MOV     #INBUF,R4            ;POINT TO THE INPUT BUFFER
  006104  122714  000015                            CMPB    #15,(R4)             ;IS THIS CHARACTER A CARRIAGE RETURN?
  006110  001420                                    BEQ     PARERR               ;IF SO, PRINT THE MESSAGE AGAIN
  006112  121427  000060                   1$:      CMPB    (R4),#60             ;IS THIS CHARACTER BELOW THE NUMERIC RANGE?
  006116  002415                                    BLT     PARERR               ;IF SO, GO PRINT THE MESSAGE AGAIN
  006120  121427  000067                            CMPB    (R4),#67             ;IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
  006124  003012                                    BGT     PARERR               ;IF SO, GO PRINT THE MESSAGE AGAIN
  006126  142714  000060                            BICB    #60,(R4)             ;ISOLATE THE NUMBER THE CHARACT↑BER REPRESENTS
  006132  152405                                    BISB    (R4)+,R5             ;CONCATENATE THESE BITS TO THE ALREADY EXISTING STRING
  006134  122714  000015                            CMPB    #15,(R4)             ;IS THE NEXT CHARACTER A CARRIAGE RETURN?
  006140  001406                                    BEQ     LIMITS               ;IF SO, GO SEE IF NUMBER IS WITHIN LIMITS
  006142  006305                                    ASL     R5                   ;CLEAR BIT POSITION 0, MOVE EXISTING STRING TO LEFT
```

```
      006144  006305                   ASL     R5              ;CLEAR POSITION 1, MOVE STRING TO LEFT AGAIN
      006146  006305                   ASL     R5              ;MOVE THE STRING ONE MORE TIME TO MAKE ROOM FOR
↑B                                                                       ;NEXT THREE BITS
      006150  000760                   BR      1$              ;GO GET THE NEXT CHARACTER
      006152  104404          PARERR:  INSTER                  ;THERE WAS AN ERROR... GO PRINT MESSAGE AGAIN
      006154  000750                   BR      PARAM1          ;TRY GETTING THE PARAMETERS AGAIN

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

      006156  020537  006230  LIMITS:  CMP     R5,HILIM        ;DOES RESULT EXCEED ITS MAXIMUM CORRECT VALUE?
      006162  101373                   BHI     PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
      006164  020537  006226           CMP     R5,LOLIM        ;IS THE RESULT LOWER THAN ALLOWED?
      006170  103770                   BLO     PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
      006172  133705  006234           BITB    LOBITS,R5       ;ARE ANY INCORRECT BITS SET IN THE RESULT?
      006176  001365                   BNE     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN

                                       ;STORE NUMBER AT SPECIFIED ADDRESS

      006200  013704  006232           MOV     DEVADR,R4       ;POINT TO THE LOCATION WHERE THE RESULT WILL BE STORED
      006204  010524          1$:      MOV     R5,(R4)+        ;STORE THE RESULT
      006206  062705  000002           ADD     #2,R5           ;CALCULATE THE NEXT DATUM
      006212  105337  006235           DECB    ADRCNT          ;REDUCE COUNT OF STORED RESULTS. IS IT EXCEEDED?
      006216  001372                   BNE     1$              ;IF NOT, ↑BGO STORE THE NEXT DATUM
      006220  012604                   MOV     (SP)+,R4        ;RESTORE R4
      006222  012605                   MOV     (SP)+,R5        ;RESTORE R5
      006224  000002                   RTI                     ;RETURN TO THE MAIN PROGRAM

      006226  000000          LOLIM:   0                       ;LOWEST ACCEPTABLE VALUE
      006230  000000          HILIM:   0                       ;HIGHEST ACCEPTABLE
      006232  000000          DEVADR:  0                       ;LOCATION WHERE RESULT WILL BE STORED
      006234     000          LOBITS:  .BYTE   0               ;INCORRECT BITS MASK
      006235     000          ADRCNT:  .BYTE   0               ;COUNT OF ITEMS TO BE STORED

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

      ↑B006236          016637   000004   001402   .SAV05: MOV     4(SP),SAVPC     ;SAVE R7 (PC)

                                       ;SAVE R0-R5

      006244  010537  001214  SV05:    MOV     R5,$REG5        ;SAVE R5
      006250  010437  001212           MOV     R4,$REG4        ;SAVE R4
      006254  010337  001210           MOV     R3,$REG3        ;SAVE R3
      006260  010237  001206           MOV     R2,$REG2        ;SAVE R2
      006264  010137  001204           MOV     R1,$REG1        ;SAVE R1
      006270  010037  001202           MOV     R0,$REG0        ;SAVE R0
      006274  000002                   RTI                     ;LEAVE.

                                       ;RESTORE R0-R5

      006276  013700  001202  .RES05:  MOV     $REG0,R0        ;RESTORE R0
      006302  013701  001204           MOV     $REG1,R1        ;RESTORE R1
      006306  013702  001206           MOV     $REG2,R2        ;RESTORE R2
      006312  013703  001210           MOV     $REG3,R3        ;RESTORE R3
      006316  013704  001212           MOV     $REG4,R4        ;RESTORE R4
```

```
006322  013705  001214                  MOV     $REGS,R5        ;RESTORE R5
006326  000002                          RTI                     ;LEAVE

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

006330  104402  001231          .CONVR: TYPE    ,$CRLF          ;PRINT A CARRIAGE RETURN
006334  010046                  .CNVRT: MO↑BV   R0,-(SP)        ;SAVE R0
006336  010146                          MOV     R1,-(SP)        ;SAVE R1
006340  010346                          MOV     R3,-(SP)        ;SAVE R3
006342  010446                          MOV     R4,-(SP)        ;SAVE R4
006344  010546                          MOV     R5,-(SP)        ;SAVE R5
006346  017601  000012                  MOV     @12(SP),R1      ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
006352  062766  000002  000012          ADD     #2,12(SP)       ;POINT TO WHERE MAIN PROGRAM WILL RESUME
006360  012137  006504                  MOV     (R1)+,WRDCNT    ;GET NUMBER OF WORDS TO BE PRINTED
006364  112105                  1$:     MOVB    (R1)+,R5        ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
006366  112100                          MOVB    (R1)+,R0        ;GET THE NUMBER OF SPACES TO PRINT
0↑B06370        013104                  MOV     @(R1)+,R4       ;COPY THE WORD TO BE CONVERTED
006372  110537  006506                  MOVB    R5,CHRCNT       ;COPY THE CHARACTER COUNT
006376  010403                  3$:     MOV     R4,R3           ;COPY THE ARGUMENT WORD AGAIN
006400  042703  177770                  BIC     #↑C<7>,R3       ;ISOLATE THREE BITS TO BE TREATED AS A CHARACTER
006404  062703  000060                  ADD     #060,R3         ;MAKE AN ASCII CHARACTER OUT OF THEM
006410  110346                          MOVB    R3,-(SP)        ;SAVE THAT CHARACTER
006412  006004                          ROR     R4              ;MOVE THE NEXT THREE BITS INTO PLACE
006414  006204                          ASR     R4              ;MOVE THEM AGAIN
006416  006204                          ASR     R4              ;AND FINALLY A THIRD TIME
006420  005305                          DEC     R5              ;REDUCE CHARACTER COUNT.ARE ALL CHARACTERS
                                                                ;BUILT?
006422  001365                          BNE     3$              ;IF NO, GO BUILD THE NEXT ONE.
006424  012703  010606                  MOV     #MDATA,R3       ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
006430  112623                  4$:     MOVB    (SP)+,(R3)+     ;STORE THE CHARACTER, STARTING WITH THE MOST
006432  105337  006506                  DECB    CHRCNT          ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
006436  001374                          BNE     4$              ;IF NO, GO TRANSFER ANOTHER
006440  105700                          TSTB    R0              ;ARE ANY SPACES TO BE PRINTED?
006442  001404                          BEQ     6$              ;IF NO, DON'T SET UP ANY
006444  112723  000040          5$:     MOVB    #0↑B40,(R3)+    ;ADD A SPACE TO THE OUTPUT BUFFER
006450  105300                          DECB    R0              ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
006452  001374                          BNE     5$              ;IF YES, GO ADD ANOTHER SPACE
006454  105013                  6$:     CLRB    (R3)            ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
006456  104402  010606                  TYPE    ,MDATA          ;PRINT THE STRING WE JUST BUILT
006462  005337  006504                  DEC     WRDCNT          ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
006466  001336                          BNE     1$              ;IF YES, GO CONVERT THEM
006470  012605                          MOV     (SP)+,R5        ;RESTORE R5
006472  012604                          MOV     (SP)+,R4        ;RESTORE R4
006474  012603                          MOV     (SP)+,R3        ;RESTORE R3
↑B1517          006476  012601          MOV     (SP)+,R1        ;RESTORE R1
006500  012600                          MOV     (SP)+,R0        ;RESTORE R0
006502  000002                          RTI                     ;RETURN TO THE MAIN PROGRAM
006504  000000                  WRDCNT: 0
006506     000                  CHRCNT: .BYTE                   ;NUMBER OF CHARACTERS TO PRINT
006507     000                  SPACNT: .BYTE   0               ;NUMBER OF SPACES TO PRINT

006510  000000                  BINWRD: 0


                                ;TRAP DISPATCH SERVICE
```

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

      006512  010046                   .TRPSR: MOV     R0,-(SP)          ;SAVE R0. USE R0 TO FIND TRAP ROUTINE
      006514  016600  000002                   MOV     2(SP),R0          ;GET TRAP ADDRESS
      006520  005740                           TST     -(R0)             ;GET TRAP
      006522  111000                           MOVB    (R0),R0           ;GET RIGHT BYTE OF TRAP(TRAP OFFSET)
      006524  006300                           ASL     R0                ;POSITION OFFSET FOR TABLE INDEXING
      006526  016000  002002                   MOV     .TRPTAB(R0),R0    ;PLACE INDEXED ADDRESS OF TABLE IN R0
      006532  000200                           RTS     R0                ;TRANSFER TO THAT ADDRESS AND RESTORE OLD R0

                                                ;DEVICE CLEAR ROUTINE
                                                ;ISSUE A DEVICE CLEAR
                                                ;---↑B-----------------
      006534                           .DEVICE.CLR:
      006534  052777  000020  173300           BIS     #DCLR,@DZCSR      ;SET DCLR
      006542  032777  000020  173272   1$:     BIT     #DCLR,@DZCSR      ;DID IT CLEAR?
      006550  001374                           BNE     1$                ;BR IF NO
      006552  000002                           RTI                       ;EXIT ROUTINE

                                                ;ROUTINE TO HANDLE MAINTENANCE BIT SETTING WITH DEVICE CLEAR
                                                ;-------------------------------------------------------------
      006554  104413                   .DCLASM:DEVICE.CLR                ;ISSUE A DEVICE CLEAR
      006556  153777  001417  173256           BISB    MNTFLG,@DZCSR     ;LOAD THE MAINTENANCE BIT IF IT IS I MODE
      006564  000002                           RTI                       ;RETURN TO CALLI↑BNG ROUTINE

      006566                           .DELAY:
      006566  010046                           MOV     R0,-(SP)          ;SAVE R0
      006570  013700  006604                   MOV     DLYCNT,R0         ;SET COUNT
      006574  005300                   1$:     DEC     R0                ;DELAY
      006576  001376                           BNE     1$                ;
      006600  012600                           MOV     (SP)+,R0          ;RESTORE R0
      006602  000002                           RTI                       ;LEAVE ROUTINE
      006604  000001                   DLYCNT: .WORD      1              ;PATCHABLE LOC FOR MORE TIME

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

      006606  013716  001360           .ADVANCE:MOV    NEXT,(SP)         ;CRUNCH STACK WITH ADDRESS OF SCOPE CALL
      006612  005037  001362                   CLR     LOCK              ;RESET TIGHT LOOP ADDRESS
      006616  000002                           RTI                       ;CHECK TO SEE IF OLD TEST GETS REPEATED

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

      006620  004737  007242           $ERROR: JSR     PC,SERV.G         ;FIND OUT IF <↑G> WAS HIT
      006624  032777  010000  172326           BIT     #SW12,@SWR        ;BELL ON ERROR?
      006632  001406                           BEQ     XBX               ;BR IF NO BELL
      006634  105777  172330                   TSTB    @$TPS             ;TTY READY.
      006640  100003                           BPL     XBX               ;DON'T WAIT IF TTY NOT READY.
      006642  112777  000207  172322           MOVB    #207,@$TPB        ;PUSH A BELL AT THE TTY.
      006650  032777  020000  172302   XBX:    BIT     #SW13,@SWR        ;DELETE ERROR PRINT OUT?
  ↑B          006656  001113                   BNE        HALTS              ;BR IF NO PRINT OUT WANTED.
      006660  021637  001136                   CMP     (SP),$ERRPC       ;WAS THIS ERROR FOUND LAST TIME?
      006664  001404                           BEQ     1$                ;BR IF YES
```

```
006666  011637  001136                         MOV     (SP),$ERRPC      ;RECORD BEING HERE
006672  105037  001123                         CLRB    $ERFLG           ;PREPARE HEADER
006676  104407                         1$:     SAVOS                    ;SAVE ALL PROC REGISTERS
006700  011605                                 MOV     (SP),R5          ;GET THE PC OF ERROR
006702  162705  000002                         SUB     #2,R5            ;GET ADDRESS OF TRAP CALL
006706  011504                                 ↑BMOV   (R5),R4          ;GET ERROR INSTRUCTION
006710  110437  001134                         MOVB    R4,$ITEMB        ;COPY TEST NUMBER FOR APT HANDLING
006714  006304                                 ASL     R4               ;MULT BY TWO
006716  061504                                 ADD     (R5),R4          ;DOUBLE IT
006720  006304                                 ASL     R4               ;MULT AGAIN
006722  042704  177001                         BIC     #177001,R4       ;CLEAR JUNK
006726  062704  026222                         ADD     #.ERRTAB,R4      ;GET POINTER
006732  012437  007056                         MOV     (R4)+,ERRMSG     ;GET ERROR MESSAGE
006736  012437  007070                         MOV     (R4)+,DATAHD     ;GET DATA HEADRER
006742  011437  007102                         MOV     (R4),DATABP      ;GET DATA TABLE
006746  105737  001123                         TSTB    $ERFLG           ;TYPE HEADER
006752  001403                                 BEQ     TYPMSG           ;BR IF YES
006754  005737  007102                         TST     DATABP           ;DOES DATA TABLE EXIST?
006760  001044                                 BNE     TYPDAT           ;BR IF YES.
006762  104402  001231                 TYPMSG: TYPE    ,$CRLF           ;TYPE A CARRIAGE RETURN
006766  104402  001231                         TYPE    ,$CRLF           ;AND TYPE ANOTHER
006772  005737  001362                         TST     LOCK
006776  001402                                 BEQ     1$
007000  104402  010177                         TYPE    ,MASTEK
007004  104402  010165                 1$:     TYPE    ,MTSTN
007010  104412  007234                         CNVRT   ,XTSTN           ;SHOW IT
007014  104402  010254                         TYPE    ,MERRPC          ;TYPE PC.
007020  104412  007226                         CNVRT   ,ERTABO          ;SHOW IT
007024  104402  010↑B127                               TYPE    ,MCSRX
007030  104412  004624                         CNVRT   ,XCSR
007034  104402  001231                         TYPE    ,$CRLF           ;GIVE A CR/LF
007040  112737  177777  001123                 MOVB    #-1,$ERFLG       ;NO MORE HEADER UNLESS NO DATA TABLE.
007046  005737  007056                         TST     ERRMSG           ;IS THERE AN ERROR MESSAGE?
007052  001402                                 BEQ     WTBS.FM          ;BR IF NO.
007054  104402                                 TYPE                     ;TYPE
007056  000000                         ERRMSG: 0                        ;    ERROR MESSAGE
007060                                 WTBS.FM:                         ;
007060  005737  007070                         TST     DATAHD           ;DATA HEADER?
007064  001402                                 BEQ     TYPDAT           ;BR IF NO
007066  104402                                 TYPE                     ;TYPE
007070  000000                         DATAHD: 0                        ;    DAT↑BA HEADER
007072  005737  007102                 TYPDAT: TST     DATABP           ;DATA TABLE?
007076  001402                                 BEQ     RESREG           ;BR IF NO.
007100  104411                                 CONVRT                   ;SHOW
007102  000000                         DATABP: 0                        ;    DATA TABLE
007104  104410                         RESREG: RESOS                    ;RESTORE PROC REGISTERS
007106  122737  000001  001254         HALTS:  CMPB    #APTENV,$ENV     ;IS APT RUNNING?
007114  001007                                 BNE     2$               ;SKIP APT CALL IF NOT
007116  113737  001134  007130                 MOVB    $ITEMB,7$        ;COPY ERROR NUMBER
007124  004737  005462                         JSR     PC,$ATY4         ;CALL APT SERVICE
007130  000000                         7$:     .WORD   0                ;ERROR NUMBER STUCK HERE
007132  000777                         8$:     BR      8$               ;LOCK UP HERE
007134  022737  004610  000042         2$:     CMP     #$ENDAD,@#42     ;CHECK TO SEE IF IN ACT-11 MODE
007142  001403                                 BEQ     1$               ;IF SO, HANDLE ACCORDINGLY
007144  005777  172010                         TST     @SWR             ;HALT ON ERROR?
007150  100004                                 BPL     EXITER           ;BR IF NO HALT ON ERROR
```

```
007152  016677  000002  172002  1$:      MOV     2(SP),@DISPLAY    ;SHOW ERROR PC IN DATA DISPLAY
007160  000000                           HALT                      ;HALT
007162  005237  001132          EXITER:  INC     $ERTTL            ;UPDATE ERROR COUNT
007166  032777  000400  171764           BIT     #↑BSW08,@SWR      ;GOTO TOP OF TEST?
007174  001007                           BNE     1$                ;BR IF YES
007176  032777  002000  171754           BIT     #SW10,@SWR        ;GOTO NEXT TEST?
007204  001407                           BEQ     2$                ;BR IF NO
007206  013737  001360  001126           MOV     NEXT,$LPADR       ;SET FOR NEXT TEST
007214  012706  001120          1$:      MOV     #STACK,SP         ;RESET SP
007220  000177  171702                   JMP     @$LPADR           ;GOTO SPECIFIED TEST
007224  000002                  2$:      RTI                       ;RETURN
007226  000001                  ERTABO:  1
007230     006     002                   .BYTE   6,2
007232  001402                           SAVPC
007234  000001                  XTSTN:   1
007236     002     002                   .BYTE   2,2
007240  001122                           $TSTNM
B7      007242  022737  177570  001160   SERV.G: CMP     #177570,SWR     ;IS THE SWITCH REGISTER HARDWIRED?
007250  001513                           BEQ     6$                ;IF SO, IGNORE ↑G
007252  017746  171710                   MOV     @$TKB,-(SP)       ;OTHERWISE, GET THE LAST CHARACTER TYPED
007256  042716  000200                   BIC     #BIT7,(SP)        ;STRIP PARITY(EIGHTH) BIT
007262  122726  000007                   CMPB    #7,(SP)+          ;IS IT ↑G?
007266  001104                           BNE     6$                ;IF NOT, IGNORE INPUT
007270  032777  004000  171666           BIT     #4000,@$TKS       ;RX BUSY?
007276  001361                           BNE     SERV.G            ;BR IF YES
007300  017737  171654  007522           MOV     @SWR,90$          ;SAVE (SWR).
007306  013777  007522  171644  1$:      MOV     90$,@SWR          ;
007314  104402  007502                   TYPE    ,89$              ;
007320  104412  007514                   CNVRT   ,88$              ;
007324  104402  007524                   TYPE    ,91$              ;
007330  105777  171630                   TSTB    @$TKS             ;WAIT FOR DONE.
007334  100375                           BPL     .-4
007336  017746  171624                   MOV     @$TKB,-(SP)       ;
007342  042716  000200                   BIC     #BIT7,(SP)        ;
007346  122726  000015                   CMPB    #15,(SP)+         ;
007352  001450                           BEQ     5$                ;
007354  005077  171600                   CLR     @SWR              ;
007360  105777  171604          2$:      TSTB    @$TPS             ;
007364  100375                           BPL     .-4               ;
007366  016677  177776  171576           MOV     -2(SP),@$TPB      ;
007374  000241                           CLC                       ;
007376  006177  ↑B171556                 ROL       @SWR                    ;
007402  006177  171552                   ROL     @SWR              ;
007406  006177  171546                   ROL     @SWR              ;
007412  103735                           BCS     1$                ;ERROR
007414  026627  177776  000060           CMP     -2(SP),#60        ;
007422  002731                           BLT     1$                ;
007424  026627  177776  000067           CMP     -2(SP),#67        ;
007432  003325                           BGT     1$                ;
007434  042766  177770  177776           BIC     #↑C<7>,-2(SP)     ;
007442  056677  177776  171510           BIS     -2(SP),@SWR       ;
007450  105777  171510                   TSTB    @$TKS             ;
007454  100375                           BPL     .-4               ;
007456  017746  171504                   MOV     @$TKB,-(SP)
007462  042716  000200                   BIC     #BIT7,(SP)
007466  122726  000015  ↑B               CMPB    #15,(SP)+
007472  001332                          BNE     2$                    ;
007474  104402  001231          5$:     TYPE    ,$CRLF                ;
007500  000207                  6$:     RTS     PC

007502  020200  051450  051127  89$:    .ASCIZ  <200>? (SWR)=/?
007510  036451  000057
                                .EVEN
007514  000001                  88$:    1
007516     006     000                  .BYTE   6,0
007520  007522                          90$
007522  000000                  90$:    .WORD   0
007524  036457  000057          91$:    .ASCIZ  ?/=/?
                                .EVEN
                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;*****************************************************************
                                ;POWER DOWN ROUTINE
007530  012737  007674  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
007536  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
007544  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
007546  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
007550  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
007552  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
007554  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
007556  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
007560  017746  171374                  MOV↑B   @SWR,-(SP)      ;;PUSH @SWR ON STACK
007564  010637  007700                  MOV     SP,$SAVR6       ;;SAVE SP
007570  012737  007602  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
007576  000000                          HALT
007600  000776                          BR      .-2             ;;HANG UP

                                ;;*****************************************************************
                                ;POWER UP ROUTINE
007602  012737  007674  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
007610  013706  007700                  MOV     $SAVR6,SP       ;;GET SP
007614  005037  007700                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
007620  005237  007700          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
007624  ↑B001375                                BNE     1$        ;;OF  WORD
007626  012677  171326                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
007632  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
007634  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
007636  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
007640  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
007642  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
007644  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
007646  012737  007530  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
007654  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
007662  104402                          TYPE                    ;;REPORT THE POWER FAILURE
007664  007702                  $PWRMG: .WORD   MPFAIL          ;;POWER FAIL MESSAGE POINTER
007666  012716                          MOV     (PC)+,(SP)      ;;RESTART AT RESTART
007670  011304                  $PWRAD: .WORD   RESTART         ;;RESTART ADDRESS
007672  000002                          RTI
007674  000000                  $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
007676  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
007700  000000                  $SAVR6: 0                       ;;PUT THE SP HERE
007702  050200  051127  043040  MPFAIL: .ASCIZ  <200>/PWR FAILED. RESTART AT LAST TEST /
```

```
 (2)  007745      200  047105  020104  MEPASS: .ASCIZ  <200>/END PASS DZDZA-C  /
 (2)  007771      200  052522  047116  MR:     .ASCIZ  <200>/RUNNING    /
 (2)  010005      200  051120  043517  MERR2:  .ASCIZ  <200>/PROGRAM INDICATES NO DEVICES PRESENT./
 (2)  010054   044600  051516  043125  MERR3:  .ASCIZ  <200>/INSUFFICIENT DATA!/
 (2)  010100   046200  041517  020113  MLOCK:  .ASCIZ  <200>/LOCK ON SELECTED TEST/
 (2)  010127      103  051123  020072  MCSRX:  .ASCIZ   /CSR: /
 (2)  010135      126  041505  020072  MVECX:  .ASCIZ  /VEC: /
 (2)  010143      120  051501  042523  MPASSX: .ASCIZ  /PASSES: /
 (2)  010154   051105  047522  05152↑B2        MERRX:  .ASCIZ  /ERRORS: /
 (2)  010165      124  051505  020124  MTSTN:  .ASCIZ  /TEST NO: /
 (2)  010177      052  000040          MASTEK: .ASCIZ  /* /
 (2)  010202   051600  052105  051440  MNEW:   .ASCIZ  <200>/SET SWITCH REG TO DZ11'S DESIRED ACTIVE./
 (2)  010254   041520  020072     000  MERRPC: .ASCIZ  /PC: /
 (2)  010261      200  040515  020120  XHEAD:  .ASCIZ  <200>/MAP OF DZ11 STATUS/<200>
 (2)  010306   044600  046114  043505  MBADLN: .ASCIZ  <200>/ILLEGAL ENTRY IN STAGGERED MODE/<200>
 (2)                                           .EVEN
 (2)  010350   000002                  XSTATQ: 2
 1752 010352      006     003                  .BYTE   6,3
 1753 010354   001220                          $TMP1
 1754 010356      006     002                  .BYTE   6,2
 1755 010360   001222                          $TMP2
 1756                                  .EVEN
 1757                                          ;THIS ROUTINE ESTABLISHES WHICH MAINTENANCE MODE THE DEVICE IS IN
 1758                                          ;-----------------------------------------------------------------
 1759                                          ;E=EXTERNAL LOOP BACK
 1760                                          ;I=INTERNAL LOOP BACK
 1761                                          ;S=STAGGERED LOOP BACK
 1762 010362   017605  000000          .SETFLG:MOV     @(SP),R5                ;PICK UP ADDRESS OF TAG
 1763 010366   042737  000040  010502          BIC     #40,INBUF               ;STRIP LOWER CASE
 1764 010374   122737  000105  010502          CMPB    #'E,INBUF               ;IS IT EXTERNAL LOOP BACK ?
 1765 010402   001005                          BNE     4$                      ;NO
 1766 010404   013715  010474                  MOV     1$,(R5)                 ;YES STORE INFO
 1767 010410   1050↑B37        001417                  CLRB    MNTFLG                  ;SET MAINT BIT =0
 1768 010414   000422                          BR      7$                      ;GET OUT
 1769 010416   122737  000111  010502  4$:     CMPB    #'I,INBUF               ;IS IT INTERNAL LOOP BACK ?
 1770 010424   001006                          BNE     5$                      ;NO
 1771 010426   013715  010476                  MOV     2$,(R5)                 ;YES STORE INFO
 1772 010432   112737  000010  001417          MOVB    #MAINT,MNTFLG           ;SET UP THE MAINTENANCE FLAG LOADER
 1773 010440   000410                          BR      7$                      ;GET OUT
 1774 010442   122737  000123  010502  5$:     CMPB    #'S,INBUF               ;IS IT STAGGERED LOOP BACK ?
 1775 010450   001007                          BNE     6$                      ;WHAT ?
 1776 010452   013715  010500                  MOV     3$,(R5)                 ;YES STORE INFO
 1777 010456   105037  001417                  CLRB    MNTFLG                  ;ZERO BITS
↑B1778         010462  062716  000002          7$:     ADD     #2,(SP)                 ;POP AROUND
 1779 010466   000002                          RTI
 1780 010470   104404                  6$:     INSTER                          ;RETRY
 1781 010472   000733                          BR      .SETFLG                 ;DITTO
 1782 010474   000200                  1$:     .WORD   200                     ;EXTERNAL =  E
 1783 010476   000000                  2$:     .WORD   0                       ;INTERNAL = I
 1784 010500   100000                  3$:     .WORD   100000                  ;STAGGERED = S
 1785
 1786                                          ;BUFFERS FOR INPUT-OUTPUT
 1787
 1788 010502   000000                  INBUF:  0
 1789          010544                  .=.+40
 1790 010544   000000                  TEMP:   0
```

```
        010606                          .=.+40
010606  000000                          MDATA:  0
        010650                          .=.+40

010650  011637  010746                  SET.PS: MOV     (SP),3$
010654  162737  000002  010746                  SUB     #2,3$
010662  017737  000060  010750                  MOV     @3$,4$
010670  022737  106427  010750                  CMP     #106427,4$
010676  001003                                  BNE     1$
010700  011637  010746                          MOV     (SP),3$
010704  000412                                  BR      2$
010706  022737  106437  010750  1$:             CMP     #106437,4$
010714  001401                                  BEQ     .+4
010716  000000                                  HALT            ;RESERVED INSTRUCTION NOT "MTPS"
010720  011637  010746                          MOV     (SP),3$
010724  017737  000016  010746                  MOV     @3$,3$
010732  062716  000002          2$:             ADD     #2,(S↑BP)
010736  017766  000004  000002                  MOV     @3$,2(SP)
010744  000002                                  RTI
010746  000000                  3$:             0
010750  000000                  4$:             0
```

```


                                              ;
                                              ;ROUTINE USED TO "CYCLE" THROUGH UP TO SIXTEEN DZ11'S
                                              ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
                                              ;AND RUNS THE SPECIFIED DZ11'S.   THIS ROUTINE *MUST*
                                              ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
                                              ;SETUP NECESSARY.
                                              ;

 010752  005737  0014↑B04                     CYCLE:  TST     DZACTV          ;ARE ANY DZ11'S TO BE TESTED?
 010756  001004                               BNE     1$                      ;BR IF OK.
 010760  104402  010005                       TYPE    ,MERR2                  ;NO DZ11'S SELECTED!!
 010764  000000                               HALT                            ;STOP THE SHOW.
 010766  000776                               BR      .-2                     ;DISQUALIFY CONT. SW.
 010770  013737  005116  001226     1$:       MOV     $MXCNT,$TIMES           ;RESTORE THE NUMBER OF ITERATIONS TO MAKE
 010776  033737  001406  001404               BIT     RUN,DZACTV              ;IS THIS ONE "ACTIVE"
 011004  001017                               BNE     2$                      ;BR IF GOOD ONE FOUND.
 011006  006137  001406                       ROL     RUN                     ;UPDATE POINTER
 011012  005537  001406                       ADC     RUN                     ;CATCH CARRY FROM RUN
 011016  062737  000014  001412               ADD     #14,ACTIVE              ;UPDATE ADDRESS POINTER.
 011024  022737  002000  001412               CMP     #DZ.END,ACTIVE          ;HAVE WE PASSED THE END OF THE MAP?
 011032  001356                               BNE     1$                      ;IF NO, KEEP GOING; NOT ALL TESTED FOR.
 011034  012737  001500  001412               MOV     #DZ.MAP,ACTIVE          ;RESET ADDRESS POINTER.
 011042  000752                               BR      1$                      ;KEEP LOOKING FOR ACTIVE DZ11
 011044  006137  001406             2$:       ROL     RUN                     ;UPDATE POINTER.
 011050  005537  001406                       ADC     RUN                     ;CATCH CARRY.
 011054  013700  001412                       MOV     ACTIVE,R0               ;GET ADDRESS POINTER.
 011060  062737  000014  001412               ADD     #14,ACTIVE              ;UPDATE.
 011066  022737  002000  001412               CMP↑B   #DZ.END,ACTIVE
                                                                              ;ALL DONE?
 011074  001003                               BNE     3$                      ;BR IF NO.
 011076  012737  001500  001412               MOV     #DZ.MAP,ACTIVE          ;RESTORE POINTER.
 011104  012037  001310             3$:       MOV     (R0)+,$BASE             ;LOAD SYSTEM CTRL. REG
 011110  012037  002072                       MOV     (R0)+,DZRIV             ;LOAD VECTOR
 011114  012037  026216                       MOV     (R0)+,DZPRT             ;LOAD PRIORITY
 011120  113737  026217  001414               MOVB    DZPRT+1,EIAFLG          ;EIA OR 20MA
 011126  042737  100000  026216               BIC     #BIT15,DZPRT            ;CLEAR FLAG
 011134  012037  001364                       MOV     (R0)+,LINE              ;SET UP LINE DZ LINES ACTIVE
 011140  012037  001366                       MOV     (R0)+,PAR               ;SET UP PARAMETERIZATION
↑B       011144  012037  001370                       MOV     (R0)+,MODE      ;SET UP MAINTENANCE MODE
 011150  004737  026010                       JSR     PC,DZLEV                ;SET UP
 011154  005737  000042                       TST     @#42                    ;ARE WE UNDER MONITOR CONTROL?
 011160  001046                               BNE     4$                      ;IF YES, SKIP THIS SETUP
 011162  032777  000002  167770               BIT     #SW01,@SWR              ;IF SW01=1, GET STARTING TEST #
 011170  001442                               BEQ     4$                      ;BR IF NO TEST IS TO BE INPUTTED
 011172  104402  001231             7$:       TYPE    ,$CRLF
 011176  104403                               INSTR                           ;CALL THE STRING INPUT ROUTINE
 011200  010165                               MTSTN                           ;POINTER TO MESSAGE TO BE PRINTED
 011202  104405                               PARAM                           ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
 011204  000001                               1                               ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
 011206  001000                               1000                            ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
 011210  001122                               $TSTNM                          ;POINTER TO MAP LOCATION TO BE FILLED
 011212     000                               .BYTE   0                       ;MASK OF INVALID BITS FOR THIS PARAMETER
 011213     001                               .BYTE   1                       ;NUMBER OF PARAMETERS TO STORE
 011214  012700  012216                       MOV     #TST1,R0
 011220  022710  000004             5$:       CMP     #4,(R0)
```

```
011224  001015                                  BNE     6$
011226  02276↑B0          012737  000002        CMP     #12737,2(R0)
011234  001011                                  BNE     6$
011236  023760  001122    000004                CMP     $TSTNM,4(R0)      ;IS THIS THE TEST ?
011244  001005                                  BNE     6$                ;IF NOT, DON'T PROCESS NUMBER
011246  010037  001126                          MOV     R0,$LPADR         ;SAVE PC
011252  104402  001231                          TYPE    ,$CRLF
011256  000412                                  BR      8$
011260  005720                          6$:     TST     (R0)+
011262  020027  022156                          CMP     R0,#TLAST+10
011266  001354                                  BNE     5$
011270  104402  001230                          TYPE    ,$QUES
011274  000736                                  BR      7$
011276  012737  012216    001126        4$:     MOV     #TST1,$LPADR      ;PREPARE TEST ADDRESS
011304                                  8$:
011304  0↑B00177          167616                RESTART:JMP     @$LPADR           ;GO START TESTING.***WARNING!****
                                                                          ;THIS JUMP IS USED BY POWER UP ROUTINE!!!!

```

```
                                                ;-ROUTINE USED TO SET UP THE DIAGNOSTIC VIA APT.
                                                ; IF BIT7 IN THE ENVIRONMENT MODE ($ENVM) BYTE IS SET,
                                                ; THE PROGRAM WILL LOAD ITS PARAMETERS FROM THE ETABLE.

011310  012700  001500                  SETAPT: MOV     #DZ.MAP,R0      ;POINT TO THE DEVICE MAP TABLE
011314  013701  001310                          MOV     $BASE,R1        ;BUILD DEVICE ADDRESSES IN R1
011320  013702  001304                          MOV     $VECT1,R2       ;BUILD DEVICE VECTORS IN R2
011324  042702  177007                          BIC     #↑C<770>,R2     ;STRIP AWAY OTHER INFORMATION

011330  113703  001305                          MOVB    $VECT1+1,R3     ;LOAD THE INTERRUPT PRIORITY FROM R3
011334  106003                                  RORB    R3              ;ALIGN THE NUMBER
011336  106003                                  RORB    R3              ;ALIGN THE NUMBER
011340  106003                                  RORB    R3              ;ALIGN THE NUMBER
011342  106003                                  RORB    R3              ;ALIGN THE NUMBER
011344  106003                                  RORB    R3              ;ALIGN THE NUMBER
011346  042703  177770                          BIC     #↑C<7>,R3       ;REMOVE ALL BUT BUS LEVEL NUMBER
011352  012704  001320                          MOV     #$DDW0,R4       ;POINT TO THE BEGINNING OF DEVICE↑B PARAMETERS
011356  013705  001312                          MOV     $DEVM,R5        ;GET THE MAP OF ACTIVE DEVICES
011362  010537  001404                          MOV     R5,DZACTV       ;SAVE THE BIT MAP
011366  006005                          1$:     ROR     R5              ;GET A DEVICE SELECTION BIT
011370  103407                                  BCS     3$              ;IF IT IS SELECTED, GO SET UP A MAP
011372  001425                                  BEQ     5$              ;IF NO MORE ARE SELECTED, GET OUT OF SETUP
011374  005724                                  TST     (R4)+           ;POINT TO NEXT DEVICE DESCRIPTOR
011376  062701  000010                  2$:     ADD     #10,R1          ;SET UP THE NEXT ADDRESS
011402  062702  000010                          ADD     #10,R2          ;SET UP THE NEXT VECTOR GROUP
011406  000767                                  BR      1$              ;GO SEE IF MORE DEVICES REMAIN
        011410  010120                          3$:     MOV     R1,(R0)+        ;LOAD DEVICE ADDRESS
011412  010220                                  MOV     R2,(R0)+        ;LOAD THE VECTOR ADDRESS
011414  010320                                  MOV     R3,(R0)+        ;LOAD THE INTERRUPT PRIORITY LEVEL
011416  013720  001314                          MOV     $CDW1,(R0)+     ;GET THE NUMBER OF LINES IN OPERATION
011422  012420                                  MOV     (R4)+,(R0)+     ;LOAD DEVICE PARAMETERS
011424  100406                                  BMI     4$              ;IF 20MA MODE SELECTED, SET IT UP
011426  052760  100000  177772                  BIS     #100000,-6(R0)  ;SET THE 20MA FLAG IN DZLVN
011434  042760  100000  177776                  BIC     #100000,-2(R0)  ;CLEAR THE FLAG IN DZPARN
011442  005020                          4$:     CLR     (R0)+           ;DEFAULT OPERATION TO INTERNAL MAINTENANCE MODE
011444  000754                                  BR      2$              ;GO BUILD THE NEXT ADDRESS
011446  012710  177777                  5$:     MOV     #-1,(R0)        ;TERMINATE THE DEVICE MAP
011452  012737  001256  001160                  MOV     #$SWREG,SWR     ;SET TO SOFTWARE APT SWITCH REGISTER
011460  000207                                  RTS     PC              ;RETURN TO PRINT STATUS TABLE


                                                ;*ROUTINE USED TO "AUTO SIZE" THE DZ11
                                                ;*CSR AND VECTOR.
                                                ;*NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
                                                ;*       ADDRESS RANGE (160000:163700)
                                                ;*       AND THE VECTOR MAY BE ANY WHERE IN THE
                                                ;*       FLOATING VECTOR RANGE (300:770)
                                                ;*

011462                                  AUTO.SIZE:
011462  000005                                  RESET                   ;INSURE A BUS INIT.
011464  105337  001415                          DECB    INIFLG          ;SHOW THAT I WAS HERE
011470  012702  001500                  CSRMAP: MOV     #DZ.MAP,R2      ;LOAD MAP POINTER.
011474  012703  001320                          MOV     #$DDW0,R3       ;POINT TO ETABLE DEVICE DESCRIPTOR WORDS
011500  005022                          1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
011502  022702  002000                          CMP     #DZ.END,R2      ;ALL DONE?
```

```
011506  001374                            BNE     1$              ;BR IF NO
011510  105037  001410                    CLRB    DZNUM           ;SET OCTAL NU↑BMBER OF DZ11'S TO 0
011514  012702  001500                    MOV     #DZ.MAP,R2
011520  012701  160000                    MOV     #160000,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
011524  012737  012020  000004            MOV     #6$,@#4         ;SET FOR NON-EXISTENT DEVICE TIME OUT
011532  052711  000040          2$:       BIS     #BIT5,(R1)      ;TRY TO SET MASTER SCAN ENABLE
011536  052761  000200  000004            BIS     #BIT7,4(R1)     ;TRY TO TRANSMIT ON LINE 7
011544  005000                            CLR     R0              ;USE R0 AS A COUNTER
011546  005711                  7$:       TST     (R1)            ;HAS TRANSMITTER READY COME UP?
011550  100403                            BMI     8$              ;IF SO, GO GET A FINAL CHECK
011552  005300                            DEC     R0              ;REDUCE COUNT. TIME UP?
011554  001374                            BNE     7$              ;IF NOT, KEEP WAITING
011556  000451                            BR      3$              ;ASSUME IT'S NOT A DZ11
011560  032761  000200  000004  8$:       BIT     #BIT7,4(R1)     ;IS LINE 7 ENABLE STILL SET? IT SHOULD BE
011566  001445                            BEQ     3$              ;IF IT'S NOT, ASSUME IT'S NOT A DZ11
011570  032711  000040                    BIT     #BIT5,(R1)      ;IS MASTER SCAN ENABLE STILL SET?
011574  001442                            BEQ     3$              ;IF NOT, ASSUME IT'S NOT A DZ11
011576  005000                            CLR     R0
011600  052711  000020                    BIS     #20,(R1)        ;SET DEVICE CLEAR
011604  032711  000020                    BIT     #20,(R1)        ;SHOULD STAY SET FOR A WHILE IF DZ
011610  001434                            BEQ     3$              ;BR IF NOT ↑BDZ11
011612  032711  000020                    BIT     #20,(R1)        ;WAIT FOR BIT TO CLEAR
011616  001404                            BEQ     .+12            ;BR WHEN CLEARED
011620  104414                            DELAY
011622  005200                            INC     R0
011624  001372                            BNE     .-12
011626  000425                            BR      3$              ;BIT NOT CLEARED! MUST NOT BE DZ11
011630  005011                            CLR     (R1)            ;GET RID OF MASTER SCAN ENABLE
011632  005061  000004                    CLR     4(R1)           ;GET RID OF LINE 7 ENABLE
                                ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DZ11 CSR ADDRESS.
011636  010122                            MOV     R1,(R2)+        ;STORE CSR IN CORE TABLE.
011640  005722                            TST     (R2)+           ;POP OVER VECTOR STORE AREA
011642  ↑B012722        000005            MOV        #5,(R2)+     ;SET THE DEFAULT BUS LEVEL
011646  012722  000377                    MOV     #377,(R2)+      ;SET THE DEFAULT LINE SELECTION PARAMETER
011652  012712  017470                    MOV     #17470,(R2)     ;SET THE DEFAULT PARAMETERS
011656  012223                            MOV     (R2)+,(R3)+     ;COPY PARAMETERS INTO ETABLE DESCRIPTOR
011660  005022                            CLR     (R2)+           ;SET THE DEFAULT MODE OF OPERATION
011662  012712  177777                    MOV     #-1,(R2)        ;TERMINATE LIST
011666  105237  001410                    INCB    DZNUM           ;UPDATE DEVICE COUNTER
011672  122737  000020  001410            CMPB    #20,DZNUM       ;ARE MAX. NO. OF DEV FOUND?
011700  001405                            BEQ     100$            ;YES DON'T LOOK FOR ANY MORE.
011702  062701  000010          3$:       ADD     #10,R1          ;UPDATE CSR POINTER ADDRESS
011706  022701  163700                    CMP     #163700,R1
011712  001307                            BNE     2$              ;BR IF MORE ADDRESS TO CHECK.
011714                          100$:
011714  105737  001410                    TSTB    DZNUM           ;WERE ANY DZ11'S FOUND AT ALL?
011720  001432                            BEQ     5$              ;ERROR AUTO SIZER FOUND NO DZ11'S IN THIS SYS.
011722  113701  001410                    MOVB    DZNUM,R1
011726  110137  001411                    MOVB    R1,SAVNUM       ;SAVE NUMBER OF DEVICES
011732  012737  000001  001404            MOV     #1,DZACTV
011740  005301                  4$:       DEC     R1
011742  001404                            BEQ     98$
011744  000261                            SEC
011746↑B        006137  001404            ROL        DZACTV
011752  000772                            BR      4$
011754  013737  001500  001310  98$:      MOV     DZCR0,$BASE     ;POINT TO THE ADDRESS OF FIRST DEVICE
```

```
011762  013737  001512  001314          MOV     MANTO,$CDW1        ;INDICATE TO ETABLE WHAT MODE IS BEING USED
011770  012737  000006  000004  99$:    MOV     #6,@#4             ;RESTORE TRAP VECTOR
011776  013737  001404  001312          MOV     DZACTV,$DEVM       ;SAVE ACTIVE REGISTER
012004  000410                          BR      VECMAP             ;GO FIND THE VECTOR NOW.
012006  104402  010005          5$:     TYPE    ,MERR2             ;NOTIFY OPR↑B THAT NO DZ11'S FOUND.
012012  005000                          CLR     R0                 ;MAKE DATA DISPLAY ZERO
012014  000000                          HALT                       ;STOP THE SHOW
012016  000776                          BR      .-2                ;DISABLE CONT. SW.
012020  012716  011702          6$:     MOV     #3$,(SP)           ;ENTERED BY NON-EXISTENT TIME-OUT
012024  000002                          RTI                        ;RETURN TO MAINSTREAM

012026  012737  000340  000022  VECMAP: MOV     #340,@#22          ;SET IOT TRAP PRIORITY TO 7
012034  012737  012150  000020          MOV     #4$,@#20           ;SET IOT TRAP VECTOR
012042  012702  001500                  MOV     #DZ.MAP,R2         ;SET SOFTWARE POINTER
012046  012700  000300                  MOV     #300,R0            ;FLOATING VECTORS START HERE.
012052  012701  000302                  MOV     #302,R1            ;PC OF IOT INSTR.
012056  010120                  1$:     MOV     R1,(R0)+           ;START FILLING VECTOR AREA
012060  012721  000004                  MOV     #4,(R1)+           ;WITH .+2; IOT
012064  022021                          CMP     (R0)+,(R1)+        ;ADD 2 TO R0 +R1
012066  020127  001000                  CMP     R1,#1000           ;HAS THE VECTOR AREA BEEN EXCEEDED?
012072  101771                          BLOS    1$                 ;BR IF MORE TO FILL
012074  013704  001404                  MOV     DZACTV,R4          ;STORE TEMPORARILY
012100  006004                  2$:     ROR     R4                 ;BRING OUT A BIT
012102  103036                          BCC     5$                 ;BR IF ALL DONE
012104  106427  000000                  MTPS    #0                 ;ZERO CPU PRIO
012110  012772  040040  000000          MOV     #BIT14+BIT5,@(R2)↑B
012116  011201                          MOV     (R2),R1            ;GET CSR
012120  112761  000200  000004          MOVB    #BIT7,4(R1)        ;SET THE TCR BIT!
                                                                   ;ATTEMPT TO FORCE AN INTERRUPT
012126  005200                          INC     R0                 ;STALL
012130  001376                          BNE     .-2                ;        FOR TIME TO INTERRUPT
012132  012762  000300  000002          MOV     #300,2(R2)         ;NO INTERRUPT ASSUME 300 AND FIX DZ11 LATER
012140  000005                  3$:     RESET                      ;INIT
012142  062702  000014                  ADD     #14,R2             ;POP SOFTWARE POINTER
012146  000754                          BR      2$                 ;KEEP GOING
012150  011662  000002          4$:     MOV     (SP),2(R2)         ;GET VECTOR ADDRESS
012154  162762  000010  000002          SUB     #10,2(R2)          ;POINT B↑BACK TO THE CORRECT VECTOR
012162  042762  000007  000002          BIC     #7,2(R2)           ;CLEAR JUNK
012170  022626                          POP2SP                     ;POP IOT JUNK OFF STACK
012172  012716  012140                  MOV     #3$,(SP)           ;SET FOR RETURN
012176  000002                          RTI
012200  013737  001502  001304  5$:     MOV     DZVCO,$VECT1       ;COPY VECTOR OF FIRST DEVICE INTO ETABLE
012206  012737  004654  000020          MOV     #.SCOPE,IOTVEC     ;RESTORE THE SCOPE TRAP
012214  000207                          RTS     PC                 ;ALL DONE WITH "AUTO SIZING"
```

```

                                               ;************************ TEST 1 ****************************
                                               ;*THIS TEST PROVES THE SLAVE SYNC RESPONSE
                                               ;*DURING A READ OR WRITE TO THE FOLLOWING ADDRESS:
                                               ;*        DZCSR, DZRBUF, DZTCR, DZMSR
                                       ::*  TEST 1
                                       ;;*********************************************************
 012216  000004                        TST1:   SCOPE
 012220  012737  000001  001122                MOV      #1,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
 012226  012737  012406  001360                MOV      #TST2,NEXT         ;POINT TO THE START OF THE NEXT TEST
 012234  012737  012374  000004                MOV      #5$,4              ;SET TRAP VECTOR
 012242  01↑B2737        000340  000006        MOV      #PR7,6                  ;SET PRIORITY TO LEVEL 7
 012250  012737  012256  001362                MOV      #1$,LOCK           ;SET RETURN IF SW09=11
 012256  013700  002042                1$:     MOV      DZCSR,R0           ;SET ADDRESS TO TEST
 012262  011001                                MOV      (R0),R1            ;READ THE ADDRESS
 012264  000240                                NOP                         ;WASTE TIME
 012266  005010                                CLR      (R0)               ;WRITE THE ADDRESS
 012270  000240                                NOP                         ;WASTE TIME
 012272  012737  012300  001362                MOV      #2$,LOCK           ;SET RETURN ADDRESS FOR SW09
 012300  013700  002046                2$:     MOV      DZRBUF,R0          ;SET ADDRESS TO TEST
 012304  011001                                MOV      (R0),R1            ;READ THE ADDRESS
 012306  000240                                NOP                         ;
 0↑B12310        005010                                CLR      (R0)               ;WRITE THE ADDRESS
 012312  000240                                NOP                         ;WASTE TIME
 012314  012737  012322  001362                MOV      #3$,LOCK           ;SET RETURN ADDRESS FOR SW09
 012322  013700  002056                3$:     MOV      DZTCR,R0           ;SET ADDRESS TO TEST
 012326  011001                                MOV      (R0),R1            ;READ THE ADDRESS
 012330  000240                                NOP
 012332  005010                                CLR      (R0)               ;WRITE THE ADDRESS
 012334  000240                                NOP
 012336  012737  012344  001362                MOV      #4$,LOCK           ;SET RETURN ADDRESS
 012344  013700  002062                4$:     MOV      DZMSR,R0           ;SET ADDRESS TO TEST
 012350  011001                                MOV      (R0),R1            ;READ FROM ADDRESS
 012352  000240                                NOP
 012354  005010                                CLR      (R0)               ;WRITE THE ADDRESS
 012356  000240                                NOP
 012360  012737  000006  000004                MOV      #6,4               ;SET TRAP CATCHER BACK TO NORMAL
 012366  005037  000006                        CLR      6
 012372  104400                                ADVANCE                     ;SCOPE THIS TEST
 012374  011601                        5$:     MOV      (SP),R1            ;SAVE PC OF TRAP
 012376  022626                                CMP      (SP)+,(SP)+        ;POP TRAP OFF STACK
 012400  104001                                ERROR    1                  ;*NO SLAVE SYNC RESPONSE.
 012402  104401                                SCOP1                       ;SW09=1?
 012404  000111                                JMP      (R1)               ;RTI
                                               ;********************** TEST 2 ******************************
                                               ;*THIS TEST PROVES THAT BIT "DCLR"
 ↑B                                            ;*CAN BE SET AND THAT IT WILL CLEAR
                                               ;*BY ITSELF AFTER A PERIOD OF TIME.
                                       ::*  TEST 2
                                       ;;*********************************************************
 012406  000004                        TST2:   SCOPE
 012410  012737  000002  001122                MOV      #2,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
 012416  012737  012472  001360                MOV      #TST3,NEXT         ;POINT TO THE START OF THE NEXT TEST
 012424  013700  002042                        MOV      DZCSR,R0           ;SET POINTER
 012430  012705  000020                        MOV      #DCLR,R5           ;SET DCLR
 012434  010510                                MOV      R5,(R0)            ;WRITE DCLR INTO DZCSR
```

```
012436  011004                               MOV     (R0),R4         ;READ BACK DZCSR
012440  020504                               CMP     R5,R4           ;DZCSR OK?
012442  001401                               BEQ     1$              ;IF IT IS SET SKIP THE ERROR CALL
012444  104002                               ERROR   2               ;*DCLR SHOULD BE SET..MOMENTARILY
                                     ;NOW LETS WATCH IT DISAPPEAR
012446  005002                       1$:     CLR     R2              ;SET COUNTER TO 0
012450  005005                               CLR     R5              ;SET EXPECTED TO 0
012452  005003                               CLR     R3              ;DUAL LOOP COUNTER
012454  011004                       2$:     MOV     (R0),R4         ;IS DCLR CLEAR?
012456  001405                               BEQ     3$              ;IF YES , GO TO THE NEXT TEST
012460  005203                               INC     R3              ;IF NO COUNT 1 OF 65535 TICKS
                                                                     ;THE WORD CREATED BY THE IMMEDIATE 0 WILL BE
                                                                     ;THE COUNTER
012462  001374                               BNE     2$              ;HAS THE TIME EXPIRED? IF NO, GO TEST BIT AGAIN
012464  005302                               DEC     R2              ;HAS THE TOTAL TIME EXPIRED?
012466  001372                               BNE     2$              ;IF NO, CHECK THE BIT AGAIN
012470  104002                               ERROR   2               ;*DCLR FAILED TO CLEAR
012472                               3$:
                                             ;*********************** TEST 3 ******************************
                                             ;*TEST TO VERIFY THAT BIT "MAINT" CAN
                                             ;*BE SET. THEN VERIFY THAT BIT "MAINT" CAN
                                     ↑B      ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                             ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                                             ;*CLEARED BY A "DEVICE CLEAR"
                                     ;;*  TEST 3
                                     ;;**********************************************************
012472  000004                       TST3:   SCOPE
012474  012737  000003  001122               MOV     #3,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
012502  012737  012564  001360               MOV     #TST4,NEXT      ;POINT TO THE START OF THE NEXT TEST
012510  013700  002042                       MOV     DZCSR,R0        ;GET BASE ADDRESS
012514  012705  000010                       MOV     #MAINT,R5       ;SET BIT
012520  010510                               MOV     R5,(R0)         ;SET SET IN DEVICE
01↑B2522        011004                       MOV         (R0),R4         ;READ THE BIT FROM DEVICE
012524  020504                               CMP     R5,R4           ;WAS BIT SET?
012526  001401                               BEQ     1$              ;BR IF YES
012530  104002                               ERROR   2               ;*BIT R/W FAILURE
012532  040510                       1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
012534  011004                               MOV     (R0),R4         ;READ DEVICE
012536  001404                               BEQ     2$              ;BR IF BITS WERE CLEARED.
012540  010546                               MOV     R5,-(SP)        ;SAVE THE BIT
012542  005005                               CLR     R5              ;SET EXPECTED RESULTS TO 0
012544  104002                               ERROR   2               ;*BIT FAILED TO CLEAR
012546  012605                               MOV     (SP)+,R5        ;RESTORE THE BIT.
012550  010510                       2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
012552  104413                               DEVICE.CLR              ;ISSUE DEVICE CLEAR
012554  011004                               MOV     (R0),R4         ;READ THE BIT.
012556  001402                               BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
012560  005005                               CLR     R5              ;SET EXPECTED TO ZERO
012562  104002                               ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
012564                               3$:
                                             ;*********************** TEST 4 ******************************
                                             ;*TEST TO VERIFY THAT BIT "MSENAB" CAN
                                             ;*BE SET. THEN VERIFY THAT BIT "MSENAB" CAN
                                             ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                             ;*VERIFY THAT AFTER BEING SET AGAIN IT CA↑BN BE
                                             ;*CLEARED BY A "DEVICE CLEAR"
```

```
                                                ;;*  TEST 4
                                                ;;*************************************************************
        012564  000004                          TST4:   SCOPE
        012566  012737  000004  001122                  MOV     #4,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
        012574  012737  012656  001360                  MOV     #TST5,NEXT      ;POINT TO THE START OF THE NEXT TEST
        012602  013700  002042                          MOV     DZCSR,R0        ;GET BASE ADDRESS
        012606  012705  000040                          MOV     #MSENAB,R5      ;SET BIT
        0↑B12612        010510                                  MOV     R5,(R0)         ;SET SET IN DEVICE
        012614  011004                                  MOV     (R0),R4         ;READ THE BIT FROM DEVICE
        012616  020504                                  CMP     R5,R4           ;WAS BIT SET?
        012620  001401                                  BEQ     1$              ;BR IF YES
        012622  104002                                  ERROR   2               ;*BIT R/W FAILURE
        012624  040510                          1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
        012626  011004                                  MOV     (R0),R4         ;READ DEVICE
        012630  001404                                  BEQ     2$              ;BR IF BITS WERE CLEARED.
        012632  010546                                  MOV     R5,-(SP)        ;SAVE THE BIT
        012634  005005                                  CLR     R5              ;SET EXPECTED RESULTS TO 0
        012636  104002                                  ERROR   2               ;*BIT FAILED TO CLEAR
        012640  012605                                  MOV     (SP)+,R5        ;RESTORE THE BIT.
        012642  010510                          2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
        012644  104413                                  DEVICE.CLR              ;ISSUE DEVICE CLEAR
        012646  011004                                  MOV     (R0),R4         ;READ THE BIT.
        012650  001402                                  BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
        012652  005005                                  CLR     R5              ;SET EXPECTED TO ZERO
        012654  104002                                  ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
        012656                                  3$:

                                                        ;*********************** TEST 5 ****************************
                                                        ;*TEST TO VERIFY THAT BIT "SILOEN" CAN
                                                        ;*BE SET. THEN VERIFY THAT BIT "SILOEN" CAN
                                                        ;*BE CLEARED (WRITTEN TO A ZERO). AND FINA↑BLLY
                                                        ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                                                        ;*CLEARED BY A "DEVICE CLEAR"
                                                ;;*  TEST 5
                                                ;;*************************************************************
        012656  000004                          TST5:   SCOPE
        012660  012737  000005  001122                  MOV     #5,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
        012666  012737  012750  001360                  MOV     #TST6,NEXT      ;POINT TO THE START OF THE NEXT TEST
        012674  013700  002042                          MOV     DZCSR,R0        ;GET BASE ADDRESS
        012700  012705  010000                          MOV     #SILOEN,R5      ;SET BIT
        012704  010510                                  MOV     R5,(R0)         ;SET SET IN DEVICE
        012706  011004                                  MOV     (R0),R4         ;READ THE BIT ↑BFROM DEVICE
        012710  020504                                  CMP     R5,R4           ;WAS BIT SET?
        012712  001401                                  BEQ     1$              ;BR IF YES
        012714  104002                                  ERROR   2               ;*BIT R/W FAILURE
        012716  040510                          1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
        012720  011004                                  MOV     (R0),R4         ;READ DEVICE
        012722  001404                                  BEQ     2$              ;BR IF BITS WERE CLEARED.
        012724  010546                                  MOV     R5,-(SP)        ;SAVE THE BIT
        012726  005005                                  CLR     R5              ;SET EXPECTED RESULTS TO 0
        012730  104002                                  ERROR   2               ;*BIT FAILED TO CLEAR
        012732  012605                                  MOV     (SP)+,R5        ;RESTORE THE BIT.
        012734  010510                          2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
        012736  104413                                  DEVICE.CLR              ;ISSUE DEVICE CLEAR
        012740  011004                                  MOV     (R0),R4         ;READ THE BIT.
        012742  001402                                  BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
        012744  005005                                  CLR     R5              ;SET EXPECTED TO ZERO
```

```
012746  104002                              ERROR   2                       ;*BIT NOT CLEARED BY DEVICE CLEAR
012750                              3$:
                                            ;*********************** TEST 6 *****************************
                                            ;*TEST TO VERIFY THAT BIT "RIE" CAN
                                            ;*BE SET. THEN VERIFY THAT BIT "RIE" CAN
                                                    ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                            ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                                            ;*CLEARED BY A "DEVICE CLEAR"
                                    ;;*  TEST 6
                                    ;;*******************************************************
012750  000004                      TST6:   SCOPE
012752  012737  000006  001122              MOV     #6,$TSTNM               ;LOAD THE NUMBER OF THIS TEST
012760  012737  013042  001360              MOV     #TST7,NEXT              ;POINT TO THE START OF THE NEXT TEST
012766  013700  002042                      MOV     DZCSR,R0                ;GET BASE ADDRESS
012772  012705  000100                      MOV     #RIE,R5 ;SET BIT
012776  010510                              MOV     R5,(R0)                 ;SET SET IN DEVICE
        013000  011004                      MOV             (R0),R4                 ;READ THE BIT FROM DEVICE
013002  020504                              CMP     R5,R4                   ;WAS BIT SET?
013004  001401                              BEQ     1$                      ;BR IF YES
013006  104002                              ERROR   2                       ;*BIT R/W FAILURE
013010  040510                      1$:     BIC     R5,(R0)                 ;CLEAR THE BIT.
013012  011004                              MOV     (R0),R4                 ;READ DEVICE
013014  001404                              BEQ     2$                      ;BR IF BITS WERE CLEARED.
013016  010546                              MOV     R5,-(SP)                ;SAVE THE BIT
013020  005005                              CLR     R5                      ;SET EXPECTED RESULTS TO 0
013022  104002                              ERROR   2                       ;*BIT FAILED TO CLEAR
013024  012605                              MOV     (SP)+,R5                ;RESTORE THE BIT.
013026  010510                      2$:     MOV     R5,(R0)                 ;SET THE BIT AGAIN
013030  104413                              DEVICE.CLR                      ;ISSUE DEVICE CLEAR
013032  011004                              MOV     (R0),R4                 ;READ THE BIT.
013034  001402                              BEQ     3$                      ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
013036  005005                              CLR     R5                      ;SET EXPECTED TO ZERO
013040  104002                              ERROR   2                       ;*BIT NOT CLEARED BY DEVICE CLEAR
013042                              3$:
                                            ;*********************** TEST 7 *****************************
                                            ;*TEST TO VERIFY THAT BIT "TIE" CAN
                                            ;*BE SET. THEN VERIFY THAT BIT "TIE" CAN
                                            ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                            ;*VERIFY THAT AFTER BEING SET AGAIN IT C↑BAN BE
                                            ;*CLEARED BY A "DEVICE CLEAR"
                                    ;;*  TEST 7
                                    ;;*******************************************************
013042  000004                      TST7:   SCOPE
013044  012737  000007  001122              MOV     #7,$TSTNM               ;LOAD THE NUMBER OF THIS TEST
013052  012737  013134  001360              MOV     #TST10,NEXT             ;POINT TO THE START OF THE NEXT TEST
013060  013700  002042                      MOV     DZCSR,R0                ;GET BASE ADDRESS
013064  012705  040000                      MOV     #TIE,R5 ;SET BIT
013070  010510                              MOV     R5,(R0)                 ;SET SET IN DEVICE
013072  011004                              MOV     (R0),R4                 ;READ THE BIT FROM DEVICE
013074  020504                              CMP     R5,R4                   ;WAS BIT SE↑BT?
013076  001401                              BEQ     1$                      ;BR IF YES
013100  104002                              ERROR   2                       ;*BIT R/W FAILURE
013102  040510                      1$:     BIC     R5,(R0)                 ;CLEAR THE BIT.
013104  011004                              MOV     (R0),R4                 ;READ DEVICE
013106  001404                              BEQ     2$                      ;BR IF BITS WERE CLEARED.
013110  010546                              MOV     R5,-(SP)                ;SAVE THE BIT
```

```
  013112  005005                               CLR     R5              ;SET EXPECTED RESULTS TO 0
  013114  104002                               ERROR   2               ;*BIT FAILED TO CLEAR
  013116  012605                               MOV     (SP)+,R5        ;RESTORE THE BIT.
  013120  010510                       2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
  013122  104413                               DEVICE.CLR              ;ISSUE DEVICE CLEAR
  013124  011004                               MOV     (R0),R4         ;READ THE BIT.
  013126  001402                               BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
  013130  005005                               CLR     R5              ;SET EXPECTED TO ZERO
  013132  104002                               ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
  013134                               3$:
                                               ;*********************** TEST 10 ****************************
                                               ;*THIS TESTS THAT ALL OF THE FOLLOWING
                                               ;*BITS CAN BE: SET, CLEARED, CLEARED BY "DEVICE CLEAR "
                                               ;*BITS TESTED ARE:
                                 ↑B            ;* TCR0, TCR1, TCR2, TCR3, TCR4, TCR5, TCR6, TCR7
                                       ;;*  TEST 10
                                       ;;***********************************************************
  013134  000004                       TST10:  SCOPE
  013136  012737  000010  001122               MOV     #10,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
  013144  012737  013272  001360               MOV     #TST11,NEXT     ;POINT TO THE START OF THE NEXT TEST
  013152  013700  002056                       MOV     DZTCR,R0        ;SET DEVICE ADDRESS
  013156  012705  000001                       MOV     #TCR0,R5        ;SET EXPECTED RESULTS
  013162  012737  013170  001362               MOV     #1$,LOCK        ;SET FOR SW09
  013170  010510                       1$:     MOV     R5,(R0)         ;SET THE BIT
  013172  011004                               MOV↑B   (R0),R4         ;READ THE BIT FROM THE DEVICE
  013174  042704  177400                       BIC     #↑C<377>,R4     ;CLEAR HIGH BYTE
  013200  020504                               CMP     R5,R4           ;WAS BIT OK?
  013202  001401                               BEQ     2$              ;BR IF YES
  013204  104002                               ERROR   2               ;*BIT FAILED TO SET.
  013206  040510                       2$:     BIC     R5,(R0)         ;CLEAR THE BIT
  013210  011004                               MOV     (R0),R4         ;READ THE REGISTER
  013212  042704  177400                       BIC     #↑C<377>,R4     ;CLEAR HIGH BYTE
  013216  005704                               TST     R4              ;BITS CLEAR?
  013220  001404                               BEQ     3$              ;BR IF YES
  013222  010546                               MOV     R5,-(SP)        ;SAVE GOOD RESULTS
  013224  005005                               CLR     R5              ;SET EXPECTED TO 0
  013226  104002                               ERROR   2               ;*REPORT BIT NOT CLEAR
  013230  012605                               MOV     (SP)+,R5        ;RESTORE R5
  013232  010510                       3$:     MOV     R5,(R0)         ;SET THE BIT AGAIN.
  013234  104413                               DEVICE.CLR              ;ISSUE DEVICE CLEAR
  013236  011004                               MOV     (R0),R4         ;READ THE REGISTER
  013240  042704  177400                       BIC     #↑C<377>,R4     ;CLEAR HIGH BYTE
  013244  005704                               TST     R4              ;BITS CLEAR?
  013246  001404                               BEQ     4$              ;BR IF YES
  013250  010546                               MOV     R5,-(SP)        ;SAVE GOOD RESULTS
  013252  005005                               CLR     R5              ;SET EXPECTED TO 0
  013254  104002                               ERROR   2               ;*REPORT BIT NOT CLEAR
  013256  012605                               MOV     (SP)+,R5        ;RESTORE R5
↑B013260  104401                       4$:     SCOP1                   ;LOCK ON BIT? SET SW09=1
  013262  106305                               ASLB    R5              ;CHANGE TO NEXT BIT
  013264  001341                               BNE     1$              ;CONTINUE TESTING
  013266  005037  001362                       CLR     LOCK            ;MAKE SURE TIGHT LOOP IS CLEANED UP
                                               ;*********************** TEST 11 ****************************
                                               ;*THIS TESTS THAT ALL OF THE FOLLOWING
                                               ;*BITS CAN BE: SET, CLEARED, CLEARED BY "RESET INSTR *NOT* DEVICE CLEAR "
                                               ;*BITS TESTED ARE:
                                                 ;* DTR0, DTR1, DTR2, DTR3, DTR4, DTR5, DTR6, DTR7
                                                 ;*THIS TEST IS NOT DONE IF MODULE IS 20MA VERSION
                                        ;;*  TEST 11
                                        ;;*************************************************************
 013272  000004                         TST11:   SCOPE
 013274  012737  000011  001122                  MOV     #11,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
 013302  012737  013446  001360                  MOV     #TST12,NEXT         ;POINT TO THE START OF THE NEXT TEST
 013310  013700  002056                          MOV     DZTCR,R0            ;SET DEVICE ADDRESS
 013314  012705  000400                          MOV     #DTR0,R5            ;SET EXPECTED RESULTS
 013320  012737  013336  001362                  MOV     #1$,LOCK            ;SET FOR SW09
 013326  105737  001414                          TSTB    EIAFLG              ;20MA OR EIA
 013332  100001                                  BPL     1$                  ;BR IF EIA
 013334  104400                                  ADVANCE                     ;EXIT TEST
 013336  010510                         1$:      MOV     R5,(R0)             ;SET THE BIT
 013340  011004                                  MOV     (R0),R4             ;READ THE BIT FROM THE DEVICE
 013342  105004                                  CLRB    R4                  ;CLEAR LOW BYTE
 013344  020504                                  CMP     R5,R4               ;WAS BIT OK?
 013346  001401                                  BEQ     2$                  ;BR IF YES
 013350  104002                                  ERROR   2                   ;*BIT FAILED TO SET.
 013352  040510                         2$:      BIC     R5,(R0)             ;CLEAR THE BIT
 013354  011004                                  MOV     (R0),R4             ;READ THE REGISTER
 013356  105004                                  CLRB    R4                  ;CLEAR LOW BYTE
 ↑B013360        005704                          TST          R4                   ;BITS CLEAR?
 013362  001404                                  BEQ     3$                  ;BR IF YES
 013364  010546                                  MOV     R5,-(SP)            ;SAVE GOOD RESULTS
 013366  005005                                  CLR     R5                  ;SET EXPECTED TO 0
 013370  104002                                  ERROR   2                   ;*REPORT BIT NOT CLEAR
 013372  012605                                  MOV     (SP)+,R5            ;RESTORE R5
 013374  010510                         3$:      MOV     R5,(R0)             ;SET THE BIT AGAIN.
 013376  104413                                  DEVICE.CLR                  ;ISSUE DEVICE CLEAR
 013400  011004                                  MOV     (R0),R4             ;READ THE REGISTER
 013402  105004                                  CLRB    R4                  ;CLEAR LOW BYTE
 013404  030510                                  BIT     R5,(R0)             ;WAS BIT CLEARED BY DEVICE.CLR?
 013406  001001                                  BNE     .+4                 ;BR IF NO (IT↑B SHOULDN'T BE CLEAR)
 013410  104002                                  ERROR   2                   ;*BIT CLEARED BY DEVICE.CLR
 013412  000005                                  RESET                       ;ISSUE A BUS INIT
 013414  011004                                  MOV     (R0),R4             ;READ REGISTER
 013416  105004                                  CLRB    R4                  ;CLEAR LOW BYTE
 013420  005704                                  TST     R4                  ;BITS CLEAR?
 013422  001404                                  BEQ     4$                  ;BR IF YES
 013424  010546                                  MOV     R5,-(SP)            ;SAVE GOOD RESULTS
 013426  005005                                  CLR     R5                  ;SET EXPECTED TO 0
 013430  104002                                  ERROR   2                   ;*REPORT BIT NOT CLEAR
 013432  012605                                  MOV     (SP)+,R5            ;RESTORE R5
 013434  104401                         4$:      SCOP1                       ;LOCK ON BIT? SET SW09=1
 013436  106305                                  ASLB    R5                  ;CHANGE TO NEXT BIT
 013440  001336                                  BNE     1$                  ;CONTINUE TESTING
 013442  005037  001362                          CLR     LOCK                ;MAKE SURE TIGHT LOOP IS CLEANED UP
                                                 ;************************ TEST 12 ****************************
                                                 ;*THIS TEST PERFORMS RESET TESTING &
                                                 ;*TESTING OF WRITE ONLY OR READ ONLY BIT
                                                 ;* TEST BITS "RDONE, BIT11, BIT10, BIT9, BIT8, BIT2, BIT1
                                                 ;*            BIT0, SILOAL" ARE READ ONLY AND THAT TRDY IS
                                                 ;*            ZERO UNTIL A LINE IS SELECTED AND MSENAB IS SET.
                                                 ;*
                                        ;;*  TEST 12
```

```
                                          ;;*************************************************************
013446  000004                            TST12:  SCOPE
013450  012737  000012  001122                    MOV     #12,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
013456  012737  013564  001360                    MOV     #TST13,NEXT         ;POINT TO THE START OF THE NEXT TEST
013464  013700  002042                            MOV     DZCSR,R0            ;SET ADDRESS TO R0
013470  005005                                    CLR     R5                  ;SET EXPECTED TO 0
013472  012710  027607                            MOV     #RDONE+BIT11+BIT10+BIT9+BIT8+BIT2+BIT1+BIT0+SILOAL,(R0)
                                                                              ;WRITE THE BITS
†B013476        011004                                    MOV     (R0),R4             ;READ BACK THE BITS
013500  001401                                    BEQ     1$                  ;BR IF NONE ARE SET.
013502  104002                                    ERROR   2                   ;*BITS WERE SET.
013504  012710  100000                    1$:     MOV     #TRDY,(R0)          ;ATTEMPT TO WRITE TRDY
013510  011004                                    MOV     (R0),R4             ;READ TRDY
013512  001401                                    BEQ     2$                  ;BR IF NOT SET
013514  104002                                    ERROR   2                   ;*
013516  012705  100000                    2$:     MOV     #TRDY,R5            ;SET EXPECTED BIT
013522  005077  166324                            CLR     @DZLPR              ;LOAD LINE 0
013526  052777  000001  166322                    BIS     #TCR0,@DZTCR        ;SET TCR BIT
013534  052710  000040                            BIS     #MSENAB,(R0)        ;
013540  052705  000040                            BIS     #MSENAB,R5          ;SET SCAN ENABLE
013544  005002                                    CLR     R2                  ;SET COUNTER TO ZERO
013546  011004                            3$:     MOV     (R0),R4             ;READ THE REGISTER
013550  020504                                    CMP     R5,R4               ;BIT SET?
013552  001404                                    BEQ     4$                  ;BR IF YES
013554  104414                                    DELAY                       ;STALL TIME
013556  005202                                    INC     R2                  ;UPDATE COUNTER
013560  001372                                    BNE     3$                  ;BR IF COUNTER NOT DONE.
013562  104002                                    ERROR   2                   ;*TRDY NOT SET!
013564                                    4$:
                                                  ;*********************** TEST 13 *****************************
                                                  ;*THIS TEST PERFORMS RESET TESTING AND
                                                  ;*TESTING OF READ ONLY AND WRITE ONLY BITS
                                                          ;* IN REGISTER DZCSR
                                                  ;*VERIFY THAT "TIE", "SILOEN", "RIE", "MSENAB", "MAINT"
                                                  ;*ARE THE ONLY R/W BITS IN THE DZCSR.
                                                  ;*THEN SET "DCLR" AND VERIFY THEY ARE CLEARED
                                          ;;*  TEST 13
                                          ;;*************************************************************
013564  000004                            TST13:  SCOPE
013566  012737  000013  001122                    MOV     #13,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
013574  012737  013650  001360                    MOV     #TST14,NEXT         ;POINT TO THE START OF THE NEXT TEST
013602  104413                                    DEVICE.CLR
013604  013700  002042                            MOV     DZCSR,R0            ;SET UP FOR ERROR MESSAGE
01361†B0        012710  177757                            MOV     #†C<DCLR>,(R0)    ;TRY TO WRITE
013614  012705  050150                            MOV     #TIE!SILOEN!RIE!MSENAB!MAINT,R5 ;MAKE EXPECTED
013620  011004                                    MOV     (R0),R4             ;ACTUAL
013622  020405                                    CMP     R4,R5               ;CMP EXPECTED VS ACTUAL
013624  001401                                    BEQ     1$                  ;YES
013626  104002                                    ERROR   2                   ;*NO
013630  012705  000020                    1$:     MOV     #DCLR,R5            ;EXPECTED...NOTE THAT DCLR REMAINS
                                                                              ;SET LONG ENOUGH TO READ IT....HOWEVER
                                                                              ;IF YOU EXAMINE THIS BIT IT SHOULD BE CLEAR.
013634  052710  000020                            BIS     #DCLR,(R0)          ;DEVICE MASTER RESET
013640  011004                                    MOV     (R0),R4             ;ACTUAL
013642  020405                                    CMP     R4,R5               ;CMP ACTUAL VS EXPECTED
013644  001401                                    BEQ     2$                  ;YES
```

```
013646  104002                            ERROR   2                  ;*NO
013650                            2$:
                                          ;*********************** TEST 14 ****************************
                                          ;*THIS TEST PERFORMS RESET TESTING AND
                                          ;*TESTING OF READ ONLY REGISTER DZRBUF
                                          ;*AND TESTING OF WRITE ONLY REGISTER DZLPR
                                  ;;*  TEST 14
                                  ;;*****************************************************↑B****
013650  000004                    TST14:  SCOPE
013652  012737  000014  001122            MOV     #14,$TSTNM         ;LOAD THE NUMBER OF THIS TEST
013660  012737  013740  001360            MOV     #TST15,NEXT        ;POINT TO THE START OF THE NEXT TEST
013666  104413                            DEVICE.CLR                 ;CLEAR DZ11
013670  013700  002046                    MOV     DZRBUF,R0                ;SET UP FOR ERROR MESSAGE
013674  012777  177777  166150            MOV     #-1,@DZLPR         ;TRY TO WRITE ALL 1'S
013702  011004                            MOV     (R0),R4            ;ACTUAL
013704  010405                            MOV     R4,R5              ;MAKE EXPECTED
013706  042705  104000                    BIC     #DVALID!BIT11,R5   ;DITTO
013712  020405                            CMP     R4,R5              ;CMP ACTUAL VS EXPECTED
013714  ↑B001401                                  BEQ      1$              ;IF YES,GO CONTINUE PROCESSING
013716  104002                            ERROR   2                  ;*ERROR- BIT PATTERN NOT CORRECT
013720  010403                    1$:     MOV     R4,R3              ;GET A COPY OF THE ACTUAL BIT PATTERN
013722  005103                            COM     R3                 ;GET THE LOGICAL INVERSE OF THE BIT PATTERN
013724  010377  166122                    MOV     R3,@DZLPR          ;TRY TO WRITE
013730  011004                            MOV     (R0),R4            ;ACTUAL
013732  020405                            CMP     R4,R5              ;CMP ACTUAL VS EXPECTED
013734  001401                            BEQ     2$                 ;IF YES, GET OUT OF THIS TEST
013736  104002                            ERROR   2                  ;*NO
013740                            2$:
                                          ;*********************** TEST 15 ****************************
                                          ;*THIS TEST PERFORMS RESET TESTING AND
                                          ;*TESTING OF READ ONLY REGISTER DZMSR
                                          ;*AND TESTING OF WRITE ONLY REGISTER DZTDR
                                  ;;*  TEST 15
                                  ;;*********************************************************
013740  000004                    TST15:  SCOPE
013742  012737  000015  001122            MOV     #15,$TSTNM         ;LOAD THE NUMBER OF THIS TEST
013750  012737  014024  001360            MOV     #TST16,NEXT        ;POINT TO THE START OF THE NEXT TEST
013756  104413                            DEVICE.CLR                 ;CLEAR DZ11
013760  013700  002062                    MOV     DZMSR,R0                 ;SET UP FOR ERROR MESSAGE
013764  012777  177777  166074            MOV     #-1,@DZTDR         ↑B;TRY TO WRITE ALL 1'S
013772  011004                            MOV     (R0),R4            ;ACTUAL
013774  010405                            MOV     R4,R5              ;MAKE EXPECTED
013776  020405                            CMP     R4,R5              ;CMP ACTUAL VS EXPECTED
014000  001401                            BEQ     1$                 ;IF YES,GO CONTINUE PROCESSING
014002  104002                            ERROR   2                  ;*ERROR- BIT PATTERN NOT CORRECT
014004  010403                    1$:     MOV     R4,R3              ;GET A COPY OF THE ACTUAL BIT PATTERN
014006  005103                            COM     R3                 ;GET THE LOGICAL INVERSE OF THE BIT PATTERN
014010  010377  166052                    MOV     R3,@DZTDR          ;TRY TO WRITE
014014  011004                            MOV     (R0),R4            ;ACTUAL
014016  020405                            CMP     R4,R5              ;CMP ACTUAL VS EXPECTED
014↑B020        001401                            BEQ      2$              ;IF YES, GET OUT OF THIS TEST
014022  104002                            ERROR   2                  ;*NO
014024                            2$:

                                          ;*********************** TEST 16 ****************************
                                          ;*VERIFY THAT IF WE ARE IN "STAGGERED" MODE
```

```
                                                ;*THAT SETTING "DTR" FOR A LINE WILL
                                                ;*BRING UP "RING" AND "CARRIER" FOR THE
                                                ;*ASSOCIATED LINE IN WHICH WE ARE STAGGERED!
                                                ;* LINE0 DTR= LINE1 RING AND CARRIER
                                                ;* LINE1 DTR= LINE0 RING AND CARRIER
                                                ;* LINE2 DTR= LINE3 RING AND CARRIER
                                                ;* LINE3 DTR= LINE 4 RING AND CARRIER
                                                ;*              ETC...

                                        ;;*  TEST 16
                                        ;;*****************************************************************
  014024  000004                        TST16:  SCOPE
  014026  012737  000016  001122                MOV     #16,$TSTNM              ;LOAD THE NUMBER OF THIS TEST
  014034  012737  014214  001360                MOV     #TST17,NEXT             ;POINT TO THE START OF THE NEXT TEST
  014042  012737  014114  001362                MOV     #1$,LOCK                ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
  014050  105737  001414                        TSTB    EIAFLG                  ;EIA OR 20MA?
↑B014054  100001                                BPL     10$                     ;BR IF EIA
  014056  104400                                ADVANCE                         ;EXIT TEST
  014060  013700  002062                10$:    MOV     DZMSR,R0                ;SET REGISTER
  014064  104413                                DEVICE.CLR                      ;INIT DZ11
  014066  005003                                CLR     R3                      ;ZERO LINE NUMBER
  014070  012702  000001                        MOV     #1,R2                   ;SET POINTER
  014074  005737  001370                        TST     MODE                    ;ARE WE IN STAGGERED MODE?
  014100  100405                                BMI     1$                      ;YES WE ARE!
  014102  013737  001360  001126                MOV     NEXT,$LPADR             ;LEAVE THIS TEST! NOT STAGGERED
  014110  000177  165012                        JMP     @$LPADR                 ;EXIT
  014114  130237  001364                1$:     BITB    R2,LINE                 ;TEST THIS LINE?
  014120  0010↑B04                                      BNE     3$                      ;YES
  014122  005203                        2$:     INC     R3                      ;LINE #
  014124  106302                                ASLB    R2                      ;GET NEXT LINE
  014126  103372                                BCC     1$                      ;KEEP TESTING
  014130  104400                                ADVANCE                         ;ADVANCE THIS TEST
  014132  010204                        3$:     MOV     R2,R4                   ;SAVE BINARY BIT FOR LINE #
  014134  032703  000001                        BIT     #BIT0,R3                ;GET STAGGERED COMPANION LINE
  014140  001402                                BEQ     4$                      ;BR IF LINE EVEN
  014142  006204                                ASR     R4                      ;ADJUST LINE
  014144  000401                                BR      5$
  014146  006304                        4$:     ASL     R4                      ;ADJUST LINE
  014150  005005                        5$:     CLR     R5                      ;SET EXPECTED
  014152  150405                                BISB    R4,R5                   ;
  014154  000305                                SWAB    R5                      ;
  014156  150405                                BISB    R4,R5
  014160  150277  165674                        BISB    R2,@HDZTCR              ;SET DTR
  014164  011004                                MOV     (R0),R4                 ;READ MSR REGISTER
  014166  020504                                CMP     R5,R4                   ;OK?
  014170  001401                                BEQ     6$                      ;YES
  014172  104002                                ERROR   2                       ;*ERROR IN RING OR CARRIER
  014174  140277  165660                6$:     BICB    R2,@HDZTCR              ;CLEAR DTR
  014200  011004                                MOV     (R0),R4                 ;READ MSR
  014202  001402                                BEQ     7$                      ;BR IF THEY CLEARED
  014204  005005                                CLR     R5                      ;SET EXPECTED TO 0
  014206  104002                                ERROR   2                       ;*BITS NOT CLEARED
  014210  104401                        7$:     SCOP1                           ;LOCK ON SIGNAL?
  0↑B14212        000743                                BR      2$                      ;CONTINUE TEST

                                                ;********************** TEST 17 ***************************
                                                ;*TEST TO VERIFY THAT IF IN "EXTERNAL"
                                                ;*MODE; SETTING DTR FOR SELECTED LINES
                                                ;*WILL BRING UP "CARRIER" AND "RING"
                                                ;*FOR THAT SAME LINE. NOTE: IF YOU HAVE
                                                ;*SELECTED MODE AS "EXTERNAL"; THE H325 TEST CONNECTER
                                                ;*MUST BE USED ON ALL SPECIFIED LINES.
                                                ;*LINES MAY BE SPE↑BCIFIED BY SWR03=1
                                                ;*AND SWR00=1 AT START TIME OR ALTERING
                                                ;*STATUS MAP.
                                        ;:*  TEST 17
                                        ;:*****************************************************************
014214  000004                          TST17:  SCOPE
014216  012737  000017  001122                  MOV     #17,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
014224  012737  014352  001360                  MOV     #TST20,NEXT     ;POINT TO THE START OF THE NEXT TEST
014232  012737  014266  001362                  MOV     #3$,LOCK        ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
014240  105737  001370                          TSTB    MODE            ;EXTERNAL?
014244  100401                                  BMI     2$              ;BR IF YES
014246  104400                          1$:     ADVANCE                 ;EXIT TEST
014250  105737  001414                  2$:     TSTB    EIAFLG          ;YOU BETTER BE IN
014254  100774                                  BMI     1$              ;EIA MODE FOR THIS TEST.
014256  013700  002062                          MOV     DZMSR,R0        ;SET REGISTER
014262  012702  000001                          MOV     #1,R2           ;SET LINE POINTER
014266  130237  001364                  3$:     BITB    R2,LINE         ;LINE SELECTED?
014272  001003                                  BNE     5$              ;BR IF YES
014274  106302                          4$:     ASLB    R2              ;NEXT LINE
014276  103373                                  BCC     3$              ;CONTINUE TEST
014300  104400                                  ADVANCE                 ;ADVANCE THIS TEST
014302  005005                          5$:     CLR     R5              ;SET EXPECTED
014304  150205                                  BISB    R2,R5
014306  000305                                  SWAB    R5              ↑B;
014310  150205                                  BISB    R2,R5           ;
014312  150277  165542                          BISB    R2,@HDZTCR      ;SET DTR
014316  104414                                  DELAY                   ;CABLE DELAY
014320  011004                                  MOV     (R0),R4         ;READ MSR
014322  020504                                  CMP     R5,R4           ;BITS OK?
014324  001401                                  BEQ     6$              ;BR IF YES
014326  104002                                  ERROR   2               ;CARRIER OR RING ERROR
014330  140277  165524                  6$:     BICB    R2,@HDZTCR      ;CLEAR DTR
014334  104414                                  DELAY                   ;CABLE DELAY
014336  011004                                  MOV     (R0),R4         ;READ MSR
014340  001402                                  BEQ     7$              ;BR IF BITS CLEARED
014342  005005                                  CLR     R5              ;CLEAR EXPECTED LOC.
014344  104002                                  ERROR   2               ;BITS ↑BNOT CLEARED.
014346  104401                          7$:     SCOP1                   ;LOCK ON LINE?
014350  000751                                  BR      4$              ;CONTINUE TEST

                                                ;*********************** TEST 20 ****************************
                                                ;* THIS TEST VERIFIES THAT TRDY IS SET WHEN A LINE
                                                ;* IS READY TO BE LOADED, AND THAT THE LINE SPECI-
                                                ;* FIED IN BITS 8-10 OF DZCSR CORRESPOND
                                                ;* TO THE LINE SELECTED IN DZTCR
                                        ;:*  TEST 20
                                        ;:*****************************************************************
014352  000004                          TST20:  SCOPE
014354  012737  000020  001122                  MOV     #20,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
014362  012737  014466  001360                  MOV     #TST21,NEXT     ;POINT TO THE START OF THE NEXT TEST
```

```
014370  104413                                  DEVICE.CLR                      ;ISSUE A "DEVICE CLEAR" (RESET)
014372  013700  002042                          MOV     DZCSR,R0                ;SET POINTER
014376  012705  100040                          MOV     #MSENAB!TRDY,R5         ;START THE EXPECTED LINE NUMBER AT 0
014402  012702  000001                          MOV     #1,R2                   ;USING R2 AS A BIT POINTER, POINT TO LINE 0
014406  130237  001364                  1$:     BITB    R2,LINE                 ;IS THIS LINE SELECTED?
0144↑B12        001420                          BEQ     5$
                                                                                        ;IF NO, SKIP THE STARTUP
014414  050277  165436                  2$:     BIS     R2,@DZTCR               ;SET THE GO BIT FOR THIS LINE
014420  052710  000040                          BIS     #MSENAB,(R0)            ;START THE SCANNER
014424  005004                                  CLR     R4                      ;SET FOR DELAY
014426  032710  100000                  3$:     BIT     #TRDY,(R0)              ;TX READY?
014432  001004                                  BNE     4$                      ;BR IF YES
014434  104414                                  DELAY                           ;DELAY
014436  005204                                  INC     R4                      ;COUNTER
014440  001372                                  BNE     3$                      ;BR IF <>0!
014442  104002                                  ERROR   2                       ;*TX NOT READY!
014444  011004                          4$:     MOV     (R0),R4                 ;GET THE LINE POINTED TO BY THE SCANNER
014446  020405                                  CMP     R4,R5                   ;IS↑B THE LINE NUMBER WHAT IT SHOULD BE?
014450  001401                                  BEQ     5$                      ;IF YES,GO WORK ON THE NEXT LINE
014452  104002                                  ERROR   2                       ;*LINE NUMBER DID NOT MATCH TCR BIT
014454  062705  000400                  5$:     ADD     #400,R5                 ;POINT TO THE NEXT EXPECTED LINE
014460  104413                                  DEVICE.CLR                      ;ISSUE A "DEVICE CLEAR" (RESET)
014462  106302                                  ASLB    R2                      ;POINT TO THE NEXT LINE.ARE ALL LINES TESTED?
014464  103350                                  BCC     1$                      ;IF NOT, GO DO THE NEXT LINE
014466                                  6$:
                                                ;************************ TEST 21 *****************************
                                                ;*TEST TO TRANSMIT ONE CHAR AND
                                                ;*RECEIVE ONE CHAR ON ONE LINE
                                                ;*AT A TIME. THE CHAR IS "252" AND
                                                ;*ALL SELECTED LINES WILL BE TURNED ON
                                                ;*ONE AT A TIME. THIS IS THE FIRST TIME ANY
                                                ;*DATA IS CHECKED IN THE RECEIVER.
                                                ;*USING SWITCH NINE WITH THIS TEST CREATES A TIGHT SCOPE LOOP
                                                ;*WHICH TRANSMITS A STEADY STREAM OF CHARACTERS.
                                        ;:*  TEST 21
                                        ;:*************************************************************
014466  000004                          TST21:  SCOPE
014470  012737  000021  001122                  MOV     #21,$TSTNM              ;LOAD THE NUMBER OF THIS TEST
014476  012737  015002  001360                  MOV     #TST22,NEXT             ;POINT TO ↑BTHE START OF THE NEXT TEST
014504  012737  014760  001362                  MOV     #16$,LOCK               ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
014512  104417                                  DCLASM                          ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
014514  013701  001366                          MOV     PAR,R1                  ;PICK UP PARAMETERS
014520  012702  000001                          MOV     #1,R2                   ;PICK UP INIT POINTER
014524  030237  001364                  1$:     BIT     R2,LINE                 ;SHOULD THIS LINE BE SET UP ?
014530  001402                                  BEQ     2$                      ;NO
014532  010177  165314                          MOV     R1,@DZLPR               ;SET UP LINE PARAMETERS
014536  005201                          2$:     INC     R1                      ;POSITION POINTER TO THE NEXT LINE
014540  106302                                  ASLB    R2                      ;GOT 'EM ALL ?
2↑B650          014542  103370                  BCC     1$
                                                                                        ;IF NO, GO SET UP THE NEXT LINE
014544  005037  001372                          CLR     SAVLIN                  ;CLEAR LINE # INDICATOR
014550  012702  000001                          MOV     #1,R2                   ;LINE POINTER
014554  052777  000040  165260                  BIS     #MSENAB,@DZCSR          ;START SCANNER
014562  030237  001364                  3$:     BIT     R2,LINE                 ;VALID LINE ?
014566  001462                                  BEQ     14$                     ;NO SET UP NEXT LINE
014570  010277  165262                          MOV     R2,@DZTCR               ;SET TCR BIT
014574  032777  000200  165240          4$:     BIT     #RDONE,@DZCSR           ;IS REC DONE = 0 ?
014602  001401                                  BEQ     5$                      ;IF YES, ALLOW TIME FOR TRDY TO SET
```

```
014604  104020                          ERROR   20              ;*REC DONE SHOULD = 0
014606  005005                  5$:     CLR     R5
014610  032777  100000  165224  6$:     BIT     #TRDY,@DZCSR
014616  001004                          BNE     7$
014620  104414                          DELAY
014622  105205                          INCB    R5
014624  001371                          BNE     6$
014626  104003                          ERROR   3               ;*TRDY FAILED TO SET!
014630  112777  000252  165230  7$:     MOVB    #252,@DZTDR     ;LOAD CHARACTER
014636  013705  001372                  MOV     SAVLIN,R5       ;MAKE EXPECTED LINE #
014642  105737  001371                  TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MOD↑BE?
014646  001406                          BEQ     10$             ;IF NOT, SKIP STAGGERED SETUP

                                        ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER

014650  006205                          ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
014652  103402                          BCS     8$              ;IF IT IS SET, GO CLEAR IT
014654  000261                          SEC                     ;IF IT IS CLEAR SET IT HERE
014656  000401                          BR      9$              ;SKIP THE CLEARING
014660  000241                  8$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
014662  006105                  9$:     ROL     R5              ;GET THE NEW BIT BACK INTO R5
014664  000305                  10$:    SWAB    R5              ;MOVE THE LINE NUMBER TO THE UPPER BYTE
014666  152705  0002↑B52                        BISB    #252,R5         ;ADD CHARACTER
014672  052705  100000                  BIS     #DVALID,R5      ;ADD DATA VALID
014676  005003                          CLR     R3
014700  032777  000200  165134  11$:    BIT     #RDONE,@DZCSR
014706  001004                          BNE     12$
014710  104414                          DELAY
014712  105203                          INCB    R3
014714  001371                          BNE     11$
014716  104004                          ERROR   4               ;*RDONE FAILED TO SET!
014720  017704  165122          12$:    MOV     @DZRBUF,R4      ;LOAD THE VALUE ACTUALLY RECEIVED
014724  020405                          CMP     R4,R5           ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
014726  001401                          BEQ     13$             ;IF YES, GO DO THE NEXT LINE
014730  104006                          ERROR   6               ;*NO DATA/CONTENTS DID NOT COMPARE
014732  104401                  13$:    SCOP1                   ;CHECK TO SEE IF SWITCH NINE IS SET
014734  040277  165116          14$:    BIC     R2,@DZTCR       ;CLEAR TCR BIT FOR THAT LINE.
014740  005237  001372          15$:    INC     SAVLIN          ;INC EXPECTED LINE
014744  013700  001372                  MOV     SAVLIN,R0       ;SET UP CHARACTER OFFSET
014750  006300                          ASL     R0              ;MAKE THE OFFSET A POWER OF TWO
014752  106302                          ASLB    R2              ;SHIFT THE LINE POINTER. ARE WE ALL DONE?
014754  103302                          BCC     3$              ;IF NO, GO AROUND AGAIN FOR NEXT LINE
014756  104400                          ADVANCE                 ;GO TO NEXT TEST

                                ;TIGHT SCOPE LOOP FOR THIS TEST. L↑BOOP TRANSMITS CHARACTERS ONLY

014760  032777  100000  165054  16$:    BIT     #TRDY,@DZCSR    ;IS TRANSMITTER READY?
014766  001774                          BEQ     16$             ;IF NOT, WAIT FOR IT
014770  112777  000252  165070          MOVB    #252,@DZTDR     ;LOAD THE CHARACTER
014776  104401                          SCOP1                   ;LOOP AGIN IF SW09=1
015000  000755                          BR      14$             ;OTHERWISE, GO PICK UP THE TEST NORMALLY


                                        ;************************ TEST 22 ****************************
                                        ;* THIS TEST PROVES THAT THE TRANSMITTER TRANSMITS
                                        ;*CHARACTERS (FLAG MODE)AND THE RECEIVER RECEIVES (FLAG MODE)
                                        ;*(ONE LINE AT A TIME   BA↑BSED UPON VALID LINES)
```

```
                                                  ;*THIS IS THE FIRST TIME THAT ALL DATA IS CHECKED
                                        ;;*  TEST 22
                                        ;;*****************************************************************
015002  000004                          TST22:  SCOPE
015004  012737  000022  001122                  MOV     #22,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
015012  012737  015330  001360                  MOV     #TST23,NEXT     ;POINT TO THE START OF THE NEXT TEST
015020  012737  015134  001362                  MOV     #4$,LOCK        ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
015026  104417                                  DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
015030  013701  001366                          MOV     PAR,R1          ;PICK UP PARAMETERS
015034  012702  000001                          MOV     #1,R2           ;PICK UP INIT POINTER
015040  030237  001364                  1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
015044  001402                                  BEQ     2$              ;NO
015046  010177  165000                          MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
015052  005201                          2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
015054  106302                                  ASLB    R2              ;GOT 'EM ALL ?
015056  103370                                  BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
015060  005037  001372                          CLR     SAVLIN          ;CLEAR LI↑BNE # INDICATOR
015064  012700  001422                          MOV     #TD0,R0         ;POINT TO THE DATA AREA
015070  005020                                  CLR     (R0)+           ;CLEAR A DATA WORD
015072  022700  001462                          CMP     #STOP,R0        ;FINISHED ?
015076  001374                                  BNE     .-6             ;NO
015100  005000                                  CLR     R0              ;CLEAR OFFSET
015102  013737  002046  001400                  MOV     DZRBUF,REGIST   ;SAVE FOR ERROR MSG
015110  012702  000001                          MOV     #1,R2           ;LINE POINTER
015114  052777  000040  164720                  BIS     #MSENAB,@DZCSR  ;START SCANNER
015122  030237  001364                  3$:     BIT     R2,LINE         ;VALID LINE ?
015126  001465                                  BEQ     14$             ;NO SET UP NEXT LINE
015130  010277  164722                          MOV     R2,@DZTC↑BR     ;SET TCR BIT
015134  032777  000200  164700          4$:     BIT     #RDONE,@DZCSR   ;IS REC DONE = 0 ?
015142  001401                                  BEQ     5$              ;IF YES, ALLOW TIME FOR TRDY TO SET
015144  104020                                  ERROR   20              ;*REC DONE SHOULD = 0
015146  005005                          5$:     CLR     R5
015150  032777  100000  164664          6$:     BIT     #TRDY,@DZCSR
015156  001004                                  BNE     7$
015160  104414                                  DELAY
015162  105205                                  INCB    R5
015164  001371                                  BNE     6$
015166  104003                                  ERROR   3               ;*TRDY FAILED TO SET!
015170  116077  001422  164670          7$:     MOVB    TD0(R0),@DZTDR  ;LOAD CHARACTER
015176  013705  001372                          MOV     SAVLIN,R5       ;MAKE EXPECTED LINE #
015202  105737  001371                          TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
015206  001406                                  BEQ     10$             ;IF NOT, SKIP STAGGERED SETUP

                                                ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER

015210  006205                                  ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
015212  103402                                  BCS     8$              ;IF IT IS SET, GO CLEAR IT
015214  000261                                  SEC                     ;IF IT IS CLEAR SET IT HERE
015216  000401                                  BR      9$              ;SKIP THE CLEARING
015220  000241                          8$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
015222  006105                          9$:     ROL     R5              ;GET THE NEW BIT BACK INTO R5
015224  000305                          10$:    SWAB↑B  R5              ;MOVE THE LINE NUMBER TO THE UPPER BYTE
015226  156005  001422                          BISB    TD0(R0),R5      ;ADD CHARACTER
015232  052705  100000                          BIS     #DVALID,R5      ;ADD DATA VALID
015236  005003                                  CLR     R3
015240  032777  000200  164574          11$:    BIT     #RDONE,@DZCSR
```

```
015246  001004                          BNE     12$
015250  104414                          DELAY
015252  005204                          INC     R4
015254  001371                          BNE     11$
015256  104004                          ERROR   4                  ;*RDONE FAILED TO SET!
015260  017704  164562          12$:    M↑BOV   @DZRBUF,R4         ;LOAD THE VALUE ACTUALLY RECEIVED
015264  020405                          CMP     R4,R5              ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
015266  001401                          BEQ     13$                ;IF YES, GO DO THE NEXT LINE
015270  104006                          ERROR   6                  ;*NO DATA/CONTENTS DID NOT COMPARE
015272  104401                  13$:    SCOP1                      ;CHECK TO SEE IF SWITCH NINE IS SET
015274  105260  001422                  INCB    TDO(RO)            ;INCREMENT BINARY PATTERN FOR THIS LINE
015300  001315                          BNE     4$                 ;GO 'ROUND AGAIN FOR NEXT CHARACTER
015302  040277  164550          14$:    BIC     R2,@DZTCR        . ;CLEAR TCR BIT FOR THAT LINE.
015306  005237  001372          15$:    INC     SAVLIN             ;INC EXPECTED LINE
015312  013700  001372                  MOV     SAVLIN,RO          ;SET UP CHARACTER OFFSET
015316  006300                          ASL     RO                 ;MAKE THE OFFSET A POWER OF TWO
015320  106302                          ASLB    R2                 ;SHIFT THE LINE POINTER. ARE WE ALL DONE?
015322  103277                          BCC     3$                 ;IF NO, GO AROUND AGAIN FOR NEXT LINE
015324  005037  001362                  CLR     LOCK               ;MAKE SURE LOCK IS CLEAR FOR NEXT TEST


                                        ;************************ TEST 23 ****************************
                                        ;*THIS TEST WILL PROVE THAT:
                                        ;* 1) THE TRANSMITTER "BREAK BIT" WORKS
                                        ;* 2) THE RECEIVER CAN FLAG "FRAMING ERRORS"
                                        ;* 3) THE RECEIVER CAN↑B FLAG "PARITY ERRORS"
                                        ;*ONLY ONE LINE AT A TIME WILL BE EXERCISED.
                                        ;*THIS TEST WILL NOT BE  EXERCISED UNLESS
                                        ;*CONNECTED BY EXTERNAL PLUG.
                                ;;*  TEST 23
                                ;;*********************************************************
015330  000004                  TST23:  SCOPE
015332  012737  000023  001122          MOV     #23,$TSTNM         ;LOAD THE NUMBER OF THIS TEST
015340  012737  015606  001360          MOV     #TST24,NEXT        ;POINT TO THE START OF THE NEXT TEST
015346  012737  015444  001362          MOV     #3$,LOCK           ;SET FOR LOOP
015354  005737  001370                  TST     MODE               ;ARE WE RUNNING IN INTERNAL MODE?
015↑B360        001510                          BEQ     12$             ;IF SO, SKIP THIS TEST
015362  104417                          DCLASM                     ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
015364  013701  001366                  MOV     PAR,R1             ;PICK UP PARAMETERS
015370  052701  000300                  BIS     #ODDPAR!PARITY,R1  ;FORCE ODD PARITY
015374  012700  000001                  MOV     #1,RO              ;PICK UP INIT POINTER
015400  030037  001364          1$:     BIT     RO,LINE            ;SHOULD THIS LINE BE SET UP ?
015404  001402                          BEQ     2$                 ;IF NOT DON'T SET IT UP
015406  010177  164440                  MOV     R1,@DZLPR          ;OTHERWISE, SET UP LINE PARAMETERS
015412  005201                  2$:     INC     R1
015414  106300                          ASLB    RO                 ;GOT 'EM ALL ?
015416  103370                          BCC     1$                 ;NO
015420  005037  001372                  CLR     SAVLIN             ;CLEAR LINE #
015424  012702  000001                  MOV     #1,R2              ;LINE POINTER
015430  052777  000040  164404          BIS     #MSENAB,@DZCSR     ;SET MASTER SCAN ENABLE
015436  013737  002046  001400          MOV     DZRBUF,REGIST      ;SAVE FOR ERRR MESSAGE
015444  030237  001364          3$:     BIT     R2,LINE
015450  001443                          BEQ     10$
015452  010277  164400                  MOV     R2,@DZTCR          ;SET TCR BIT
015456  110277  164406                  MOVB    R2,@HDZTDR         ;SET BREAK BIT
015462  112777  000377  164376  4$:     MOVB    #377,@DZTDR        ;LOAD CHARACTER
```

```
015470  013705  001372                    MOV     SAVLIN,R5               ;MAKE EXPECTED DATA
015474  105737  001371                    TSTB    MODE+1                  ;IS THIS TEST IN STAGGERED MODE?
015500  001406                            BEQ     7$                      ;IF NOT, SKIP STAGGERED SETUP

                                          ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER

015502  006205                            ASR     R5                      ;GET THE LAST BIT INTO THE CARRY BIT
015504  103402                            BCS     5$                      ;IF IT IS SET, GO CLEAR IT
015506  000261                            SEC                             ;IF IT IS CLEAR SET IT HERE
015510  000401                            BR      6$                      ;SKIP THE CLEARING
015512  000241                    5$:     CLC                             ;CLE↑BAR THE CARRY BIT (INVERSION OF LINE PARITY)
015514  006105                    6$:     ROL     R5                      ;GET THE NEW BIT BACK INTO R5
015516  000305                    7$:     SWAB    R5                      ;PUT LINE NUMBER IN UPPER BYTE
015520  052705  130000                    BIS     #DVALID!PARER!FRMERR,R5 ;ADD EXPECTED
015524  005004                            CLR     R4
015526  032777  000200  164306    8$:     BIT     #RDONE,@DZCSR
015534  001004                            BNE     9$
015536  104414                            DELAY
015540  005204                            INC     R4
015542  001371                            BNE     8$
015544  104004                            ERROR   4                       ;*RDONE FAILED TO SET!
015546  017704  164274            9$:     MOV     @DZRBUF,R4              ;ACTUAL
015552  020405                            CMP     R4,R5                   ;CMP ACTUAL VS EXPECTED. DO THEY MATCH?
015554  001401                            BEQ     10$                     ;IF YES, GO CLEAN UP
015556  104006                            ERROR   6                       ;*DATA/CONTENTS FAILED TO COMPARE
015560  105077  164304            10$:    CLRB    @HDZTDR                 ;CLEAR BREAK BITS
015564  104401                            SCOP1                           ;LOOP?
015566  005237  001372            11$:    INC     SAVLIN                  ;INC LINE #
015572  040277  164260                    BIC     R2,@DZTCR               ;CLEAR TCR BIT
015576  106302                            ASLB    R2
015600  103321                            BCC     3$
015602  005037  001362            12$:    CLR     LOCK                    ;MAKE SURE LOCK IS CLEAR FOR NEXT TEST
                                          ;*********************** TEST 24 ******************************
                                          ;* THIS TEST VERIFIES THAT THE DE↑BVICE DOES NOT INTERRUPT
                                          ;*WHILE THE PROCESSOR STATUS IS SET EXACTLY
                                          ;*TO WHAT THE DZ11 PRIORITY IS SET TO.
                                          ;*DEFAULT PRIORITY IS AT  5 (240).
                                  ;:*  TEST 24
                                  ;;*************************************************************
015606  000004                    TST24:  SCOPE
015610  012737  000024  001122            MOV     #24,$TSTNM              ;LOAD THE NUMBER OF THIS TEST
015616  012737  016114  001360            MOV     #TST25,NEXT             ;POINT TO THE START OF THE NEXT TEST
015624  104417                            DCLASM                          ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
015626  013701  001366                    MOV     PAR,R1                  ;PICK UP PARAMETERS
↑B015632                012702  000001    MOV     #1,R2                   ;PICK UP INIT POINTER
015636  030237  001364            1$:     BIT     R2,LINE                 ;SHOULD THIS LINE BE SET UP ?
015642  001402                            BEQ     2$                      ;NO
015644  010177  164202                    MOV     R1,@DZLPR               ;SET UP LINE PARAMETERS
015650  005201                    2$:     INC     R1                      ;POSITION POINTER TO THE NEXT LINE
015652  106302                            ASLB    R2                      ;GOT 'EM ALL ?
015654  103370                            BCC     1$                      ;IF NO, GO SET UP THE NEXT LINE
015656  005037  001372                    CLR     SAVLIN                  ;CLEAR LINE # INDICATOR
015662  106437  026216                    MTPS    @#DZPRT                 ;SET CPU STATUS TO DZ11 PRIO,
015666  113777  001364  164162            MOVB    LINE,@DZTCR             ;ENABLE THE VALID LINES
015674                            3$:
015674  012777  015762  164174            MOV     #6$,@DZTIV              ;SET UP THE TRANSMITTER INTERRUPT VECTOR
```

```
015702  012777  015770  164162             MOV     #7$,@DZRIV      ;SET UP THE RECEIVER INTERRUPT VECTOR
015710  013777  026216  164156             MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
015716  013777  026216  164154             MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
015724  052777  040040  164110             BIS     #TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
015732  00↑B5005                                   CLR     R5
015734  032777  100000  164100    4$:      BIT     #TRDY,@DZCSR
015742  001403                             BEQ     5$
015744  000240                             NOP
015746  000240                             NOP
015750  000411                             BR      8$
015752  104414                    5$:      DELAY
015754  005205                             INC     R5
015756  001366                             BNE     4$
015760  104003                             ERROR   3               ;*TRDY NOT SET!
015762  104010                    6$:      ERROR   10              ;*TRANSMITTER SHOULD NOT INTERRUPT
015764  022626                             CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
015766  000402                             BR      8$              ;CONTINUE TEST
015770  104012                    7$:      ERROR   12              ;*RECEIVER SHOULD NOT INTERRUPT
015772  022626                             CMP     (SP)+,(SP)+     ;POP FOR ↑BFAKE RTI
015774  042777  040000  164040    8$:      BIC     #TIE,@DZCSR     ;RESET TRANSMITTER INTERRUPT ENABLE
016002  113777  001422  164056             MOVB    TD0,@DZTDR      ;PUT ANY RANDOM CHARACTER IN TRANSMITTER BUFFER
016010  012777  016100  164060             MOV     #11$,@DZTIV     ;SET UP THE TRANSMITTER INTERRUPT VECTOR
016016  012777  016106  164046             MOV     #12$,@DZRIV     ;SET UP THE RECEIVER INTERRUPT VECTOR
016024  013777  026216  164042             MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
016032  013777  026216  164040             MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
016040  052777  000140  163774             BIS     #RIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
016046  005005                             CLR     R5
016050  032777  000200  163764    9$:      BIT     #RDONE,@DZCSR
016056  001403                             BEQ     10$
016060  000240                             NOP
016062  000240                             NOP
016064  000412                             BR      13$
016066  104414                    10$:     DELAY
016070  005205                             INC     R5
016072  001366                             BNE     9$
016074  104004                             ERROR   4               ;*NO RX DONE! (NOT SET)
016076  000405                             BR      13$             ;CONTINUE TEST
016100  104010                    11$:     ERROR   10              ;*TRANSMITTER SHOULD NOT INTERRUPT
016102  022626                             CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
016104  000402                             BR      13$             ;CONT TEST
016106  104012                    12$:     ERROR   12              ;*R↑BECEIVER SHOULD NOT INTERRUPT
016110  022626                             CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
016112                            13$:
016112  104413                             DEVICE.CLR              ;ISSUE DEVICE CLEAR (RESET)
                                           ;********************** TEST 25 ******************************
                                           ;* THIS TEST VERIFIES THAT THE DEVICE DOES INTERRUPT
                                           ;*WHILE THE PROCESSOR STATUS IS SET TO EXACTLY
                                           ;*ONE LEVEL LOWER THAN THE DZ11. DZ11 PRIORITY
                                           ;*DEFAULT TO LEVEL 5 MINUS ONE LEVEL IS LEVEL 4.
                                  ;:*  TEST 25
                                  ;;*************************************************************
016114  000004                    TST25:   SCOP↑BE
016116  012737  000025  001122             MOV     #25,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
016124  012737  016450  001360             MOV     #TST26,NEXT     ;POINT TO THE START OF THE NEXT TEST
016132  104417                             DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
016134  013701  001366                     MOV     PAR,R1          ;PICK UP PARAMETERS
```

```
016140  012702  000001                  MOV     #1,R2                   ;PICK UP INIT POINTER
016144  030237  001364          1$:     BIT     R2,LINE                 ;SHOULD THIS LINE BE SET UP ?
016150  001402                          BEQ     2$                      ;NO
016152  010177  163674                  MOV     R1,@DZLPR               ;SET UP LINE PARAMETERS
016156  005201                  2$:     INC     R1                      ;POSITION POINTER TO THE NEXT LINE
016160  106302                          ASLB    R2                      ;GOT 'EM ALL ?
016162  103370                          BCC     1$                      ;IF NO, GO SET UP THE NEXT LINE
016164  005037  001372                  CLR     SAVLIN                  ;CLEAR LINE # INDICATOR
016170  106437  026216                  MTPS    @#DZPRT                 ;SET CPU STATUS TO DZ11 PRIO.
016174  106437  026220                  MTPS    @#LESS1                 ;MAKE CPU ONE LEVEL LOWER THAN DZ11
016200  113777  001364  163650          MOVB    LINE,@DZTCR             ;ENABLE THE VALID LINES
016206                          3$:
016206  012777  016276  163662          MOV     #6$,@DZTIV              ;S↑BET UP THE TRANSMITTER INTERRUPT VECTOR
016214  012777  016320  163650          MOV     #7$,@DZRIV              ;SET UP THE RECEIVER INTERRUPT VECTOR
016222  013777  026216  163644          MOV     DZPRT,@DZRIS            ;SET THE INTERRUPT VECTOR STATUS
016230  013777  026216  163642          MOV     DZPRT,@DZTIS            ;SET TRANSMITTER INTERRUPT PRIORITY
016236  052777  040040  163576          BIS     #TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
016244  005005                          CLR     R5
016246  032777  100000  163566  4$:     BIT     #TRDY,@DZCSR
016254  001404                          BEQ     5$
016256  000240                          NOP
016260  000240                          NOP
016262  104007                          ERROR   7                       ;*TRANSMITTER FAILED TO INTERRUPT
↑B016264        000417                          BR      8$
016266  104414                  5$:     DELAY
016270  005205                          INC     R5
016272  001365                          BNE     4$
016274  104003                          ERROR   3                       ;*TRDY NOT SET!
016276  022626                  6$:     POP2SP                          ;REMOVE THE INTERRUPT FROM THE STACK
016300  042777  040000  163534          BIC     #TIE,@DZCSR             ;DON'T LET ANY MORE INTERRUPTS OCCUR
016306  106437  026216                  MTPS    @#DZPRT                 ;SET CPU STATUS TO DZ11 PRIORITY
016312  106437  026220                  MTPS    @#LESS1                 ;MAKE CPU ONE LEVEL LOWER THAN DZ11
016316  000402                          BR      8$                      ;RETURN TO THE NORMAL FLOW
016320  104012                  7$:     ERROR   12                      ;*RECEIVER SHOULD NOT INTERRUPT
016322  022626                          CMP     (SP)+,(SP)+             ;POP FOR FAKE RTI
016324  042777  040000  163510  8$:     BIC     #TIE,@DZCSR             ;RESET TRANSMITTER INTERRUPT ENABLE
016332  113777  001422  163526          MOVB    TDO,@DZTDR              ;PUT ANY RANDOM CHARACTER IN TRANSMITTER BUFFER
016340  012777  016432  163530          MOV     #11$,@DZTIV             ;SET UP THE TRANSMITTER INTERRUPT VECTOR
016346  012777  016440  163516          MOV     #12$,@DZRIV             ;SET UP THE RECEIVER INTERRUPT VECTOR
016354  013777  026216  163512          MOV     DZPRT,@DZRIS            ;SET THE INTERRUPT VECTOR STATUS
016362  013777  026216  163510          MOV     DZPRT,@DZTIS            ;SET TRANSMITTER INTERRUPT PRIORITY
016370  052777  000140  163444          BIS     #R↑BIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
016376  005005                          CLR     R5
016400  032777  000200  163434  9$:     BIT     #RDONE,@DZCSR
016406  001404                          BEQ     10$
016410  000240                          NOP
016412  000240                          NOP
016414  104011                          ERROR   11                      ;*RECEIVER FAILED TO INTERRUPT
016416  000413                          BR      13$
016420  104414                  10$:    DELAY
016422  005205                          INC     R5
016424  001365                          BNE     9$
016426  104004                          ERROR   4                       ;*NO RX DONE! (NOT SET)
016430  000406                          BR      13$                     ;CONTINUE TEST
016432  104010                  11$:    ERROR   10                      ;*TRANSMITTER SHOULD NOT INTERRUPT
016434  022626                          CMP     (SP)+,(SP)+             ;POP ↑BFOR FAKE RTI
```

```
016436  000403                          BR      13$             ;CONT TEST
016440  022626                  12$:    POP2SP                  ;REMOVE THE INTERRUPT FROM THE STACK
016442  005077  163374                  CLR     @DZCSR          ;DON'T ALLOW ANY MORE INTERRUPTS
016446                          13$:
016446  104413                          DEVICE.CLR              ;ISSUE DEVICE CLEAR (RESET)

                                        ;********************* TEST 26 ***************************
                                        ;*THIS TEST VERIFIES THAT THE RECEIVER WILL
                                        ;*INTERRUPT BEFORE THE TRANSMITTER EVEN
                                        ;*THOUGH THE TRANSMITTER WAS ENABLED
                                        ;*FIRST.  SET PS TO LEVEL 7;
                                        ;*GET RDONE AND TRDY TO SET;
                                        ;*SET TX IE AND RX IE;
                                        ;*CLEAR PS AND EXPECT RX TO INTERRUPT FIRST
                                ;;*  TEST 26
                                ;;*****************************************************************
016450  000004                  TST26:  SCOPE
016452  012737  000026  001122          MOV     #26,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
016460  012737  017102  001360          MOV     #TST27,NEXT     ;POINT TO THE START OF THE NEXT TEST
016466  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
01647↑B0        013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
016474  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
016500  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
016504  001402                          BEQ     2$              ;NO
016506  010177  163340                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
016512  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
016514  106302                          ASLB    R2              ;GOT 'EM ALL ?
016516  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
016520  005037  001372                  CLR     SAVLIN          ;CLEAR LINE # INDICATOR
016524  012777  016754  163340          MOV     #8$,@DZRIV      ;SETUP INTERRUPT STUFF
016532  0↑B13777        026216  163334          MOV     DZPRT,@DZRIS    ;
016540  012777  017044  163330          MOV     #12$,@DZTIV     ;
016546  013777  026216  163324          MOV     DZPRT,@DZTIS    ;
016554  052777  000040  163260          BIS     #MSENAB,@DZCSR
016562  012702  000001                  MOV     #1,R2           ;LINE POINTER
016566  030237  001364          3$:     BIT     R2,LINE         ;VALID LINE ?
016572  001004                          BNE     4$
016574  005237  001372                  INC     SAVLIN
016600  106302                          ASLB    R2
016602  000771                          BR      3$
016604  106427  000340          4$:     MTPS    #PR7
016610  000240                          NOP
016612  000240                          NOP
016614  110277  163236                  MOVB    R2,@DZTCR       ;SET TCR BIT
016620  005777  163222                  TST     @DZRBUF         ;VALID DATA?
016624  100001                          BPL     .+4             ;IT BETTER NOT BE SET
016626  104017                          ERROR   17              ;DATA VALID SHOULD NOT BE SET
016630  105777  163206          5$:     TSTB    @DZCSR          ;RECEIVER DONE ?
016634  100001                          BPL     .+4
016636  104020                          ERROR   20              ;RECEIVER DONE BIT SHOULD NOT BE SET
016640  005005                          CLR     R5
016642  005004                          CLR     R4
016644  005777  163172          99$:    TST     @DZCSR          ;WAIT FOR TRDY
016650  100404                          BMI     100$            ;BR IF READY
016652  104414                          DELAY                   ;STALL TIME
016654  005204                          INC     R4              ;
```

```
016656  001372                          BNE     99$
016660  104003                          ERROR   3                       ;TRDY FAILED TO SET
016662  105077  163200          100$:   CLRB    @DZTDR
016666  005004                          CLR     R4
016670  032777  000200  163144  6$:     BIT     #RDONE,@DZCSR
016676  001004                          BNE     7$
016700  104414                          DELAY
016702  005204                          INC     R4
016704  001371                          BNE     6$
016706  104004                          ERROR   4                       ;*RDONE FAILED TO SET!
016710  005777  163126          7$:     TST     @DZCSR                  ;TRANS DONE BIT = 1 ?
016714  100401                          BMI     .+4                     ;YES
016716  104003  ↑B                      ERROR   3                       ;*NO   TRANS DONE FAILED TO SET
                                        ;NOW THAT BOTH TRANSMITTER AND RECEIVER DONE BIT =1
                                        ;SET INTERRUPT ENABLES AND WATCH THE FUR FLY
016720  052777  040000  163114          BIS     #TIE,@DZCSR
016726  052777  000100  163106          BIS     #RIE,@DZCSR
016734  106427  000000                  MTPS    #0
016740  000240                          NOP
016742  000240                          NOP
016744  104007                          ERROR   7                       ;*TRANSMITTER FAILED TO INTERRUPT
016746  104011                          ERROR   11                      ;*RECEIVER FAILED TO INTERRUPT
                                                                        ;CHECK BR LEVEL
016750  000137  017050                  JMP     13$      ;GET OUT

                                        ;RECEIVER INTERRUPT ROUTINE
016754  017704  163066          8$:     MOV     @DZRBUF,R4                      ;ACTUAL
016760  010403                          MOV     R4,R3
016762  000303                          SWAB    R3
016764  042703  177770                  BIC     #↑C<7>,R3               ;STRIP JUNK
016770  105737  001371                  TSTB    MODE+1                  ;IS THIS TEST IN STAGGERED MODE?
016774  001406                          BEQ     11$                     ;IF NOT, SKIP STAGGERED SETUP

                                        ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER

016776  006203                          ASR     R3                      ;GET THE LAST BIT INTO THE CARRY BIT
017000  103402                          BCS     9$                      ;IF IT IS SET, GO CLEAR IT
017002  000261                          SEC                             ;IF IT IS CLEAR SET IT HERE
017004  000401                          BR      10$                     ;SKIP THE CLEARING
↑B              017006  000241          9$:     CLC                             ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
017010  006103                  10$:    ROL     R3                      ;GET THE NEW BIT BACK INTO R3
017012  020337  001372          11$:    CMP     R3,SAVLIN               ;IS THIS A VALID LINE
017016  001401                          BEQ     .+4                     ;YES
017020  104015                          ERROR   15                      ;*INVALID LINE
017022  042704  177400                  BIC     #↑C<377>,R4             ;STRIP JUNK
017026  120504                          CMPB    R5,R4                   ;DATA COMPARE ?
017030  001401                          BEQ     .+4                     ;YES
017032  104005                          ERROR   5                       ;*DATA DOES NOT COMPARE
017034  040277  163016                  BIC     R2,@DZTCR               ;CLEAR TCR BIT
017040  022626                          POP2SP                          ;REMOVE HE INTERRUPT VECTOR F↑BROM THE STACK
017042  000402                          BR      13$                     ;GO GET OUT OF INTERRUPT MODE
                                        ;TRANSMITTER INTERRUPT SVC ROUTINE
017044  104011                  12$:    ERROR   11                      ;THE RECEIVER INTERRUPT FAILED
                                                                        ;TO OVERRIDE THE TRANSMITTER
017046  022626                          POP2SP                          ;REMOVE THE INTERRUPT VECTOR FROM THE STACK
017050  042777  040100  162764  13$:    BIC     #TIE!RIE,@DZCSR ;CLEAR INTERRUPT ENABLES
```

```
017056  013777  002074  163006            MOV     DZRIS,@DZRIV      ;RESTORE TRAPCATCHER
017064  005077  163004                    CLR     @DZRIS
017070  013777  002100  163000            MOV     DZTIS,@DZTIV
017076  005077  162776                    CLR     @DZTIS
                                          ;********************** TEST 27 *****************************
                                          ;*THIS TEST VERIFIES OVERRUN AND SILO ALARM
                                          ;*ONE LINE AT A TIME  - BASED UPON VALID LINES
                                          ;*AS EACH OF THE FIRST 16 CHARS ARE SENT; SILO ALARM IS
                                          ;*TESTED TO BE CLEARED. ON THE 16TH CHAR THE PROGRAM THEN
                                          ;*EXPECTS SILO ALARM TO SET. THEN THE ENTIRE
                                          ;*SILO IS FILLED AND AN OVERRUN IS EXPECTED ON THE 65TH
                                          ;*CHAR PULLED OUT OUT THE SILO↑B.
                                          ;*USING SWITCH NINE FOR THIS TEST SENDS 20. CHARACTERS
                                          ;*ON DZ LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
                                          ;*USED TO SCOPE SILO ALARM PULSES, ETC.
                                  ;:*  TEST 27
                                  ;:*********************************************************
017102  000004                    TST27:  SCOPE
017104  012737  000027  001122            MOV     #27,$TSTNM        ;LOAD THE NUMBER OF THIS TEST
017112  012737  017630  001360            MOV     #TST30,NEXT       ;POINT TO THE START OF THE NEXT TEST
017120  012737  017534  001362            MOV     #18$,LOCK         ;SET FOR LOOP
017126  104417                            DCLASM                    ;CLEAR DEVICE AND SET MAINT BIT I↑BF I MODE
017130  013701  001366                    MOV     PAR,R1            ;PICK UP PARAMETERS
017134  012702  000001                    MOV     #1,R2             ;PICK UP INIT POINTER
017140  030237  001364            1$:     BIT     R2,LINE           ;SHOULD THIS LINE BE SET UP ?
017144  001402                            BEQ     2$                ;NO
017146  010177  162700                    MOV     R1,@DZLPR         ;SET UP LINE PARAMETERS
017152  005201                    2$:     INC     R1                ;POSITION POINTER TO THE NEXT LINE
017154  106302                            ASLB    R2                ;GOT 'EM ALL ?
017156  103370                            BCC     1$                ;IF NO, GO SET UP THE NEXT LINE
017160  005037  001372                    CLR     SAVLIN            ;CLEAR LINE # INDICATOR
017164  012700  001422                    MOV     #TD0,R0           ;POINT TO THE DATA AREA
017170  005020                            CLR     (R0)+             ;CLEAR A DATA WORD
017172  022700  001462                    CMP     #STOP,R0          ;FINISHED ?
017176  001374                            BNE     .-6               ;NO
017200  005000                            CLR     R0                ;CLEAR OFFSET
017202  012702  000001                    MOV     #1,R2             ;LINE POINTER
017206  052777  010040  162626            BIS     #MSENAB!SILOEN,@DZCSR   ;START SCANNER & SET SILO ENABLE
017214  030237  001364            3$:     BIT     R2,LINE           ;VALID LINE?
017220  001002                            BNE     .+6               ;YES
017222  000137  017510                    JMP     17$               ;TRY NEXT LINE
017226  013700  001372                    MOV     SAVLIN,R0         ;MAKE OFFSET
017232  006300                            ASL     R0                ;MAKE POWER OF TWO
017234  010277  162616  ↑B                MOV     R2,@DZTCR         ;SET TCR BIT
017240  105777  162576            4$:     TSTB    @DZCSR            ;REC DONE = 1 ?
017244  100001                            BPL     .+4
017246  104020                            ERROR   20                ;REC DONE SHOULD NOT = 1
017250  005003                            CLR     R3                ;SET CHARACTER COUNT
017252  005004                    5$:     CLR     R4
017254  032777  100000  162560    6$:     BIT     #TRDY,@DZCSR
017262  001004                            BNE     7$
017264  104414                            DELAY
017266  105204                            INCB    R4
017270  001371                            BNE     6$
017272  104003                            ERROR   3                 ;*TRDY FAILED TO SET
017274  116077  001422  162564    7$:     MOVB    TD0(R0),@DZTDR    ;LOAD A CHARACTER
```

```
017302  005260  001422                  INC     TDO(R0)             ;SET UP NEXT CHARACTER
017306  020327  000017                  CMP     R3,#15.             ;16 CHARACTERS ?
017312  103006                          BHIS    8$
017314  032777  020000  162520          BIT     #SILOAL,@DZCSR      ;SILO ALARM = 0 ?
017322  001401                          BEQ     .+4                 ;YES
017324  104013                          ERROR   13                  ;*SILO ALARM SHOULD NOT = 1
                                                                    ;UNTIL 16. DATA CHARACTERS
017326  000411                          BR      10$
017330  005004                  8$:     CLR     R4
017332  032777  020000  162502  9$:     BIT     #SILOAL,@DZCSR
017340  001004                          BNE     10$
017342  104414                          DELAY
017344  005204                          INC     R4
017346  001371                          BNE     9$
017350  104014                          ERROR   14                  ;*SILO ALARM FAILED TO SET!
                                                                    ;SILO ALARM SHOULD =1 AFTER 16.
                                                                    ;DATA CHARACTERS
017352  005203                  10$:    INC     R3                  ;INC CHAR COUNT
017354  022703  000102                  CMP     #66.,R3             ;FINISHED SENDING CHARACTERS ?
017360  001334                          BNE     5$                  ;NO
017362  005004                          CLR     R4
017364  104414                          DELAY
017366  105204                          INCB    R4
017370  001375                          BNE     .-4
                                        ;NOW LETS READ THE SILO
017372  013705  001372                  MOV     SAVLIN,R5           ;MAKE EXPECTED L↑BINE #
017376  105737  001371                  TSTB    MODE+1              ;IS THIS TEST IN STAGGERED MODE?
017402  001406                          BEQ     13$                 ;IF NOT, SKIP STAGGERED SETUP

                                        ;WE MUST NOW INVERT THE LAST BIT OF THE LINE NUMBER

017404  006205                          ASR     R5                  ;GET THE LAST BIT INTO THE CARRY BIT
017406  103402                          BCS     11$                 ;IF IT IS SET, GO CLEAR IT
017410  000261                          SEC                         ;IF IT IS CLEAR SET IT HERE
017412  000401                          BR      12$                 ;SKIP THE CLEARING
017414  000241                  11$:    CLC                         ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
017416  006105                  12$:    ROL     R5                  ;GET THE NEW BIT BACK INTO R5
017420  00030↑B5                        13$:    SWAB    R5          ;PUT IN UPPER BYTE
017422  052705  100000                  BIS     #DVALID,R5          ;ADD DATA VALID
017426  017704  162414          14$:    MOV     @DZRBUF,R4          ;ACTUAL
017432  020405                          CMP     R4,R5               ;ACTUAL VS. EXPECTED
017434  001401                          BEQ     15$                 ;YES
017436  104006                          ERROR   6                   ;*DATA/CONTENTS DID NOT COMPARE
017440  032777  020000  162374  15$:    BIT     #SILOAL,@DZCSR      ;SILO ALARM= 0 ?
017446  001401                          BEQ     16$                 ;YES
017450  104016                          ERROR   16                  ;READING DZRBUF DID NOT CLEAR SILO ALARM
017452  005205                  16$:    INC     R5                  ;UP CHARACTER
017454  120527  000077                  CMPB    R5,#63.             ;LAST SILO CHAR ?....64TH CHAR
017460  101762                          BLOS    14$
017462  005205                          INC     R5                  ;ADD 1 MORE FOR THE CLOBBERED CHAR
017464  052705  040000                  BIS     #OVRRUN,R5          ;ADD OVERRUN TO EXPECTED
017470  120527  000101                  CMPB    R5,#65.             ;LAST CHARACTER ?
017474  001754                          BEQ     14$
017476  017704  162344                  MOV     @DZRBUF,R4          ;FOR GOOD MEASURE
017502  005704                          TST     R4                  ;DATA VALID SHOULD = 0
017504  100001                          BPL     17$                 ;YES
```

```
017506  104017                               ERROR    17                  ;DATA VALID SHOULD = ↑B0
017510  040277  162342              17$:     BIC      R2,@DZTCR           ;CLR TCR BIT
017514  104401                               SCOP1                        ;LOOP?
017516  005237  001372                       INC      SAVLIN              ;INC EXPECTED LINE
017522  106302                               ASLB     R2                  ;NEXT LINE
017524  103402                               BCS      .+6                 ;NO
017526  000137  017214                       JMP      3$                  ;YES
017532  104400                               ADVANCE                      ;GO TO NEXT TEST

                                    ;TIGHT SCOPE LOOP FOR THIS TEST. SENDS 20. CHARACTERS
                                    ;ON DZ LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
                                    ;USED TO SCOPE SILO ALARM PULSES, ETC.

017534  052777  010040  162300  18$:     BIS      #MSENAB!SILOEN,@DZCSR   ;SETUP DEVICE
01↑B7542        012777  017620  162326            MOV      #20$,@DZTIV     ;SETUP TRANSMITTER VECTOR
017550  012737  000024  001216           MOV      #20.,$TMPO          ;TEMPORARY COUNT OF CHARACTER BURST
017556  050277  162274                   BIS      R2,@DZTCR           ;ENABLE LINE
017562  052777  040000  162252           BIS      #TIE,@DZCSR         ;ENABLE INTERRUPTS
017570  106427  000000                   MTPS     #0                  ;LOWER PRIORITY
017574  000001                      19$: WAIT                         ;ALLOW INTERRUPTS
017576  005337  001216                   DEC      $TMPO               ;REDUCE COUNT. ALL CHARACTERS SENT?
017602  001374                           BNE      19$                 ;IF NO, WAIT FOR MORE
017604  042777  050040  162230           BIC      #SILOEN!MSENAB!TIE,@DZCSR ;RESET SILO COUNTER, CLEAR STROBE
017612  104401                           SCOP1                        ;LOOP AGAIN?
017614  000137  017510                   JMP      17$                 ;IF NOT, RETURN TO WHERE YOU LEFT OFF
017620  112777  000252  162240  20$:     MOVB     #252,@DZTDR         ;SEND A CHARACTER
017626  000002                           RTI                          ;ALLOW MORE CHARACTERS TO COME
                                         ;*********************** TEST 30 ******************************
                                         ;*THIS TEST THAT "SILO ENABLE" WILL INHIBIT
                                         ;*RECEIVER INTERRUPTS AND THAT ON THE
                                         ;*16TH CHAR THAT "SILO ALARM" WILL CAUSE AN
                                         ;*INTERRUPT WITH "RIE" SET.
                                         ;*THIS WILL DO ALL SELECTED LINES ONE AT A TIME.
                                    ;;*  TEST 30
                                    ;;*↑B************************************************************
017630  000004                      TST30:   SCOPE
017632  012737  000030  001122           MOV      #30,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
017640  012737  020212  001360           MOV      #TST31,NEXT         ;POINT TO THE START OF THE NEXT TEST
017646  012737  017734  001362           MOV      #3$,LOCK            ;SET FOR LOOP
017654  104417                           DCLASM                       ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
017656  013701  001366                   MOV      PAR,R1              ;PICK UP PARAMETERS
017662  012702  000001                   MOV      #1,R2               ;PICK UP INIT POINTER
017666  030237  001364          1$:      BIT      R2,LINE             ;SHOULD THIS LINE BE SET UP ?
017672  001402                  ↑B       BEQ      2$                  ;NO
017674  010177  162152                   MOV      R1,@DZLPR           ;SET UP LINE PARAMETERS
017700  005201                  2$:      INC      R1                  ;POSITION POINTER TO THE NEXT LINE
017702  106302                           ASLB     R2                  ;GOT 'EM ALL ?
017704  103370                           BCC      1$                  ;IF NO, GO SET UP THE NEXT LINE
017706  005037  001372                   CLR      SAVLIN              ;CLEAR LINE # INDICATOR
017712  012700  001422                   MOV      #TD0,R0             ;POINT TO THE DATA AREA
017716  005020                           CLR      (R0)+               ;CLEAR A DATA WORD
017720  022700  001462                   CMP      #STOP,R0            ;FINISHED ?
017724  001374                           BNE      .-6                 ;NO
017726  005000                           CLR      R0                  ;CLEAR OFFSET
017730  012702  000001                   MOV      #1,R2               ;LINE POINTER
017734  012777  020154  162130  3$:      MOV      #11$,@DZRIV         ;SET FOR UNEXPECTED INTER.
```

```
017742  012777  000340  162124          MOV     #PR7,@DZRIS         ;SET PRIO.
017750  052777  010140  162064          BIS     #MSENAB!SILOEN!RIE,@DZCSR
                                                                    ;START SCANNER & SET SILO ENABLE
017756  030237  001364                  BIT     R2,LINE             ;VALID LINE?
017762  001002                          BNE     .+6                 ;YES
017764  000137  020164                  JMP     17$                 ;TRY NEXT LINE
017770  005777  162052                  TST     @DZRBUF             ;EMPTY THE↑B SILO
017774  100775                          BMI     .-4                 ;BR IF DATA VALID IS SET!
017776  106427  000000                  MTPS    #0                  ;SET PROCESSOR PRIORITY TO 0
020002  013700  001372                  MOV     SAVLIN,R0           ;MAKE OFFSET
020006  006300                          ASL     R0                  ;MAKE POWER OF TWO
020010  010277  162042                  MOV     R2,@DZTCR           ;SET TCR BIT
020014  005004                  5$:     CLR     R4
020016  032777  100000  162016  6$:     BIT     #TRDY,@DZCSR
020024  001004                          BNE     7$
020026  104414                          DELAY
020030  005204                          INC     R4
020032  001371                          BNE     6$
020034  104003                          ERROR   3                   ;*TRDY FAILED TO SET
020036  116077  001422  162022  7$:     MOVB    TDO(R0),@DZTDR      ;LO↑BAD A CHARACTER
020044  005260  001422                  INC     TDO(R0)             ;SET UP NEXT CHARACTER
020050  022760  000017  001422          CMP     #15.,TDO(R0)        ;15 CHARS YET?
020056  001406                          BEQ     8$
020060  032777  020000  161754          BIT     #SILOAL,@DZCSR      ;SILO ALARM = 0 ?
020066  001401                          BEQ     .+4                 ;YES
020070  104013                          ERROR   13                  ;*SILO ALARM SHOULD NOT = 1
                                                                    ;UNTIL 16. DATA CHARACTERS
020072  000751                          BR      6$
020074  012777  020162  161770  8$:     MOV     #12$,@DZRIV         ;SET NEW VECTOR
020102  032777  100000  161732          BIT     #TRDY,@DZCSR        ;READY FOR 16TH CHAR
020110  001774                          BEQ     .-6
020112  016077  001422  161746          MOV     TDO(R0),@DZTDR      ;LOAD THE 16TH CHAR.
020120  005004                          CLR     R4
020122  032777  020000  161712  9$:     BIT     #SILOAL,@DZCSR
020130  001005                          BNE     10$
020132  104414                          DELAY
020134  005204                          INC     R4
020136  001371                          BNE     9$
020140  104014                          ERROR   14                  ;*SILO ALARM FAILED TO SET!
020142  000410                          BR      17$                 ;SILO ALARM SHOULD =1 AFTER 16.
                                                                    ;DATA CHARACTERS
020144  000240                  10$:    NOP                         ;STALL
020146  000240                          NOP
020150  104000                          ERROR                       ;SILO ALARM NOT INTERRUPTING.
020152  000404                          BR      17$                 ;CONTINUE TEST.
020154  022626                  11↑B$:  CMP     (SP)+,(SP)+         ;FAKE RTI
020156  104012                          ERROR   12                  ;RX SHOULD NOT INTERRUPT
020160  000401                          BR      17$                 ;CONTINUE
020162  022626                  12$:    CMP     (SP)+,(SP)+         ;GOOD INTERRUPT TO HERE.
020164  040277  161666          17$:    BIC     R2,@DZTCR           ;CLR TCR BIT
020170  104401                          SCOP1                       ;LOOP?
020172  005237  001372                  INC     SAVLIN              ;INC EXPECTED LINE
020176  106302                          ASLB    R2                  ;NEXT LINE
020200  103402                          BCS     .+6                 ;NO
020202  000137  017734                  JMP     3$                  ;YES
020206  005037  001362                  CLR     LOCK                ;CLEAR TIGHT LOOP FOR NEXT TEST
```

```
                                            ;*********************** TEST 31 ****************************
                                            ;*THIS TEST RUNS ALL LINES FULL BORE
                                            ;*BASED UPON QUALIFIED LINES
                                            ;*..THIS IS AN INTERRUPT TEST ON THE RECEIVER AND
                                            ;*TRANSMITTER
                                    ;;*  TEST 31
                                    ;;*****************************************************
020212  000004                      TST31:  SCOPE
020214  012737  000031  001122              MOV     #31,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
020222  012737  021020  001360              MOV     #TST32,NEXT     ;POINT TO THE START OF THE NEXT TEST
020230  104417                              DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
020232  013737  001364  021016              MOV     LINE,RXTCR      ;SET IMAGE OF TCR BITS
020240  013701  001366              RSTART: MOV     PAR,R1          ;PICK UP PARAMETER
020244  012700  000001                      MOV     #1,R0           ;;PICK UP INIT POINTER
020250  030037  001364              INIT:   BIT     R0,LINE         ;SHOULD THIS LINE BE SET UP
020254  001402                              BEQ     1$              ;NO
020256  010177  161570                      MOV     R1,@DZLPR       ;SET UP LINE PARAM REGISTER
020262  005201                      1$:     INC     R1
020264  106300                              ASLB    R0              ;GOT 'EM ALL ?
020266  103370                              BCC     INIT            ;NO
020270  012700  001422  ↑B                  MOV     #TD0,R0         ;CLEAR TRANS DATA POINTER & REC POINTERS
020274  005020                      INIT1:  CLR     (R0)+
020276  022700  001462                      CMP     #STOP,R0                ;FINISHED ?
020302  001374                              BNE     INIT1           ;NO, CONTINUE CLEARING
020304  012777  020542  161560              MOV     #RXSVC,@DZRIV   ;SET UP REC INTR VECTOR
020312  012777  000340  161554              MOV     #PR7,@DZRIS     ;STATUS
020320  012777  020442  161550              MOV     #TXSVC,@DZTIV   ;SET UP TRANS INTR VECTOR
020326  012777  000340  161544              MOV     #PR7,@DZTIS     ;STATUS
020334  052777  000040  161500              BIS     #MSENAB,@DZCSR  ;SET MASTER SCAN ENABLE
020342  052777  000100  161472              BIS     #RIE,@DZCSR             ;SET REC I↑BNTR ENABLE
020350  052777  040000  161464              BIS     #TIE,@DZCSR             ;SET TRANS INTR ENABLE
020356  113777  001364  161472              MOVB    LINE,@DZTCR     ;SET TCR BITS...UP UP AND AWAY !
020364  106437  026220                      MTPS    @#LESS1                 ;ALLOW INTERRUPTS


020370  005037  020440              SNAP:   CLR     66$
020374  013727  006604              67$:    MOV     DLYCNT,(PC)+    ;SET FOR DELAY
020400  000000                      68$:    0
020402  005337  020400                      DEC     68$
020406  001375                              BNE     .-4
020410  105737  021016                      TSTB    RXTCR           ;WAIT FOR ALL RECIEVERS TO FINISH
020414  001002                              BNE     3$              ;
020416  000137  020716                      JMP     OUT
020422  005237  020440              3$:     INC     66$
020426  001362                              BNE     67$
020430  104007                              ERROR   7               ;*TRANSMITTER FAILED TO INTERRUPT
020432  104011                              ERROR   11              ;*RECEIVER FAILED TO INTERRUPT
020434  000137  020770                      JMP     FINI
020440  000000                      66$:    0

                                    ;TRANS INTR SVC ROUTINE
020442  005777  161374              TXSVC:  TST     @DZCSR          ;TRANS INTR ?
020446  100401                              BMI     .+4
020450  104003                              ERROR   3               ;*TRANSMITTER FAILED
020452  117703  161366                      MOVB    @HDZCSR,R3      ;SAVE IT
                                    ;NOW TEST FOR LINE # ETC
```

```
020456  042703  177770                    BIC     #↑C<7>,R3            ;STRIP JUNK
020462  010304                            MOV     R3,R4                ;SAVE
020464  012702  000001                    MOV     #1,R2                ;SET UP POSITION POINTER
020470  105303                    3$:     DECB    R3                   ;IS IT THIS LINE ?
020472  100402                            BMI     4$                   ;YES
020474  006302                            ASL     R2                   ;UP THE LINE #
020476  000774                            BR      3$                   ;GO 'ROUND AGAIN
020500  030237  001364            4$:     BIT     R2,LINE              ;VALID LINE?
020504  001001                            BNE     .+4                  ;YES
020506  104011                            ERROR   11                   ;NO,INVALID LINE!!!!
020510  042704  177770                    BIC     #↑C<7>,R4            ;STRIP JUNK
        020514  006304                    ASL             R4           ;MAKE POWER OF 2
020516  116477  001422  161342            MOVB    TDO(R4),@DZTDR       ;LOAD CHARACTER
020524  105264  001422                    INCB    TDO(R4)              ;SET UP NEXT CHARACTER
020530  001002                            BNE     5$                   ;LAST CHARACTER ?
020532  040277  161320                    BIC     R2,@DZTCR            ;YES ,CLEAR TCR BIT
020536  005200                    5$:     INC     R0                   ;INCR RECEIVER TIMER
020540  000002                            RTI



                                  ;REC INTR SVC ROUTINE
020542  105777  161274            RXSVC:  TSTB    @DZCSR               ;REC DONE ?
020546  100401                            BMI     .+4                  ;YES
020550  104004                            ERROR   4                    ;FALSE INTERRUPT
020552  032777  020000  161262            BIT     #SILOAL,@DZCSR       ;SILO ALARM?
020560  001401                            BEQ     .+4                  ;NO
020562  104000                            ERROR                        ;SILO ALARM SHOULD NOT =1
020564  017704  161256                    MOV     @DZRBUF,R4           ;SAVE IT
020570  100401                            BMI     .+4                  ;YES
020572  104000                            ERROR                        ;YOU LOSE  ...DATA VALID WAS'NT SET
020574  032704  070000                    BIT     #OVRRUN!FRMERR!PARER,R4
020600  001401                            BEQ     .+4
020602  104000                            ERROR                        ;RECEIVER ERROR FLAG/S WERE SET
020604  010403                            MOV     R4,R3
020606  000303                            SWAB    R3
020610  042703  177770                    BIC     #↑C<7>,R3            ;STRIP JUNK
020614  010337  001372                    MOV     R3,SAVLIN            ;SAVE LINE NUMBER
020620  ↑B012702        000001            MOV             #1,R2                ;SET UP POSITION POINTER
020624  105303                    5$:     DECB    R3
020626  100402                            BMI     6$
020630  006302                            ASL     R2                   ;RE POSITION POINTER
020632  000774                            BR      5$                   ;GO 'ROUND AGAIN
020634  030237  001364            6$:     BIT     R2,LINE              ;LINE VALID ?
020640  001001                            BNE     .+4                  ;YES
020642  104011                            ERROR   11                   ;INVALID LINE #
020644  013703  001372                    MOV     SAVLIN,R3            ;GET THE LINE NUMBER AGAIN
020650  006303                            ASL     R3                   ;USE R3 AS A POINTER IN THE DATA TABLE
020652  126304  001442                    CMPB    TRO(R3),R4           ;DOES THE DATA CHARACTER COMPARE ?
020656  001405                            BEQ     2$                   ;YES
        020660  016305  001442            MOV             TRO(R3),R5   ;SAVE EXPECTED
020664  042704  177400                    BIC     #↑C<377>,R4          ;CLEAR JUNK
                                                                       ;R2 = LINE # BY BIT POSITION
                                                                       ;R4 = ACTUAL DATA
                                                                       ;R5 = EXPECTED DATA
020670  104005                            ERROR   5                    ;#NO, DATA DOES NOT COMPARE
020672  005263  001442            2$:     INC     TRO(R3)              ;SET UP FOR NEXT CHARACTER
```

```
020676  105763  001442                  TSTB    TRO(R3) ;ALL CHARS DONE?
020702  001002                          BNE     .+6
020704  040237  021016                  BIC     R2,RXTCR        ;ZERO LINE DONE INDICATOR.
020710  012716  020370                  MOV     #SNAP,(SP)      ;RESET THE BACKGROUND TIMING LOOP
020714  000002                          RTI


                                        ;FINISH UP ROUTINE
020716  106427  000340          OUT:    MTPS    #PR7                    ;STOP ALL INTERRUPTS
020722  104413                          DEVICE.CLR              ;CLEAR ALL INTERRUPTS AWAY
020724  005003                          CLR     R3
020726  005037  001372                  CLR     SAVLIN
020732  012702  000001                  MOV     #1,R2
020736  030237  001364          1$:     BIT     R2,LINE         ;VALID LINE ?
020742  001405                          BEQ     2$              ;NO
020744  022763  000400  001442          CMP     #400,TRO(R3)    ;RECEIVED A BINARY COUNT PATTER↑BN ?
020752  001401                          BEQ     .+4             ;YES
020754  104000                          ERROR   0               ;THE LINE FAILED TO RECEIVE A FULL
                                                                ;BINARY COUNT PATTERN
020756  005237  001372          2$:     INC     SAVLIN          ;SET UP FOR NEXT LINE
020762  005723                          TST     (R3)+           ;ADD 2
020764  106302                          ASLB    R2              ;SET UP NEXT LINE POINTER
020766  103363                          BCC     1$              ;FINISHED ?
020770                          FINI:
020770  013777  002074  161074          MOV     DZRIS,@DZRIV    ;RESTORE TRAPCATCHER
020776  005077  161072                  CLR     @DZRIS
021002  013777  002100  161066          MOV     DZTIS,@DZTIV
021010  005077  161064                  CLR     @DZTIS
021014  104400                          ADVANCE                 ;G↑BO TO THE NEXT TEST
021016  000000                  RXTCR:  0                       ;RX IMAGE OF TCR BITS


                                        ;************************ TEST 32 ******************************
                                        ;*DZ11 RELATIVE TIMING TEST.
                                        ;*EACH SELECTED LINE WILL IN TURN RUN 16. CHARS
                                        ;*AT ALL BAUD RATES AND THEN THE HIGHEST BAUD
                                        ;*WITH ALL CHAR LENGTHS. EACH NEW PARAMETER SHOULD
                                        ;*DECREASE IN TIME FROM THE PREVIOUS PARAMETERS SELECTED.
                                        ;*THE TIME IS CHECKED AGAINST THE LAST PARAMETER USED
                                        ;* AND A LOWER TIME IS EXPECTED ON THE CURRENT PARAMETER.
                                        ;*PARAMETERS ARE:
                                        ;*  EIGHT BITS/PER/CHAR - TWO STOP BITS AT
                                        ;*       50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000
                                        ;*       2400, 3600, 4800, 7200, 9600 BAUD.
                                        ;*  19.2 K BAUD - TWO STOP BITS AT
                                        ;*       SEVEN, SIX, FIVE BITS/PER/CHAR.
                                        ;*AFTER EACH LINE HAS FINISHED ALL THE ABOVE PARAMETERS
                                        ;*THE NEXT SELECTED LINE IS THE TESTED.
                                ;;*  TEST 32
                                ;*****************************************************************
021020  000004                  TST32:  SCOPE
021022  012737  000032  001122          MOV     #32,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
021030  012737  000002  ↑B001226                MOV     #2,$TIMES
021036  012737  021514  001360          MOV     #TST33,NEXT     ;POINT TO THE START OF THE NEXT TEST
021044  012737  021170  001362          MOV     #3$,LOCK        ;SET FOR LOOP
021052  005037  023140                  CLR     OFFSET          ;RESET THIS VARIABLE
```

```
021056  005037  001372                           CLR     SAVLIN          ;RESET LINE NUMBER INDICATOR
021062  005037  001374                           CLR     XMTLIN          ;USE THIS WORD TO TELL WHAT LINE TRANSMITTED
021066  012737  000001  001216                   MOV     #1,$TMPO        ;USE $TMPO AS A BIT POINTER
        021074  012737  010070  021512           MOV     #RCVON!SSO!EIGHT!TWOSTOP,7$ ;BUILD TEMPORARY PARAMETERS
021102  033737  001216  001364  1$:              BIT     $TMPO,LINE      ;IS THIS LINE ACTIVE?
021110  001027                                   BNE     3$              ;IF SO, GO GET STARTED
021112  012737  010070  021512  2$:              MOV     #RCVON!SSO!EIGHT!TWOSTOP,7$ ;LOAD PARAMETERS TEMPORARILY
021120  012700  001422                           MOV     #TDO,RO         ;POINT TO THE DATA AREA
021124  005020                                   CLR     (RO)+           ;CLEAR A DATA WORD
021126  022700  001462                           CMP     #STOP,RO        ;FINISHED ?
021132  001374                                   BNE     .-6             ;NO
021134  005237  001374                           INC     XMTLIN          ;POINT TO THE NEXT LINE TO TRANSMIT
021140  042737  000007  021512                   BIC     #7,7$           ;MAKE SURE TEMPORARY PARAMETERS POINT TO 0
021146  053737  001374  021512                   BIS     XMTLIN,7$       ;ADD DESIRED LINE NUMBER
021154  005037  023140                           CLR     OFFSET
021160  106337  001216                           ASLB    $TMPO           ;POINT TO THE NEXT LINE
021164  103346                                   BCC     1$              ;PROCESS THE NEXT LINE
021166  104400                                   ADVANCE                 ;TEST TO SEE IF THIS TEST GETS REPEATED
021170                          3$:
021170  104417                                   DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
021172  042737  010000  021512                   BIC     #RCVON,7$       ;ZERO PARAMTERS FOR TX LINE
021200  013777  021512  160644                   MOV     7$,@DZLPR       ;LOAD PARAMTERS FOR TX↑B
021206  005737  001370                           TST     MODE            ;STAGGERED?
021212  100011                                   BPL     100$            ;BR IF NO
021214  000241                                   CLC                     ;SET UP LINE
021216  006037  021512                           ROR     7$              ;
021222  103002                                   BCC     98$             ;BR IF LINE WAS EVEN
021224  000241                                   CLC                     ;PREPARE TO MKE LINE EVEN
021226  000401                                   BR      99$             ;CONTINUE
021230  000261                  98$:             SEC                     ;PREPARE TO MAKE LINE ODD
021232  006137  021512          99$:             ROL     7$              ;SET ALTERED LINE
021236  052737  010000  021512  100$:            BIS     #RCVON,7$       ;SET RX ON
021244  013777  021512  160600                   MOV     7$,@DZLPR       ;LOAD RX PARAMETERS
021252  042737  000007  021512                   BIC     ↑B07,7$         ;CLEAR OLD LINE #
021260  053737  001374  021512                   BIS     XMTLIN,7$       ;SET LINE UP AGAIN
021266  013737  021512  001400                   MOV     7$,REGIST       ;SAVE PARAMETERS FOR PRINTOUT
021274  012700  001422                           MOV     #TDO,RO         ;POINT TO THE DATA AREA
021300  005020                                   CLR     (RO)+           ;CLEAR A DATA WORD
021302  022700  001462                           CMP     #STOP,RO        ;FINISHED ?
021306  001374                                   BNE     .-6             ;NO
021310  005002                                   CLR     R2              ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
021312  005003                                   CLR     R3              ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
021314  005037  001220                           CLR     $TMP1           ;INITIALIZE THE TIMER
021320  005037  001224                           CLR     $TMP3           ;INITIALIZE THESE BITS ALSO
021324  012737  000020  001376                   MOV     #20,XMTCNT      ;SET HOW MANY CHARACTERS TO TRANSMIT
021332  012777  022600  160536                   MOV     #XMTSRV,@DZTIV
021340  012777  022724  160524                   MOV     #RXISR1,@DZRIV
021346  013777  026216  160520                   MOV     DZPRT,@DZRIS
021354  013777  026216  160516                   MOV     DZPRT,@DZTIS
021362  113777  001216  160466                   MOVB    $TMPO,@DZTCR    ;START THE VALID LINE
021370  052777  040140  160444                   BIS     #TIE!RIE!MSENAB,@DZCSR
021376  106427  000000                           MTPS    #0              ;LOWER THE PRIORITY TO ALLOW INTERRUPTS
021402  032777  000100  160432  4$:              BIT     #RIE,@DZCSR     ;IS ROUTIN↑BE DONE?
021410  001407                                   BEQ     5$              ;WHEN ALL IS DONE RX IE IS CLEARED IN ISR.
021412  005237  001220                           INC     $TMP1           ;COUNT TIME
021416  001371                                   BNE     4$              ;CONTINUE TEST
```

```
021420  105237  001224                    INCB    $TMP3           ;DOUBLE COUNT
021424  001366                            BNE     4$              ;CONTINUE TEST
021426  104011                            ERROR   11              ;INTERRUPTS NOT FINISHED
021430  004737  007242          5$:       JSR     PC,SERV.G       ;<↑G>?
021434  104401                            SCOP1                   ;LOOP?
021436  062737  000002  023140  ↑B        ADD     #2,OFFSET
021444  013700  021512                    MOV     7$,R0
021450  042700  170377                    BIC     #↑C<17*400>,R0
021454  022700  007400                    CMP     #<17*400>,R0
021460  001010                            BNE     6$
021462  032737  000030  021512            BIT     #BIT4+BIT3,7$
021470  001610                            BEQ     2$
021472  162737  000010  021512            SUB     #BIT3,7$
021500  000633                            BR      3$
021502  062737  000400  021512  6$:       ADD     #400,7$
021510  000627                            BR      3$
021512  000000                  7$:       0
                                          ;*********************** TEST 33 ****************************
                                          ;* THIS TEST VERIFIES THAT EVEN PARITY WORKS
                                          ;* FOR ALL ODD LINES SELECTED AND THAT ODD PARITY WORKS FOR ALL
                                          ;* EVEN LINES SELECTED.
                                          ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
                                          ;*THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
                                          ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
                                          ;*YOU ARE IN "STAGGERED" MODE.
                                          ;*40(8) CHARS ARE USED FOR THIS TEST.
                                          ;*ALL SELECTED LINES WILL BE ENABLED
                                          ;*AT THE SAME TIME!
                                ;;*  TEST 33
                                ;;*************************************************************
021514  000004                  ↑ST33:    SCOPE
021516  012737  000033  001122            MOV     #33,$TSTNM      ;LOAD THE NUMBER↑B OF THIS TEST
021524  012737  022146  001360            MOV     #TST34,NEXT     ;POINT TO THE START OF THE NEXT TEST
021532  005737  001370                    TST     MODE            ;IS THIS STAGGERED MODE?
021536  100111                            BPL     6$              ;IF NOT, DON'T DO THIS TEST
021540  104417                            DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
021542  013701  001366                    MOV     PAR,R1          ;USE R1 TO BUILD PARAMETERS TO BE LOADED
021546  042701  000200                    BIC     #ODDPAR,R1      ;MAKE SURE ODD PARITY ISN'T SET
021552  052701  000100                    BIS     #PARITY,R1      ;MAKE SURE PARITY IS TURNED ON
021556  012702  000001                    MOV     #1,R2           ;USE R2 AS A LINE POINTER
021562  030237  0013↑B64        1$:       BIT         R2,LINE         ;IS THIS A VALID LINE?
021566  001411                            BEQ     3$              ;IF NOT, SKIP TO THE NEXT LINE
021570  032701  000001                    BIT     #BIT0,R1        ;IS THIS LINE AN ODD LINE?
021574  001002                            BNE     2$              ;IF IT'S ODD, USE EVEN PARITY
021576  052701  000200                    BIS     #ODDPAR,R1      ;IF IT'S EVEN, USE ODD PARITY
021602  010177  160244          2$:       MOV     R1,@DZLPR       ;LOAD THE LINE PARAMETER REGISTER
021606  042701  000200                    BIC     #ODDPAR,R1      ;SET UP THE NEXT PARITY TO EVEN
021612  005201                  3$:       INC     R1              ;POINT TO THE NEXT LINE
021614  106302                            ASLB    R2              ;MOVE THE BIT POINTER IN R2 TO THE NEXT LINE
021616  103361                            BCC     1$              ;IF WE'RE NOT DONE, GO CHECK THE NEXT LINE
021620  005037  001372                    CLR     SAVLIN          ;CLEAR THE LINE NUMBER INDICATOR
021624  005002                            CLR     R2              ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
021626  005003                            CLR     R3              ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
021630  012737  000040  001376            MOV     #40,XMTCNT      ;TRANSMIT A BINARY COUNT PATTERN(00-40)
021636  012700  001422                    MOV     #TD0,R0         ;POINT TO THE DATA AREA
021642  005020                            CLR     (R0)+           ;CLEAR A DATA WORD
```

```
021644  022700  001462                   CMP     #STOP,R0         ;FINISHED ?
021650  001374                           BNE     .-6              ;NO
021652  005000                           CLR     R0               ;CLEAR OFFSET
021654  012777  022600  160214           MOV     #XMTSRV,@DZTIV   ;SET UP THE TRANSMITTER INTERRUPT VECTOR
021662  012777  021770  160202           MOV     #9$,@DZRIV       ;SET UP THE RECEIVER INTERRUPT VECTOR
021670  013777  026216  160176           MOV     DZPRT,@DZRIS     ;SET THE INTERRUPT VECTOR STATUS
021676  013777  026216  160174           MOV     DZPRT,@DZTIS     ;SET TRANSMITTER INTERRUPT PRIORITY
021704  052777  040140  160130           BIS     #RIE!TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
021712  113777  001364  1601↑B36                 MOVB    LINE,@DZTCR      ;ENABLE ALL SELECTED LINES
021720  106427  000000                   MTPS    #0               ;ALLOW INTERRUPTS
021724  005037  021764          4$:      CLR     7$
021730  005037  021766                   CLR     8$
021734  032777  000100  160100  5$:      BIT     #RIE,@DZCSR      ;WHEN RX DONE; RIE WILL =0
021742  001407                           BEQ     6$               ;BR IF ALL DONE
021744  005237  021764                   INC     7$
021750  001371                           BNE     5$
021752  105237  021766                   INCB    8$
021756  100366                           BPL     5$
021760  104011                           ERROR   11               ;*RX FAILED TO FINISH (INTERRUPT)
021762  104400                  6$:      ADVANCE                  ;ADVANCE LOOP
021764  000000                  7$:      0
021766  000000                  8$:      0




                                         ;RECEIVER SERVICE ROUTINE

021770  017704  160052          9$:      MOV     @DZRBUF,R4       ;GET THE CHARACTER
021774  100401                           BMI     10$              ;IF IT WAS VALID, CONTINUE TESTING
021776  104000                           ERROR                    ;ERROR- ILLEGAL CHAR... DATA VALID NOT SET
022000  010401                  10$:     MOV     R4,R1            ;COPY THE RECEIVED INFORMATION
022002  000301                           SWAB    R1               ;GET THE LINE NUMBER IN THE LOWER BYTE
022004  042701  177770                   BIC     #↑C<7>,R1        ;ISOLATE THE LINE NUMBER
022010  006301                           ASL     R1               ;ALIGN IT ON A WORD BOUNDARY
022012  032704  010000                   BIT     #PARER,R4        ;PARITY ERROR SHOULD BE SET. IS IT?
↑B022016        001013                           BNE     11$              ;IF SO, GO CHECK CHARACTER
022020  013737  002046  001400           MOV     DZRBUF,REGIST    ;SET UP FOR THE ERROR MESSAGE
022026  010405                           MOV     R4,R5
022030  042705  000377                   BIC     #377,R5
022034  156105  001442                   BISB    TR0(R1),R5       ;GET THE CORRECT CHARACTER
022040  052705  110000                   BIS     #DVALID!PARER,R5 ;BUILD WHAT WAS EXPECTED
022044  104006                           ERROR   6                ;*ERROR- DID NOT GET CORRECT INFORMATION
022046  126104  001442          11$:     CMPB    TR0(R1),R4       ;CHECK THE CHARACTER. IS IT CORRECT?
022052  001413                           BEQ     12$              ;IF SO, GO SET UP NEXT CHARACTER
022054  116105  001442                   MOVB    TR0(R1)↑B,R5     ;LOAD THE CHARACTER FOR ERROR REPORTING
022060  042705  177400                   BIC     #↑C<377>,R5      ;CLEAR SIGN EXTEND
022064  010137  001372                   MOV     R1,SAVLIN        ;GET THE LINE NUMBER FOR REPORTING
022070  006237  001372                   ASR     SAVLIN           ;ALIGN IT CORRECTLY
022074  042704  177400                   BIC     #↑C<377>,R4      ;REMOVE THE JUNK FROM R4, THE ACTUAL CHARACTER
022100  104005                           ERROR   5                ;DATA ERROR
022102  005261  001442          12$:     INC     TR0(R1)          ;SET UP THE NEXT CHARACTER
022106  005203                           INC     R3               ;ADD TO THE TOTAL RECEIVED COUNT
022110  032777  040000  157724           BIT     #TIE,@DZCSR      ;ARE TRANSMISSIONS DONE?
022116  001010                           BNE     13$              ;IF NO, GO RECEIVE SOME MORE
022120  020203                           CMP     R2,R3            ;ARE ALL CHARACTERS RECEIVED?
022122  001006                           BNE     13$              ;IF NO, GO RECEIVE SOME MORE
022124  042777  000100  157710           BIC     #RIE,@DZCSR      ;DISABLE RECEIVER INTERRUPTS
```

```
022132  012716  021762                  MOV     #6$,(SP)        ;CRUNCH THE STACK
022136  000002                          RTI                     ;RETURN AND FINISH
022140  012716  021724          13$:    MOV     #4$,(SP)        ;CRUNCH THE STACK
022144  000002                          RTI                     ;GO BACK TO RECEIVER WAIT L↑BOOP
                                        ;*********************** TEST 34 ******************************
                                        ;*THIS TEST VERIFIES THAT ODD PARITY WORKS FOR ALL ODD LINES
                                        ;* SELECTED AND THAT EVEN PARITY WORKS FOR ALL EVEN LINES SELECTED
                                        ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
                                        ;*THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
                                        ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
                                        ;*YOU ARE IN "STAGGERED" MODE.
                                        ;*40(8) CHARS ARE USED FOR THIS TEST.
                                        ;*ALL SELECTED LINES WILL BE ENABLED
                                        ;*AT THE SAME TIME!
                                ::*  TEST 34
                                ::**********↑B*******************************************************
022146  000004                  TST34:  SCOPE
022150  012737  000034  001122          MOV     #34,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
022156  012737  004444  001360          MOV     #$EOP,NEXT      ;POINT TO THE END-OF-PASS HANDLER
022164  005737  001370                  TST     MODE            ;IS THIS STAGGERED MODE?
022170  100111                          BPL     6$              ;IF NOT, DON'T DO THIS TEST
022172  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
022174  013701  001366                  MOV     PAR,R1          ;USE R1 TO BUILD PARAMETERS TO BE LOADED
022200  042701  000200                  BIC     #ODDPAR,R1      ;MAKE SURE ODD PARITY ISN'T SET
022204  052701  000100                  BIS     #PARITY,R1      ;MAKE SURE PARITY IS TURNED ON
022210  012702  000001                  MOV     #1,R2           ;USE R2 AS A LINE POINTER
022214  030237  001364          1$:     BIT     R2,LINE         ;IS THIS A VALID LINE?
022220  001411                          BEQ     3$              ;IF NOT, SKIP TO THE NEXT LINE
022222  032701  000001                  BIT     #BIT0,R1        ;IS THIS LINE AN ODD LINE?
022226  001402                          BEQ     2$              ;IF IT'S EVEN, USE EVEN PARITY
022230  052701  000200                  BIS     #ODDPAR,R1      ;IF IT'S ODD, USE ODD PARITY
022234  010177  157612          2$:     MOV     R1,@DZLPR       ;LOAD THE LINE PARAMETER REGISTER
022240  042701  000200                  BIC     #ODDPAR,R1      ;SET UP THE NEXT PARITY TO EVEN
↑B              022244  005201          3$:     INC     R1              ;POINT TO THE NEXT LINE
022246  106302                          ASLB    R2              ;MOVE THE BIT POINTER IN R2 TO THE NEXT LINE
022250  103361                          BCC     1$              ;IF WE'RE NOT DONE, GO CHECK THE NEXT LINE
022252  005037  001372                  CLR     SAVLIN          ;CLEAR THE LINE NUMBER INDICATOR
022256  005002                          CLR     R2              ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
022260  005003                          CLR     R3              ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
022262  012737  000040  001376          MOV     #40,XMTCNT      ;TRANSMIT A BINARY COUNT PATTERN(00-40)
022270  012700  001422                  MOV     #TD0,R0         ;POINT TO THE DATA AREA
022274  005020                          CLR     (R0)+           ;CL↑BEAR A DATA WORD
022276  022700  001462                  CMP     #STOP,R0        ;FINISHED ?
022302  001374                          BNE     .-6             ;NO
022304  005000                          CLR     R0              ;CLEAR OFFSET
022306  012777  022600  157562          MOV     #XMTSRV,@DZTIV  ;SET UP THE TRANSMITTER INTERRUPT VECTOR
022314  012777  022422  157550          MOV     #9$,@DZRIV      ;SET UP THE RECEIVER INTERRUPT VECTOR
022322  013777  026216  157544          MOV     DZPRT,@DZRIS    ;SET THE INTERRUPT VECTOR STATUS
022330  013777  026216  157542          MOV     DZPRT,@DZTIS    ;SET TRANSMITTER INTERRUPT PRIORITY
022336  052777  040140  157476          BIS     #RIE!TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
022344  113777  001364  157504          MOVB    LINE,@DZTCR     ;ENABLE ALL SELECTED LINES
022352  106427  000000                  MTPS    #0              ;ALLOW INTERRUPTS
022356  005037  022416          4$:     CLR     7$
022362  005037  022420                  CLR     8$
022366  032777  000100  157446  5$:     BIT     #RIE,@DZCSR     ;WHEN RX DONE; RIE WILL =0
022374  001407                          BEQ     6$              ;BR IF ALL DONE
```

```
022376  005237  022416                  INC     7$
022402  001371                          BNE     5$
022404  105237  022420                  INCB    8$
022410  100366                          BPL     5$
022412  104011                          ER↑BROR 11              ;*RX FAILED TO FINISH (INTERRUPT)
022414  104400                  6$:     ADVANCE                 ;ADVANCE LOOP
022416  000000                  7$:     0
022420  000000                  8$:     0



                                        ;RECEIVER SERVICE ROUTINE

022422  017704  157420          9$:     MOV     @DZRBUF,R4      ;GET THE CHARACTER
022426  100401                          BMI     10$             ;IF IT WAS VALID, CONTINUE TESTING
022430  104000                          ERROR                   ;ERROR- ILLEGAL CHAR... DATA VALID NOT SET
022432  010401                  10$:    MOV     R4,R1           ;COPY THE RECEIVED INFORMATION
022434  000301                          SWAB    R1              ;GET THE LINE NUMBER IN THE LOWER BYTE
022436  042701  177770                  BIC     #↑C<7>,R1       ;ISOLATE THE LINE NUM↑BBER
022442  006301                          ASL     R1              ;ALIGN IT ON A WORD BOUNDARY
022444  032704  010000                  BIT     #PARER,R4       ;PARITY ERROR SHOULD BE SET. IS IT?
022450  001013                          BNE     11$             ;IF SO, GO CHECK CHARACTER
022452  013737  002046  001400          MOV     DZRBUF,REGIST   ;SET UP FOR THE ERROR MESSAGE
022460  010405                          MOV     R4,R5
022462  042705  000377                  BIC     #377,R5
022466  156105  001442                  BISB    TRO(R1),R5      ;GET THE CORRECT CHARACTER
022472  052705  110000                  BIS     #DVALID!PARER,R5 ;BUILD WHAT WAS EXPECTED
022476  104006                          ERROR   6               ;*ERROR- DID NOT GET CORRECT INFORMATION
022500  126104  001442          11$:    CMPB    TRO(R1),R4      ;CHECK THE CHARACTER. IS IT CORRECT?
022504  001413                          BEQ     12$             ;IF SO, GO SET UP NEXT CHARACTER
022506  116105  001442                  MOVB    TRO(R1),R5      ;LOAD THE CHARACTER FOR ERROR REPORTING
022512  042705  177400                  BIC     #↑C<377>,R5     ;CLEAR SIGN EXTEND
022516  010137  001372                  MOV     R1,SAVLIN       ;GET THE LINE NUMBER FOR REPORTING
022522  006237  001372                  ASR     SAVLIN          ;ALIGN IT CORRECTLY
022526  042704  177400                  BIC     #↑C<377>,R4     ;REMOVE THE JUNK FROM R4, THE ACTUAL CHARACTER
022532  104005                          ERROR   5               ;DATA ERROR
022534  005261  001442          12$:    INC     TRO(R1)         ;SET UP THE NEXT CHARACTER
022540  00↑B5203                                INC     R3              ;ADD TO THE TOTAL RECEIVED COUNT
022542  032777  040000  157272          BIT     #TIE,@DZCSR     ;ARE TRANSMISSIONS DONE?
022550  001010                          BNE     13$             ;IF NO, GO RECEIVE SOME MORE
022552  020203                          CMP     R2,R3           ;ARE ALL CHARACTERS RECEIVED?
022554  001006                          BNE     13$             ;IF NO, GO RECEIVE SOME MORE
022556  042777  000100  157256          BIC     #RIE,@DZCSR     ;DISABLE RECEIVER INTERRUPTS
022564  012716  022414                  MOV     #6$,(SP)        ;CRUNCH THE STACK
022570  000002                          RTI                     ;RETURN AND FINISH
022572  012716  022356          13$:    MOV     #4$,(SP)        ;CRUNCH THE STACK
022576  000002                          RTI                     ;GO BACK TO RECEIVER WAIT LOOP↑B
```

```
                                             ;TRANSMITTER INTERRUPT SERVICE
                                             ;-----------------------------

022600  117701  157240          XMTSRV: MOVB    @HDZCSR,R1        ;GET THE LINE NUMBER. IS THE TRANSMITTER
022604  100401                          BMI     1$                ;REALLY READY? IF SO, GO LOAD THE CHARACTER
022606  104003                          ERROR   3                 ;*TRANSMITTER NOT READY- FALSE INTERRUPT
022610  042701  177770          1$:     BIC     #↑C<7>,R1         ;ISOLATE THE LINE NUMBER
022614  006301                          ASL     R1                ;MAKE SURE IT REFERENCES A WORD BOUNDARY
022616  116177  001422  157242          MOVB    TDO(R1),@DZTDR    ;LOAD THE CURRENT CHARACTER FOR THIS LINE
022624  005261  001422                  INC     TDO(R1)           ;SET UP NEXT CHARACTER FOR THIS LINE
022630  005202                          INC     R2                ;UP THE NUMBER OF TRANSMISSIONS
022632  023761  001376  001422          CMP     XMTCNT,TDO(R1)    ;HAVE WE DONE ALL PATTERNS ON THIS LINE?
022640  001015                          BNE     4$                ;IF NOT, KEEP ON TRANSMITTING
022642  012700  000001                  MOV     #1,R0             ;SET UP A DESELECTION POINTER
022646  006201                          ASR     R1                ;GET THE LINE NUMBER AGAIN
022650  005301                  2$:     DEC     R1                ;REDUCE THE COUNT. WAS THIS THE LINE?
0226↑B52        100402                          BMI     3$                ;IF SO, GO DISABLE THE ENABLE BIT FOR IT
022654  006300                          ASL     R0                ;MOVE THE POINTER TO THE NEXT LINE
022656  000774                          BR      2$                ;GO CHECK THE NEXT LINE
022660  140077  157172          3$:     BICB    R0,@DZTCR         ;DISABLE THE LINE POINTED TO BY R0
022664  001003                          BNE     4$                ;IF MORE LINES ARE ACTIVE , GO CONTINUE TRANSMIT
022666  042777  040000  157146          BIC     #TIE,@DZCSR       ;IF NOT, DISABLE TRANSMITTER INTERRUPTS
022674  000002                  4$:     RTI                       ;RETURN TO THE TIMING LOOP

                                             ; RELATIVE TIME BUILDING ROUTINE
                                             ; ------------------------------

022676  012737  0000↑B04        001222  BUILD:  MOV     #4,$TMP2          ;ROTATE 4 BITS BACK INTO $TMP1
022704  006037  001224          1$:     ROR     $TMP3             ;GET THE BITS FROM $TMP3, THE HIGH BYTE
022710  006037  001220                  ROR     $TMP1             ;OF THE RELATIVE TIME COUNTER. PUT THEM BACK
022714  005337  001222                  DEC     $TMP2             ;INTO $TMP1 USING THE CARRY BIT WITH
                                                                  ;ROTATE INSTRUCTIONS
022720  001371                          BNE     1$                ;REDUCE COUNT. ALL BITS BACK? IF NOT, GET MORE
022722  000207                          RTS     PC                ;RETURN TO CALLING TEST
```

```
                                                    ;RECEIVER SERVICE ROUTINE

022724  105777  157112          RXISR1: TSTB    @DZCSR          ;IS THE RECEIVER REALLY READY?
022730  100401                          BMI     1$              ;IF SO, GO SERVICE IT
022732  104004                          ERROR   4               ;*ERROR- RECEIVER DONE FLAG ISN'T SET
022734  017704  157106          1$:     MOV     @DZRBUF,R4      ;SAVE THE RECEIVER INFORMATION
022740  100401                          BMI     2$              ;IF IT WAS VALID, GO PROCESS IT
022742  104000                          ERROR                   ;ERROR- DATA VALID WASN'T SET
022744  032704  070000          2$:     BIT     #OVRRUN!FRMERR!PARER,R4 ;ARE ANY ERROR FLAGS SET?
022750  001404                          BEQ     3$              ;IF NOT, GO CONTINUE PROCESSING
02↑B2752        013737  002046  001400          MOV     DZRBUF,REGIST   ;SET UP FOR ERROR REPORTING
022760  104002                          ERROR   2               ;ERROR- RECEIVER ERROR FLAG SET
022762  010401                  3$:     MOV     R4,R1           ;COPY THE RECEIVER INFORMATION
022764  000301                          SWAB    R1              ;GET THE LINE NUMBER IN THE LOWER BYTE
022766  042701  177770                  BIC     #↑C<7>,R1       ;ISOLATE THE LINE NUMBER
022772  006301                          ASL     R1              ;ALIGN IT ON A WORD BOUNDARY
022774  120461  001442                  CMPB    R4,TR0(R1)      ;IS THE CHARACTER WHAT IT SHOULD BE?
023000  001413                          BEQ     4$              ;IF SO,GO CONTINUE PROCESSING
023002  116105  001442                  MOVB    TR0(R1),R5      ;GET WHAT WAS EXPECTED FOR ERROR↑B PEPORTING
023006  042705  177400                  BIC     #↑C<377>,R5     ;ELIMINATE PROPAGATED SIGN
023012  042704  177400                  BIC     #↑C<377>,R4     ;ISOLATE THE ACTUAL CHARACTER
023016  010137  001372                  MOV     R1,SAVLIN       ;GET THE LINE NUMBER OF THE RECEIVER ERROR
023022  006237  001372                  ASR     SAVLIN          ;ALIGN IT CORRECTLY FOR REPORTING
023026  104005                          ERROR   5               ;*DATA ERROR
023030  005261  001442          4$:     INC     TR0(R1)         ;SET UP THE NEXT EXPECTED CHARACTER
023034  005203                          INC     R3              ;INCREMENT THE COUNT OF RECEIVED CHARACTERS
023036  032761  000020  001442          BIT     #20,TR0(R1)     ;HAVE ALL CHARACTERS BEEN RECEIVED?
023044  001402                          BEQ     5$              ;IF NOT, GO RECEIVE SOME MORE
023046  020203                          CMP     R2,R3           ;HAVE WE RECEIVED ALL CHARACTERS?
023050  001401                          BEQ     6$              ;IF SO,GO DETERMINE THE TIMING
023052  000002                  5$:     RTI                     ;GO CONTINUE TIMING AND ALLOW INTERRUPTS
023054  004737  022676          6$:     JSR     PC,BUILD        ;GET THE RELATIVE TIME (SIGNIFICANT BITS)

023060  013700  023140                  MOV     OFFSET,R0       ;GET POINTER
023064  013760  001220  002102          MOV     $TMP1,TMTBL(R0) ; SAVE THIS TEST'S TIME
023072  005737  023140                  TST     OFFSET          ;FIRST TEST?
023076  001414                          BEQ     7$              ;IF NOT, GO CHECK THE TIME
023100  005740                          TST     -(R0↑B)         ;POINT TO THE PREVIOUS  TIME TAKEN
023102  026037  002102  001220          CMP     TMTBL(R0),$TMP1 ;IS THIS TIME WHAT IT SHOULD BE?
023110  101007                          BHI     7$              ;IF SO, GO TO THE NEXT TEST
023112  016005  002102                  MOV     TMTBL(R0),R5    ;PLACE WHAT WAS EXPECTED IN R5
023116  010137  001372                  MOV     R1,SAVLIN       ;GET THE LINE NUMBER OF THE RECEIVER
023122  006237  001372                  ASR     SAVLIN          ;MAKE SURE IT'S THE LINE NUMBER
023126  104021                          ERROR   21              ;TIMING ERROR
023130  042777  000140  156704  7$:     BIC     #RIE!MSENAB,@DZCSR ;DISABLE THE DEVICE
023136  000002                          RTI                     ;RETURN TO THE PROGRAM
023140  000000                  OFFSE↑BT:       0
```

```
                                        ;DZ11 ECHO/CABLE TEST
                                        ;COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                                ;*STARTING PROCEDURE
                                                ;*LOAD PROGRAM
                                                ;*LOAD ADDRESS 000210
                                                ;*PRESS START
                                                ;*PROGRAM WILL TYPE DZ11 ECHO/CABLE TEST
                                                ;*PROGRAM WILL TYPE  WHICH TEST- ECHO OR CABLE
                                                ;*TYPE IN  E  OR  C   RESPECTIVELY
                                                ;*PROGRAM WILL TYPE "VECTOR ADDRESS-"
                                                ;*TYPE IN THE ADDRESS OF THE RECEIVER INTERRUPT VECTOR
                                                ;*FOR THE DZ11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
                                                ;*PROGRAM WILL TYPE "CONTROL REGISTER ADDRESS-"
                                                ;*TYPE IN THE ADDRESS OF THE SYSTEM CONTROL REGISTER
                                                ;*FOR THE DZ11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
                                                ;*PROGRAM WILL TYPE "LINE NUMBER-"
                                                ;*TYPE IN THE LINE NUMBER TO BE TESTED (IN OCTAL)
                                                ;*,FOLLOWED BY <CARRIAGE RETURN>
                                                ;*PROGRAM WILL TYPE "BAUD RATE-"
                                                ;*TYPE IN THE BAUD RATE OF THE DZ11 TERMINAL
                                                ;*,FOLLOWED BY <CARRIAGE RETURN>
                                                        ;*T↑BHE FOLLOWING BAUD RATES ARE ACCEPTED IN DECIMAL
                                                ;*              50
                                                ;*              75
                                                ;*              110
                                                ;*              135      (ROUNDED OFF   134.5)
                                                ;*              150
                                                ;*              300
                                                ;*              600
                                                ;*              1200
                                                ;*              1800
                                                ;*              2000
                                                ;*              2400
                                                ;*              3600
                                                ;*              4800
                                                ;*              7200
                                                ;*              9600
                                                ;*ALL OTHERS ARE REJECTED

                                                ;*PROGRAM WILL TYPE "ECHO"  OR  "CABLE TEST" TO INDICATE THAT TESTING HAS STARTE


                                                ;PROGRAM INITIALIZATION
                                                ;LOCK OUT INTERRUPTS
                                                ;SET UP ↑BPROCESSOR STACK
                                                ;SET UP POWER FAIL VECTOR
                                                ;CLEAR PROGRAM FLAGS AND COUNTS

023142  012706  001120          XSTART: MOV     #STACK,SP       ;SET UP PROCESSOR STACK
023146  106427  000340                  MTPS    #PR7            ;LOCK OUT INTERRUPTS
023152  012737  023142  001126          MOV     #XSTART,$LPADR  ;SET UP IN CASE OF POWER FAIL
023160  005037  025322                  CLR     STFLG           ;CLEAR TEST START FLAG
023164  005037  001242                  CLR     $PASS           ;CLEAR PASS COUNT
023170  005037  001132                  CLR     $ERTTL          ;CLEAR ERROR COUNT
023174  105037  001123                  CLRB    $ERFLG          ;CLEAR ERROR FLAG
```

```
023200  005037  025326                   CLR     LAST               ;CLEAR LAST ERROR PC
023204  032777  000001  155746  VEC1:    BIT     #SW00,@SWR         ;IF SW00=1, GET NEW VECTOR
023212  001465                           BEQ     OTHER              ;AND CSR
023214  012701  000300          VEC2:    MOV     #300,R1
023220  012702  000302                   MOV     #302,R2
023224  010221                  1$:      MOV     R2,(R1)+           ;RESTORE TRAPCATCHER
023226  005022                           CLR     (R2)+              ;IN FLOATING VECTOR AREA
023230  022122                           CMP     (R1)+,(R2)+        ;UPDATE THE POINTERS
023232  020127  001000                   CMP     R1,#1000
023236  001372                           BNE     ↑B1$
023240  104403                           INSTR                      ;INPUT ADDRESS OF DEVICE VECTOR
023242  025354                           MVECTOR                    ;MESSAGE "VECTOR ADDRESS-"
023244  104405                           PARAM                      ;CONVERT STRING TO OCTAL
023246  000300                           300                        ;LOW LIMIT
023250  000770                           770                        ;HIGH  LIMIT
023252  002072                           DZRIV                      ;LOCATIONS TO BE FILLED
023254     003                  .BYTE    3                          ;LSB MASK
023255     004                  .BYTE    4                          ;NUMBER OF LOCATIONS
023256  104403                           INSTR                      ;INPUT ADDRESS OF DEVICE CSR
023260  025376                           MREGAD                     ;MESSAGE "CONTROL REGISTER ADDRESS-"
023262  104405                           PARAM                      ;CONVERT STRING TO OCTAL
023264  1600↑B00                                 160000
                                                                              ;LOW LIMIT
023266  163700                           163700                     ;HIGH LIMIT
023270  002042                           DZCSR                      ;LOCATIONS TO BE FILLED
023272     007                  .BYTE    7                          ;LSB MASK
023273     001                  .BYTE    1                          ;NUMBER OF LOCATIONS
023274  013737  002042  002046           MOV     DZCSR,DZRBUF       ;BEGIN BUILDING DEVICE ADDRESSES
023302  062737  000002  002046           ADD     #2,DZRBUF          ;FORM THE READ BUFFER ADDRESS
023310  013737  002046  002052           MOV     DZRBUF,DZLPR       ;REMEMBER THAT THIS IS ALSO LINE PARAMETER REG.
023316  013737  002046  002056           MOV     DZRBUF,DZTCR       ;BEGIN BUILDING TRANSMITTER CONTROL REGISTER
023324  062737  000002  002056           ADD     #2,DZTCR           ;FORM THE TRANSMITTER CONTROL REGISTER POINTER
023332  013737  002056  002060           MOV     DZTCR,HDZTCR
023340  005237  002060                   INC     HDZTCR
023344  013737  002056  002066           MOV     DZTCR,DZTDR        ;BEGIN FORMING TRANSMITTER DATA REGISTER
023352  062737  000002  002066           ADD     #2,DZTDR           ;FORM THE TRANSMITTER DATA REGISTER
023360  013737  002066  002062           MOV     DZTDR,DZMSR
023366  032777  000002  155564  OTHER:   BIT     #SW01,@SWR         ;RESELECT OF TEST?
023374  001427                           BEQ     XBEGIN             ;IF NOT, SKIP ASKING WHICH ONE
023376  104403                           INSTR                      ;INPUT WHICH TEST YOU ARE RUNNING
023400  025562                           MWHICH                     ;ECHO OR CA↑BBLE
023402  104416                           PAWCH                      ;SET FLAG
023404  025320                           WCHFLG                     ;THIS FLAG
023406  104403                  BAUD:    INSTR                      ;INPUT BAUD RATE
023410  025504                           MSPEED                     ;MESSAGE "BAUD RATE-"
023412  104415                           PARMD                      ;CONVERT DECIMAL STRING TO OCTAL
023414  000062                           50.                        ;LOW LIMIT
023416  022600                           9600.                      ;HIGH LIMIT
023420  025336                           LINESP                     ;LOCATION TO BE FILLED
023422     000                  .BYTE    0                          ;LSB MASK
023423     001                  .BYTE    1                          ;NUMBER OF LOCATIONS
023424  104413                  LINEX:   DEVICE.CLR                 ;CLEAR DEVICE
023426  005037  025322                   CLR     STFLG              ;CLEAR PROGRAM START FLAG
↑B              023432  104403                   INSTR
                                                                     ;INPUT LINE NUMBER
023434  025474                           MLINE                      ;MESSAGE "LINE NUMBER-"
023436  104405                           PARAM                      ;CONVERT  STRING TO OCTAL
023440  000000                           0                          ;LOW LIMIT
```

```
 023442  000007                                  7                               ;HIGH LIMIT
 023444  001372                                  SAVLIN                          ;LOCATION TO BE FILLED
 023446     000                         .BYTE    0                               ;LSB MASK
 023447     001                                  .BYTE    1                      ;NUMBER OF LOCATIONS
 023450  004537  025124                          JSR      R5,SET

 023454  106427  000340                 XBEGIN:  MTPS     #PR7                   ;LOCK OUT INTERRUPTS
 023460  012706  001120                          MOV      #STACK,SP              ;SET UP PROCESSOR STACK
 023464  005037  025324                          CLR      LOCKUP                 ;CLEAR TIMEOUT
 023470  005737  025320                          TST      WCHFLG                 ;ECHO OR CABLE TEST ?
 023474  001413                                  BEQ      2$                     ;ECHO
 023476  012737  024176  001126                  MOV      #TEST2,$LPADR          ;CABLE TEST
 023504  005737  025322                          TST      STFLG                  ;ARE YOU LOOPING ?
 023510  001017                                  BNE      1$                     ;YES
 023512  005137  025322                          COM      STFLG                  ;NO
 023516  104402  025655                          TYPE     ,MCABLE                ;TYPE CABLE TEST
 023522  000412                                  BR       1$
 023524  012737  023554  001126         2$:      MOV      #TEST1,$LPADR          ;SET U↑BP ECHO TEST
 023532  005737  025322                          TST      STFLG                  ;ARE YOU LOOPING ?
 023536  001004                                  BNE      1$                     ;YES
 023540  005137  025322                          COM      STFLG                  ;NO
 023544  104402  025630                          TYPE     ,MTERM                 ;TYPE ECHO TEST
 023550  000177  155352                 1$:      JMP      @$LPADR                ;START TESTING
                                                 ;THIS TEST WILL ACCEPT 1 CHARACTER AT A TIME
                                                 ;(IN INTERRUPT MODE) AND TRANSMIT THAT SAME CHARACTER,
                                                 ;ONE LINE AT A TIME, ANY LINE 0 THRU 7 (OCTAL)

 023554  104413                         TEST1:   DEVICE.CLR                      ;CLEAR DZ11
 023556  012737  000001  001122                  MOV      #1,$TSTNM
 023564  013777  025344  156264                  MOV      NUMTCR,@DZTCR          ;SET T↑BCR BIT
 023572  013737  025342  001366                  MOV      NUMLIN,PAR             ;SET PARAMETERS
 023600  053737  025340  001366                  BIS      SPEED,PAR              ;SET BAUD RATE
 023606  013777  001366  156236                  MOV      PAR,@DZLPR             ;LOAD PARAM.
 023614  012777  000040  156220                  MOV      #MSENAB,@DZCSR         ;SET SCANN ENABLE
 023622  005004                                  CLR      R4
 023624  012705  025672                          MOV      #MQUICK,R5             ;SET MESSAGE BUFFER
 023630  005777  156206                 3$:      TST      @DZCSR                 ;TRDY?
 023634  100404                                  BMI      2$                     ;BR IF YES
 023636  104414                                  DELAY                           ;WAIT
 023640  005304                                  DEC      R4                     ;
 023642  001372                                  BNE      3$                     ;
 023644  104003                                  ERROR    3                      ;NO TRDY SET! WHY?
 023646  005004                         2$:      CLR      R4                     ;RESET COUNTER TO 0
 023650  112577  156212                          MOVB     (R5)+,@DZTDR           ;LOAD CHAR
 023654  001365                                  BNE      3$
 023656  004737  007242                          JSR      PC,SERV.G              ;<↑G>?
 023662  122777  000377  155270                  CMPB     #377,@SWR              ;RE-DO QUICK BROWN?
 023670  001731                                  BEQ      TEST1                  ;BR IF REPEAT PATTERN
 023672  104413                                  DEVICE.CLR
 023674  106427  000340                          MTPS     #PR7                   ;LOCK OUT INTERRUPTS
 023700  012737  024634  001360                  MOV      #XEOP,NEXT
 023706  104413                                  DEVICE.CLR
 023710  013737  025342  001366                  MOV      NUMLIN,PAR             ;SELECT LINE # & SET INTERRUPT ENABLE
 023716  053737  025340  0013↑B66                         BIS      SPEED,PAR          ;SET LINE SPEED AND
                                                                                 ;CHARACTER LENGTH (TRANS. & REC.)
 023724  052737  010000  001366                  BIS      #RCVON,PAR             ;MAKE SURE RECEIVER IS TURNED ON
```

```
023732  013777  001366  156112            MOV     PAR,@DZLPR        ;LOAD THE LINE PARAMETER REGISTER
023740  012777  024014  156124            MOV     #INTSVC,@DZRIV    ;SET UP INTERRUPT SERVICE
023746  013777  025346  156120            MOV     PRIO,@DZRIS       ;AND LEVEL
023754  106437  026220                    MTPS    @#LESS1           ;ALLOW INTERRUPTS
023760↑B        012777  000140  156054            MOV     #RIE!MSENAB,@DZCSR ;SET RECEIVER INTERRUPT ENABLE
023766  104402  025522                    TYPE    ,MCHAR            ;TYPE "ANY CHARACTER"
023772  105777  155166            1$:     TSTB    @$TKS             ;IF SOMBODY HITS A KEY- GET NEW LINE #
023776  100375                            BPL     1$                ;LOOP HERE
024000  005777  155162                    TST     @$TKB             ;CLEAR CHAR
024004  004737  007242                    JSR     PC,SERV.G         ;MAKE SURE IT WASN'T <↑G>
024010  000137  023424                    JMP     LINEX             ;



                                          ;THE FOLLOWING IS THE RECEIVER INTERRUPT SVC ROUTINE
024014  105777  156022            INTSVC: TSTB    @DZCSR            ;TEST REC. FLAG
024020  100401                            BMI     .+4
024022  104004                            ERROR   4                 ;ERROR - INTERRUPT NOT CAUSED BY FLAG
024024  017737  156016  025350            MOV     @DZRBUF,RECDAT
024032  100401                            BMI     .+4
024034  104023                            ERROR   23                ;NON- VALID CHARACTER
024036  032737  020000  025350            BIT     #BIT13,RECDAT     ;CHECK FOR FRAMING ERROR
024044  001401                            BEQ     .+4               ;BR IF NO ERROR
024046  104025                            ERROR   25                ;EITHER SOMBODY HIT THE
                                                                    ;"BREAK KEY" OR YOU HAVE AN ERROR!
024050  113737  025350  025352            MOVB    RECDAT,TBUF       ;MOVE CHARACTER TO OUTPUT AREA
024056  113737  025350  010502            MOVB    RECDAT,INBUF      ;MOVE CHARACTER TO CHECK FOR ↑C
024064  042737  17760↑B0        010502            BIC     #↑C<177>,INBUF   ;STRIP JUNK PLUS PARITY
024072  042737  174377  025350            BIC     #174377,RECDAT    ;SAVE ONLY LINE  NUMBER
024100  000337  025350                    SWAB    RECDAT
024104  023737  001372  025350            CMP     SAVLIN,RECDAT     ;DOES THE LINE # COMPARE?
024112  001401                            BEQ     .+4
024114  104015                            ERROR   15                ;#WRONG LINE NUMBER
024116  013777  025344  155732            MOV     NUMTCR,@DZTCR     ;ENABLE THE LINE TO TRANSMIT
024124  012777  000040  155710            MOV     #MSENAB,@DZCSR    ;START THE TRANSMITTERS SCANNER
024132  123727  010502  000003            CMPB    INBUF,#3                  ;IS IT A ↑C ?
024140  001004                            BNE     1$                ;NO
024142  10441↑B3                                  DEVICE.CLR
024144  012716  024634                    MOV     #XEOP,(SP)        ;CRUNCH STACK
024150  000002                            RTI
024152  013777  025344  155676    1$:     MOV     NUMTCR,@DZTCR     ;ENABLE THE LINE
024160  113777  025352  155700            MOVB    TBUF,@DZTDR       ;TRANSMIT THE CHARACTER
024166  012777  000140  155646            MOV     #RIE!MSENAB,@DZCSR        ;RESTART THE RECEIVER
024174  000002                            RTI


                                  ;THIS TEST TRANSMITS A BINARY COUNT PATTERN
                                  ;VIA INTERRUPT MODE TO THE RECEIVER
                                  ;...THE LINE UNDER TEST MUST BE TERMINATED WITH THE TEST CONNECTOR
024176  106427  000340            TEST2:  MTPS    #PR7              ;DISABLE INTERRUPTS
024202  012737  000002  001122            MOV     #2,$TSTNM
024210  012737  024634  001360            MOV     #XEOP,NEXT
024216  104413                            DEVICE.CLR                ;
                                          ;*TEST TO VERIFY THAT SETTING DTR FOR A GIVEN LINE
                                          ;*WILL BRING UP "CO" AND "RING" FOR THE SAME LINE
                                          ;*THE DIST PNL MUST HAVE JUMPER FROM DTR TO RQST TO SEND
                                          ;*IN ORDER FOR THIS TEST TO WORK!
```

```
024220  012737  024226  001362          MOV     #1$,LOCK          ;LOOP
024226  113777  025344  155624  1$:     MOVB↑B  NUMTCR,@HDZTCR    ;SET DTR
024234  005005                          CLR     R5                ;
024236  153705  025344                  BISB    NUMTCR,R5         ;BUILD EXPECTED
024242  000305                          SWAB    R5                ;PUT IN HIGH BYTE
024244  153705  025344                  BISB    NUMTCR,R5
024250  104414                          DELAY                     ;WAIT FOR CABLE DELAY
024252  017704  155604                  MOV     @DZMSR,R4         ;READY MODEM BITS
024256  020504                          CMP     R5,R4             ;ARE THEY OK?
024260  001401                          BEQ     2$                ;BR IF YES
024262  104022                          ERROR   22                ;IS THE TEST CONNECTOR ON?
                                                                  ;HAS RIGHT LINE BEEN SELECTED?
                                                                  ;IF SO- YOU HAVE A PROBLEM!
                                                                  ;MODEM BITS NOT RIGHT
024264  ↑B104401                        2$:     SCOP1                     ;LOOP
024266  104413                  3$:     DEVICE.CLR                ;INIT DZ11
024270  013737  025340  001366          MOV     SPEED,PAR         ;SET LINE SPEED
024276  053737  025342  001366          BIS     NUMLIN,PAR        ;SELECT LINE # & REC.  INTERRUPT ENABLE
024304  052737  010000  001366          BIS     #RCVON,PAR        ;ENABLE THE RECEIVER FOR THIS LINE
024312  052777  040140  155522          BIS     #TIE!RIE!MSENAB,@DZCSR ;SET TRANSMITTER INTERRUPT ENABLE
024320  012777  024434  155544          MOV     #INTREC,@DZRIV    ;SET UP INTR SERVICE
024326  013777  025346  155540          MOV     PRIO,@DZRIS       ;SET UP LEVEL
024334  012777  024614  155534          MOV     #INTRAN,@DZTIV    ;SET UP INTR SERVICE
024342  013777  025346  155530          MOV     PRIO,@DZTIS       ;SET UP LEVEL
024350  005001                          CLR     R1                ;RX DATA POINTER- SET TO 0
024352  005002                          CLR     R2                ;TX DATA POINTER- SET TO 0
024354  013777  001366  155470          MOV     PAR,@DZLPR        ;SET THE PARAMETERS AND TURN ON RECEIVER
024362  106437  026220                  MTPS    @#LESS1           ;ALLOW INTERRUPTS
024366  013777  025344  155462          MOV     NUMTCR,@DZTCR     ;SET UP TCR BIT

                                ;YOU RETURN HERE AFTER EVERY RECEIVER INTERRUPT
024374  105777  154564          SPIN:   TSTB    @$TKS             ;IF SOMEBODY HITS A KEY- GET A NEW LINE #
024400  100006                          BPL     1$                ;BR IF NO KEY HIT
024402↑B2       005777  154560                  TST     @$TKB             ;CLEAR CHAR
024406  004737  007242                  JSR     PC,SERV.G         ;MAKE SURE IT WASN'T <↑G>
024412  000137  023424                  JMP     LINEX             ;SW02=1
024416  005237  025324          1$:     INC     LOCKUP            ;INC TIMEOUT FLAG
024422  001364                          BNE     SPIN              ;IF NOT 0  RETURN SPINNING
024424  104011                          ERROR   11                ;#RECEIVER FAILED TO INTERRUPT  CHECK CABLE/TERMINATOR
024426  104413                  QUITS:  DEVICE.CLR
024430  000137  024634                  JMP     XEOP              ;CALL FOR END OF PASS
024434  005037  025324          INTREC: CLR     LOCKUP            ;CLEAR TIMEOUT FLAG
024440  105777  155376                  TSTB    @DZCSR            ;TEST REC DONE
024444  100401                          BMI↑B   .+4               ;YES
024446  104004                          ERROR   4                 ;#FALSE INTERRUPT
024450  017737  155372  025350          MOV     @DZRBUF,RECDAT    ;SAVE WORD
024456  100401                          BMI     .+4
024460  104023                          ERROR   23                ;#NON VALID CHARACTER
024462  032737  040000  025350          BIT     #BIT14,RECDAT     ;DATA OVERRUN ?
024470  001401                          BEQ     .+4               ;NO
024472  104024                          ERROR   24                ;#YES
024474  032737  020000  025350          BIT     #BIT13,RECDAT     ;FRAMING ERROR ?
024502  001401                          BEQ     .+4               ;NO
024504  104025                          ERROR   25                ;#YES
024506  032737  010000  025350          BIT     #BIT12,RECDAT     ;PARITY ERROR ?
024514  001401                          BEQ     .+4               ;NO
```

```
024516  104026                                ERROR   26                  ;*YES
024520  110105                                MOVB    R1,R5               ;SET EXPECTED
024522  042705  177400                        BIC     #↑C<377>,R5         ;CLEAR HIGH BYTE
024526  113704  025350                        MOVB    RECDAT,R4           ;GET FOUND
024532  042704  177400                        BIC     #↑C<377>,R4         ;CLEAR HIGH BYTE
024536  020504                                CMP     R5,R4    ;OK?
024540  001401                                BEQ     .+4
024542  104005                                ERROR   5                   ;DATA ERROR
024544  042737  174377  025350                BIC     #174377,RECDAT      ;SAVE ONLY LINE NUMBER
024552  00↑B0337        025350                        SWAB    RECDAT
024556  023737  001372  025350                CMP     SAVLIN,RECDAT       ;DOES THE LINE # COMPARE ?
024564  001401                                BEQ     .+4                 ;YES
024566  104015                                ERROR   15                  ;*WRONG LINE #
024570  120127  000377                        CMPB    R1,#377             ;LAST CHARACTER ?
024574  001003                                BNE     1$                  ;NO
024576  012716  024426                        MOV     #QUITS,(SP)         ;CRUNCH STACK
024602  000403                                BR      2$
024604  105201                        1$:     INCB    R1                  ;UPDATE EXPECTED DATA
024606  012716  024374                        MOV     #SPIN,(SP)          ;CRUNCH STACK
024612  000002                        2$:     RTI

024614  005777  155222                INTRAN: TST     @DZCSR   ;TEST TRANSMIT FLAG
024620  100401          ↑B                    BMI     .+4
024622  104003                                ERROR   3                   ;*FALSE INTERRUPT
024624  110277  155236                        MOVB    R2,@DZTDR           ;TRANSMIT A CHARACTER
024630  105202                                INCB    R2                  ;UPDATE TX DATA
024632  000002                                RTI     ;RETURN
```

```

                                        ;END OF PASS
                                        ;RESTART TEST

 024634  104402                 XEOP:   TYPE                            ;TYPE NAME OF TEST
 024636  025432                         MPASS
 024640  005037  025326                 CLR     LAST                    ;CLEAR LAST ERROR PC
 024644  105037  001123                 CLRB    $ERFLG                  ;CLEAR ERROR FLAG
 024650  000137  023454         RSTRT:  JMP     XBEGIN

                                        ;CONVERT DECIMAL ASCII STRING TO OCTAL
 024654  011605                 .PARMD: MOV     (SP),R5
 024656  012537  025040                 MOV     (R5)+,6$
 024662  012537  025042                 MOV     (R5)+,7$
 024666  012537  025044                 MOV     (R5)+,8$
 024672  112537  025046                 MOVB    (R5)+,9$
 024676  112537  025047                 MOVB    (R5)+,10$
 024702  010516                         MOV     R5,(SP)
 024704  005005                 2$:     CLR     R5
 024706  012704  010502                 MOV     #INBUF,R4
 024712  122714  000015                 CMPB    #15,(R4)
 024716  001424                         BEQ     3$
 024720  121427  000060         1$:     CMPB    (R4),#'0
↑B024724         002421                 BLT      3$
 024726  121427  000071                 CMPB    (R4),#'9
 024732  003016                         BGT     3$
 024734  142714  000060                 BICB    #'0,(R4)
 024740  005002                         CLR     R2
 024742  152402                         BISB    (R4)+,R2
 024744  060205                         ADD     R2,R5
 024746  122714  000015                 CMPB    #15,(R4)
 024752  001410                         BEQ     4$
 024754  006305                         ASL     R5      ;X2
 024756  010502                         MOV     R5,R2   ;SAVE X2
 024760  006305                         ASL     R5      ;X4
 024762  006305                         ASL     R5      ;X8
 024764  060205                         ADD     R2,R5   ;TIMES 10
 024766  000754                         BR      1$
 024770  104404                 3$:     INSTER
 024772  000744                         BR      2$

         ↑B                             ;TEST TO SEE IF NUMBER IS WITHIN LIMITS

 024774  020537  025042         4$:     CMP     R5,7$
 025000  101373                         BHI     3$
 025002  020537  025040                 CMP     R5,6$
 025006  103770                         BLO     3$
 025010  133705  025046                 BITB    9$,R5
 025014  001365                         BNE     3$

                                        ;STORE NUMBER AT SPECIFIED ADDRESS

 025016  013704  025044                 MOV     8$,R4
 025022  010524                 5$:     MOV     R5,(R4)+
 025024  062705  000002                 ADD     #2,R5
 025030  105337  025047                 DECB    10$
```

```
025034  001372                                BNE     5$
025036  000002                                RTI
025040  000000                        6$:     0
025042  000000                        7$:     0
025044  000000                        8$:     0
025046     000                        9$:     .BYTE 0
025047     000                        10$:    .BYTE 0


                                              ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
                                              ;BUFFER TO THE CHARACTERS "E" AND "C".
                                              ;IF THE CHARACTER IS "E" CLEAR THE FLAG
                                              ;IF THE CHARACTER IS "C" SET THE FLAG

025050  017605  000000                .PAWCH:MOV      @(SP),R5
025054  142737  000040  010502               BICB     #40,INBUF          ;SET FOR LOWER CASE INPUT
025062  122737  00↑B0105              010502  CMPB     #'E,INBUF          ;IS IT "E" ?
025070  001002                                BNE     1$
025072  105015                                CLRB    (R5)                ;000
025074  000406                                BR      2$
025076  122737  000103  010502  1$:           CMPB    #'C,INBUF          ;IS IT "C" ?
025104  001005                                BNE     3$
025106  112715  177777                        MOVB    #-1,(R5)            ;3177
025112  062716  000002          2$:           ADD     #2,(SP)
025116  000002                                RTI
025120  104404                  3$:           INSTER                      ;RETRY
025122  000752                                BR      .PAWCH



                                ;THIS ROUTINE CONVERTS LINE SPEED (LINESP) AND
                                ;LINE NUMBER (SAVLIN) FOR DZLPR, DZTCR AND DZCSR
                                ;REGISTER USAGE.

0↑B25124        013737  001372  025342  SET:    MOV     SAVLIN,NUMLIN   ;SAVE SAVLIN
025132  013700  001372          XTCR0:  MOV     SAVLIN,R0       ;COPY THE LINE NUMBER FOR LOOP CONTROL
025136  005037  025344                  CLR     NUMTCR          ;SET A DEFAULT OF LINE 0 OR NO LINES
025142  012702  000001                  MOV     #1,R2           ;SET A BIT POINTER TO THE FIRST LINE
025146  005300                  XTCR1:  DEC     R0              ;REDUCE THE INDICATOR.IS IT MINUS YET?
025150  100402                          BMI     SET1            ;IF SO, R2 POINTS TO THE RIGHT LINE
025152  006302                          ASL     R2              ;IF NOT, MOVE THE POINTER TO THE NEXT LINE
025154  000774                          BR      XTCR1           ;GO SEE IF THIS LINE IS THE ONE
025156  012701  025220          SET1:   MOV     #TABLE2,R1
025162  010237  025344                  MOV     R2,NUMTCR       ;COPY THE CORRECT BIT POINTER
025166  022137  025336          1$:     CMP     (R1)+,LINESP
025172  001407                          BEQ     2$
025174  005721                          TST     (R1)+           ;IS IT THE END OF TABLE?
025176  001373                          BNE     1$              ;NO
025200  104402  025446                  TYPE    ,MINVAL         ;INVALID BAUD RATE,BEGIN AGAIN
025204  012705  023406                  MOV     #BAUD,R5        ;JUMP TO BAUD THRU R5
025210  000402                          BR      3$
025212  011137  025340          2$:     MOV     (R1),SPEED      ;SET UP BAUD RATE
025216  000205                  3$:     RTS     R5
```

```
                                                    ;THE FOLLOWING IS A TABLE OF LEGAL BAUD RATES (8 BITS/CHAR)
   025220  000062                       TABLE2: .WORD   50.             ;50 BAUD
   025222  010070                               .WORD   10070           ;
   025224  000113                               .WORD   75.             ;75 BAUD
   025226  010470                               .WORD   10470           ;
   025230  000156                               .WORD   110.            ;110 BAUD
   025232  011070                               .WORD   11070           ;TWO STOP BITS
   025234  000207                               .WORD   135.            ;134.5 BAUD
   025236  011470                               .WORD   11470           ;TWO STOP BITS
   025240  000226                               .WORD   150.            ;150 BAUD
   025242  012070                               .WORD   12070           ;TWO STOP BITS
   025244  000454                               .WORD   300.            ;300 BAUD
   025246  012470                               .WORD   12470           ;ONE STOP BIT
   025250  001130                               .WORD   600.            ;600 BAUD
   025252  013070                               .WORD   13070           ;ONE STOP BIT
   025254  002260                               .WORD   1200.           ;1200 BAUD
   025256  013470                               .WORD   13470           ;ONE STOP BIT
   025260  003410                               .WORD   1800.           ;1800 BAUD
   025262  014070                               .WORD   14070           ;ONE STOP BIT
   025264  003720                               .WORD   2000.           ;2000 BAUD
   025266  014470                               .WORD   14470           ;ONE STOP BIT
   025270  004540                               .WORD   2400.           ;2400 BAUD
   025272  015070                               .WORD   15070           ;ONE STOP BIT
   025274  007020                               .WORD   3600.           ;3600 BAUD
   025276  015470                               .WORD   15470           ;ONE STOP BIT
   025300  011300                               .WORD   4800.           ;4800 BAUD
   025302  016070                               .WORD   16070           ;ONE STOP BIT
   025304  016040                               .WORD   7200.           ;7200 BAUD
   025306  016470                               .WORD   16470           ;ONE STOP BIT
   025310  022600                               .WORD   9600.           ;9600 BAUD
   025312  017070                               .WORD   17070
   025314  177777  000000                       .WORD   -1,0            ;TABLE TERMINATOR


   025320  000000                       WCHFLG:0                        ;ECHO OR CABLE FLAG
   025322  000000                       STFLG: 0                        ;PROGRAM START FLAG
   025324  000000                       LOCKUP: 0                       ;TIMEOUT FLAG
   025326  000000                       LAST:   0                       ;LAST ERROR PC
   025330  000000                       TDATA:  0
   025332  000000                       RDATA:  0
   025334  000000                       BYTCNT: 0
   025336  000156                       LINESP: 110.                    ;DEFAULT BAUD RATE
   025340  006307                       SPEED:  6307                    ;DEFAULT 110 BAUD, 8 BITS/CHAR,
                                                                        ;FDX, 2 STOP BITS
   025342  000100                       NUMLIN: 100                     ;DEFAULT VALUE, REC. INTERRUPT ENABLED

   025344  000001                       NUMTCR: 1                       ;DEFAULT VALUE,TCR BIT 0
   025346  000240                       PRIO:   240                     ;DEFAULT DEVICE PRIORITY 5
   025350  000000                       RECDAT: 0
   025352  000000                       TBUF:   0
   025354  053200  041505  047524  MVECTO: .ASCIZ  <200>/VECTOR ADDRESS- /
   025376  041600  047117  051121↑B4        MREGAD: .ASCIZ  <200>/CONTROL REGISTER ADDRESS- /
   025432  050200  051501  020123  MPASS:  .ASCIZ  <200>/PASS DONE./
   025446  044600  053116  046101  MINVAL: .ASCIZ  <200>/INVALID BAUD RATE - /
   025474  046200  047111  035105  MLINE:  .ASCIZ  <200>/LINE: /
   025504  041200  052501  020104  MSPEED: .ASCIZ  <200>/BAUD RATE - /
```

```
025522  052200  050131  020105  MCHAR:  .ASCIZ  <200>/TYPE A CHAR, ON DZ11 TERMINAL /
025562  053600  044510  044103  MWHICH: .ASCIZ  <200>/WHICH TEST ? ECHO OR CABLE (E OR C) /
025630  052200  051105  044515  MTERM:  .ASCIZ  <200>/TERMINAL ECHO TEST /
025655     200  040503  046102  MCABLE: .ASCIZ  <200>/CABLE TEST /
025672  006777  177777  177412  MQUICK: .ASCII  <377><15><377><377><12><377><377>
025701     124  042510  050440          .ASCII  /THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789/
025776  006777  177777  177412          .ASCII  <377><15><377><377><12><377><377><377><0>
        026010                  .EVEN
                                ;;************************************************************
                                        ;UTILITIES
                                ;;************************************************************

                                ;THIS UTILITY CALCULATES PRIORITY LEVEL,SE↑BTS UP CSR'S,SETS UP VECTORS.
026010  006337  026216          DZLEV:  ASL     DZPRT           ;BUILD PRIORITY IN THIS LOCATION
026014  006337  026216                  ASL     DZPRT           ;USING ARITHMETIC SHIFTS, ROTATE
026020  006337  026216                  ASL     DZPRT           ;       THE PRIORITY LEVEL PAST
026024  006337  026216                  ASL     DZPRT           ;       THE BIT POSITIONS CORRE-
026030  006337  026216                  ASL     DZPRT           ;       SPONDING TO THE CONDITION CODES
026034  013737  026216  026220          MOV     DZPRT,LESS1     ;MOVE THIS TO LESS1
026042  162737  000001  026220          SUB     #1,LESS1        ;CREATE THE NEXT LOWEST PRIORITY
026050  042737  000037  026220          BIC     #37,LESS1       ;INSURE THAT THE  TNZVC BITS ARE CL↑BEAR
026056  013700  002072                  MOV     DZRIV,R0        ;PLACE THE BASE VECTOR ADDRESS IN R0
026062  062700  000002                  ADD     #2,R0           ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
026066  010037  002074                  MOV     R0,DZRIS        ;STORE IT HERE
026072  062700  000002                  ADD     #2,R0           ;CALCULATE THE TRANSMITTER INTERRUPT VECTOR
026076  010037  002076                  MOV     R0,DZTIV        ;STORE IT HERE
026102  062700  000002                  ADD     #2,R0           ;CALCULATE THE TRANSMITTER VECTOR STATUS ADDRESS
026106  010037  002100                  MOV     R0,DZTIS        ;STORE IT HERE

                                ;THIS SEGMENT SETS UP POINTERS FOR THE GIVEN DZ11. $BASE IS THE BASE ADDRESS
                                ;OF THE DEVICE
026112  013700  001310                  MOV     $BASE,R0        ;COPY THE ADDRESS BEING LOADED
026116  010037  002042                  MOV     R0,DZCSR        ;XXX0
026122  005200                          INC     R0
026124  010037  002044                  MOV     R0,HDZCSR       ;XXX1
026130  005200                          INC     R0
026132  010037  002046                  MOV     R0,DZRBUF       ;XXX2
026136  010037  002052                  MOV     R0,DZLPR        ;XXX2
026142  005200                          INC     R0
026144  010037  002050                  MOV     R0,HDZRBUF      ;XXX3
026150  010037  002054                  MOV     R0,HDZLPR       ;XXX3
026154  005200                          INC     R0
026156  010037  002056                  MOV     R0,DZTCR        ;XXX4
026162  005200                          INC     R0
026164  010037  002060                  MOV     R0,↑BHDZTCR     ;XXX5
026170  005200                          INC     R0
026172  010037  002062                  MOV     R0,DZMSR        ;XXX6
026176  010037  002066                  MOV     R0,DZTDR        ;XXX6
026202  005200                          INC     R0
026204  010037  002064                  MOV     R0,HDZMSR       ;XXX7
026210  010037  002070                  MOV     R0,HDZTDR       ;XXX7
026214  000207                          RTS     PC
026216  000240                  DZPRT:  PR5
026220  000200                  LESS1:  PR4     ;LEVEL TO ALLOW INTERRUPTS
```

```
                                         ;ERROR ERROR TABLE
 026222  000000                    .ERRTAB:          0↑B       ;ERROR 0
 026224  000000                                      0
 026226  000000                                      0

 026230  026434                                      EM1       ;ERROR
 026232  027634                                      DH1
 026234  030032                                      DT1

 026236  026507                                      EM2       ;ERROR 2
 026240  027657                                      DH2
 026242  030044                                      DT2

 026244  026535                                      EM3       ;ERROR 3
 026246  027712                                      DH3
 026250  030062                                      DT3

 026252  026574                                      EM4       ;ERROR 4
 026254  027712                                      DH3
 026256  030062                                      DT3

 026260  026623                                      EM5       ;ERROR 5
 026262  027724                                      DH4
 026264  030070                                      DT4

 026266  026652                                      EM6       ;ERROR 6
 026270  027724                                      DH4
 026272  030070                                      DT4

 026274  026710                                      EM7       ;ERROR 7
 026276  027712                                      DH3
 026300  030062                                      DT3

 026302  026751                                      EM8       ;ERROR 10
 026304  027712                                      DH3
 026306  030062                                      DT3

 026310  027013                                      EM9       ;ERROR 11
 026312  027712                                      DH3
 026314  030062                                      DT3

 026316  027051                                      EM10      ;ERROR 12
 026320  027712                                      DH3
 026322  030062                                      DT3

 026324  027110                                      EM13      ;ERROR 13
 026326  027712                                      DH3
 026330  030062                                      DT3

 026332  027141                    ↑B                EM14      ;ERROR 14
 026334  027712                                      DH3
 026336  030062                                      DT3

 026340  027173                                      EM15      ;ERROR 15
 026342  000000                                      0
 026344  000000                                      0
 4424
 4425  026346  027235                              EM16
 4426  026350  027712                              DH3
 4427  026352  030062                              DT3
 4428
 4429  026354  027306                              EM17     ;ERROR 17
 4430  026356  027712                              DH3
 4431  026360  030062                              DT3
 4432
 4433  026362  027344                              EM20
 4434  026364  027712                              DH3
 4435  026366  030062              ↑B              DT3
 4436
 4437  026370  027405                              EM21     ;ERROR 21
 4438  026372  027753                              DH5
 4439  026374  030106                              DT5
 4440
 4441  026376  027435                              EM22     ;ERROR 22
 4442  026400  027724                              DH4
 4443  026402  030070                              DT4
 4444
 4445  026404  027477                              EM23     ;ERROR 23
 4446  026406  027712                              DH3
 4447  026410  030062                              DT3
 4448
 4449  026412  027527                              EM24
 4450  026414  027712                              DH3
 4451  026416  030062                              DT3
 4452
 4453  026420  027555                              EM25
 4454  026422  027712                              DH3
 4455  026424  030062                              DT3
 4456
 4457  026426  027605                              EM26
 4458  026430  027712                              DH3
 4459  026432  030062                              DT3
 4460                                                    ;ERROR MESSAGES
 4461    026434  047200  020117  046123  EM1:     .ASCIZ  <200>/NO SLAVE SYNC RESPONSE FROM DZ11 REGISTER/
         026507     200  042522  044507  EM2:     .ASCIZ  <200>?REGISTER R/W FAILURE?
         026535     200  051124  047101  EM3:     .ASCIZ  <200>/TRANSMIT READY (TRDY) NOT SET/
         026574  051200  041505  044505  EM4:     .ASCIZ  <200>/RECEIVER DONE NOT SET/
         026623     200  040504  040524  EM5:     .ASCIZ  <200>/DATA COMPARISON ERROR/
         026652  042200  030532  020061  EM6:     .ASCIZ  <200>/DZ11 *RECEIVER BUFFER* ERR↑BOR/
         026710  052200  040522  051516  EM7:     .ASCIZ  <200>/TRANSMITTER FAILED TO INTERRUPT/
         026751     200  047125  054105  EM8:     .ASCIZ  <200>/UNEXPECTED TRANSMITTER INTERRUPT/
         027013     200  042522  042503  EM9:     .ASCIZ  <200>/RECEIVER FAILED TO INTERRUPT/
         027051     200  047125  054105  EM10:    .ASCIZ  <200>/UNEXPECTED RECEIVER INTERRUPT/
         027110  051600  046111  020117  EM13:    .ASCIZ  <200>/SILO ALARM SET TOO SOON/
         027141     200  044523  047514  EM14:    .ASCIZ  <200>/SILO ALARM FAILED TO SET/
         027173     200  041501  044524  EM15:    .ASCIZ  <200>/ACTION DETECTED ON INVALID LINE./
         027235     200  042522  042101  EM16:    .ASCIZ  <200>/READING DZRBUF DID NOT CLEAR SILO ↑BALARM/
         027306  042200  052101  020101  EM17:    .ASCIZ  <200>/DATA VALID SHOULD NOT BE SET/
         027344  051200  041505  044505  EM20:    .ASCIZ  <200>/RECEIVER DONE SHOULD NOT BE SET/
         027405     200  042522  040514  EM21:    .ASCIZ  <200>/RELATIVE TIMING ERROR./
         027435     200  047515  042504  EM22:    .ASCIZ  <200>/MODEM SIGNAL ERROR ON CABLE TEST/
         027477     200  040504  040524  EM23:    .ASCIZ  <200>/DATA VALID IS NOT SET!/
         027527     200  040504  040524  EM24:    .ASCIZ  <200>/DATA OVERRUN IS SET!/
         027555     200  051106  046501  EM25:    .ASCIZ  <200>/FRAMING ERROR OCCURRED/
         027605     200  040520  044522  EM26:    .ASCIZ  <200>/PARITY ERROR OCCURRED/

         027634  052200  040522  020120  DH1:     .ASCIZ  <200>/TRAP PC  DZ11 REG/
         027657     200  054105  042520  DH2:     .ASCIZ  <200>/EXPECTED  FOUND  REGISTER/
         027712  046200  047111  020105  DH3:     .ASCIZ  <200>/LINE NO./
         027724  042600  050130  041505  DH4:     .ASCIZ  <200>/EXPECTED  FOUND  LINE/
         027753     200  054124  046040  DH5:     .ASCIZ  <200>/TX LINE PREVIOUS TIME  ACTUAL TIME  PARAMETER/

                                         .EVEN
                                                  ;DATA TABLES FOR ERROR MESSAGES
         030032  000002                  DT1:     2
         030034     006     003                   .BYTE    6,3
         030036  001204                           $REG1
         030040     006     001                   .BYTE    6,1
         030042  001202                           $REG0

         030044  000003                  DT2:     3
         030046     006     004                   .BYTE    6,4
         030050  001214                           $REG5
         030052     00↑B6           001                     .BYTE    6,1
         030054  001212                           $REG4
         030056     006     001                   .BYTE    6,1
         030060  001202                           $REG0

         030062  000001                  DT3:     1
         030064     003     001                   .BYTE    3,1
         030066  001372                           SAVLIN

         030070  000003                  DT4:     3
         030072     006     004                   .BYTE    6,4
         030074  001214                           $REG5
         030076     006     001                   .BYTE    6,1
         030100  001212                           $REG4
         030102     003     001                   .BYTE    3,1
```

```
        030104  001372                      SAVLIN

        030106  000004              DT5:    4
        030110     003     005              .BYTE   3,5
        030112  001372              ↑B      SAVLIN
        030114     006     011              .BYTE   6,9.
        030116  001214                      $REGS
        030120     006     007              .BYTE   6,7
        030122  001220                      $TMP1
        030124     006     001              .BYTE   6,1
        030126  001400                      REGIST
                                                    ;TABLE OF DELAY TIMES FOR INDIVIDUAL BAUD RATES
                                                    ;-------------------------------------------------

        030130  002450              DLYTBL: 2450                    ;TIME FOR   50 BAUD
        030132  001560                      1560                    ;TIME FOR   75 BAUD
        030134  001120                      1120                    ;TIME FOR  110 BAUD
        030136  000750                       750                    ;TIME FOR  134 BAUD
        030140  000660                       660                    ;TIME FOR  150 BAUD
        030142  000330                       330                    ;TIME FOR  300 BAUD
        030144  000150                       150                    ;TIME FOR  600 BAUD
        030146  000060                        60                    ;TIME FOR 1200 BAUD
        030150  000040                        40                    ;TIME FOR 1800 BAUD
        030152  000030                        30                    ;TIME FOR 2000 BAUD
        030154  000020                        20                    ;TIME FOR 2400 BAUD
        030156  000010                        10                    ;TIME FOR 3600 BAUD
        030160  000001                         1                    ;TIME FOR 4800 BUAD
        030162  000001                         1                    ;TIME FOR 7200 BAUD
        030164  000001                         1                    ;TIME FOR 9600 BAUD
        030166  000001                         1                    ;TIME OF DELAY FOR 19200 BAUD

                                            ;DELAYS WERE COMPUTED TO ALLOW MAXIMUM TIME AT EACH BAUD RATE
                                            ;FOR ALL TESTS TO FUNCTION CORRECTLY ON A PDP11/45 WITH BIPOLAR
                                            ;MEMORY. THE TIMES WERE ALSO TESTED ON AN 11/40 AND 11/10.

        030170                      C↑BORMAX:
                000001              .END
```

```
MD-11-DZDZA-C   MACY11 27(1006)  21-OCT-76  13:09  PAGE 93                                    PAGE:  0115
DZDZAC.P11    21-OCT-76 13:07             SYMBOL TABLE

ABASE = 160010        AUTO.S  011462        C05   = 020000        DZCR4   001560          EIGHTS= 000070
ACDW1 = 000000        AVECT = 000300        C06   = 040000        DZCR5   001574          EMTVEC= 000030
ACDW2 = 000000        AVECT1= 000000        C07   = 100000        DZCR6   001610          EM1     026434
ACPUOP= 000000        AVECT2= 000000        CR    = 000015        DZCR7   001624          EM10    027051 ↑B
ACTIVE  001412        BAUD    023406        CRLF  = 000200        DZCSR   002042          EM13    027110
ADDW0 = 000000        BINWRD  006510        CSRMAP  011470        DZLEV   026010          EM14    027141
ADDW1 = 000000        BIT0  = 000001        CYCLE   010752        DZLPR   002052          EM15    027173
ADDW10= 000000        BIT00 = 000001        DATABP  007102        DZLV0   001504          EM16    027235
ADDW11= 000000        BIT01 = 000002        DATAHD  007070        DZLV1   001520          EM17    027306
ADDW12= 000000        BIT02 = 000004        DCLASM= 104417        DZLV10  001644          EM2     026507
ADDW13= 000000        BIT03 = 000010        DCLR  = 000020        DZLV11  001660          EM20    027344
ADDW14= 000000        BIT04 = 000020        DDISP = 177570        DZLV12  001674          EM21    027405
ADDW15= 000000        BIT05 = 000040        DELAY = 104414        DZLV13  001710          EM22    027435
ADDW2 = 000000        BIT06 = 000100        DEVADR  006232        DZLV14  001724     ↑B     EM23    027477
ADDW3 = 000000        BIT07 = 000200        DEVICE= 104413        DZLV15  001740          EM24    027527
ADDW4 = 000000        BIT08 = 000400        DH1     027634        DZLV16  001754          EM25    027555
ADDW5 = 000000        BIT09 = 001000        DH2     027657        DZLV17  001770          EM26    027605
ADDW6 = 000000        BIT1  = 000002        DH3     027712        DZLV2   001534          EM3     026535
ADDW7 = 000000        BIT10 = 002000        DH4     027724        DZLV3   001↑B550          EM4     026574
ADDW8 = 000000        BIT11 = 004000        DH5     027753        DZLV4   001564          EM5     026623
ADDW9 = 000000        BIT12 = 010000        DISPLA  001162        DZLV5   001600          EM6     026652
ADEVCT= 000000        BIT13 = 020000        DISPRE  000174        DZLV6   001614          EM7     026710
ADEVM = 000000        BIT14 = 040000        DLYCNT  006604        DZLV7   001630          EM8     026751
ADRCNT  006235        BIT15 = 100000        DLYTBL  030130        DZMSR   002062          EM9     027013
ADVANC= 104400        BIT2  = 000004        DONFLG  001420        DZNUM   001410          ERRMSG  007056
AENV  = 000000        BIT3  = 000010        DSWR  = 177570        DZPRT   026216          ERRVEC= 000004
AENVM = 000000        BIT4  = 000020        DTR0  = 000400        DZRBUF  002046          ERTAB0  007226
AFATAL= 000000        BIT5  = 000040        DTR1  = 001000        DZRIS   002074          EVEPAR= 000000
AMADR1= 000000        BIT6  = 000100        DTR2  = 002000   ↑B     DZRIV   002072          EXITER  007162
AMADR2= 000000        BIT7  = 000200        DTR3  = 004000        DZTCR   002056          FINI    020770
AMADR3= 000000        BIT8  = 000400        DTR4  = 010000        DZTDR   002066          FIVE  = 000000
AMADR4= 000000        BIT9  = 001000        DTR5  = 020000        DZTIS   002100          FIVES = 000040
AMAMS1= 000000        BPTVEC= 000014        DTR6  = 040000        DZTIV   002076          FRMERR= 020000
AMAMS2= 000000        BRK0  = 000400        DTR7  =↑B 100000        DZVC0   001502          HALTS   007106
AMAMS3= 000000        BRK1  = 001000        DT1     030032        DZVC1   001516          HDRFLG  001416
AMAMS4= 000000        BRK2  = 002000        DT2     030044        DZVC10  001642          HDZCSR  002044
AMSGAD= 000000        BRK3  = 004000        DT3     030062        DZVC11  001656          HDZLPR  002054
AMSGLG= 000000        BRK4  = 010000        DT4     030070        DZVC12  001672          HDZMSR  002064
AMSGTY= 000000        BRK5  = 020000        DT5     030106        DZVC13  001706          HDZRBU  002050
AMTYP1= 000000        BRK6  = 040000        DVALID= 100000        DZVC14  001722          HDZTCR  002060
AMTYP2= 000000        BRK7  = 100000        DZACTV  001404        DZVC15  001736          HDZTDR  002070
AMTYP3= 000000        BRW     005114        DZCR0   001500        DZVC16  001752          HILIM   006230
AMTYP4= 000000        BUILD   022676        DZCR1   001514        DZVC17  001766          HT    = 000011
APASS = 000000        BYTCNT  02533↑B4        DZCR10  001640        DZVC2   001532          INBUF   010502
APRIOR= 000000        CHRCNT  006506        DZCR11  001654        DZVC3   001546          INIFLG  001415
APTCSU= 000040        CNVRT = 104412        DZCR12  001670        DZVC4   001562          INIT    020250
APTENV= 000001        CONVRT= 104411        DZCR13  001704        DZVC5   001576          INIT1   020274
APTSIZ= 000200        CORMAX  030170        DZCR14  001720        DZVC6   001612          INSTER= 104404
APTSPO= 000100        C00↑B   = 000400        DZCR15  001734        DZVC7   001626          INSTR = 104403
ASWREG= 000000        C01   = 001000        DZCR16  001750        DZ.END  002000          INSTR2  006030
ATESTN= 000000        C02   = 002000        DZCR17  001764        DZ.MAP  001500          INTRAN  024614
AUNIT = 000000        C03   = 004000        DZCR2   001530        EIAFLG  001414          INTREC  024434
AUSWR = 000000        C04   = 010000        DZCR3   001544        EIGHT = 000030          INTSVC  024014
```

M09

```
MD-11-DZDZA-C   MACY11 27(1006)  21-OCT-76  13:09  PAGE 94                                              PAGE:  0116
DZDZAC.P11     21-OCT-76 13:07            SYMBOL TABLE

IOTVEC= 000020         MASTEK  010177         PAR14   001730         RL3   = 001400         SW10  = 002000
LAST    025326         MBADLN  010306         PAR15   001744         RL4   = 002000         SW11  = 004000
LESS1   026220         MCABLE  025655         PAR16   001760         RL5   = 002400         SW12  = 010000
LF    = 000012         MCHAR   025522         PAR17   001774         RL6   = 003000         SW13  = 020000
LIMITS  006156         MCSRX  ↑B 010127         PAR2    001540         RL7   = 003400         SW14  = 040000
LINE    001364         MDATA   010606         PAR3    001554         RSTART  020240         SW15  = 100000
LINESP  025336         MEPASS  007745         PAR4    001570         RSTRT   024650         SW2   = 000004
LINEX   023424         MERRPC  010254         PAR5    001604         RUN     001406         SW3   = 000010
LINE0   001506         MERRX   010154         PAR6    001620         RXISR1  022724         SW4   = 000020
LINE1   001522    ↑B     MERR2   010005         PAR7    001634         RXSVC   020542         SW5   = 000040
LINE10  001646         MERR3   010054         PAWCH = 104416         RXTCR   021016         SW6   = 000100
LINE11  001662         MINVAL  025446         PIRQ  = 177772         R6    =%000006         SW7   = 000200
LINE12  001676         MLINE   025474         PIRQVE= 000240         R7    =%000007         SW8   = 000400
LINE13  001712         MLOCK   010100         POPR0 = 012600         SAVLIN  001372         SW9   = 001000
LINE14  001726         MNEW    010202         POP1SP= 005726         SAVNUM  001411         S110  = 001000
LINE15  001742         MNTFLG  001417         POP2SP= 022626         SAVPC   001402         S1200 = 003400
LINE16  001756         MODE    001370         PRIO    025346         SAV05 = 104407         S134  = 001400
LINE17  001772         MPASS   025432         PR0   = 000000         SCOP1 = 104401         S150  = 002000
LINE2   001536         MPASSX  010143         PR1   = 000040         SERV.G  007242         S1800 = 004000
LIN↑BE3   001552          MPFAIL  007702         PR2   = 000100         SET     025124         S19200= 007400
LINE4   001566         MQUICK  025672         PR3   = 000140         SETAPT  011310         S2000 = 004400
LINE5   001602         MR      007771         PR4   = 000200         SETFLG= 104406         S2400 = 005000
LINE6   001616         MREGAD  025376         PR5   = 000240         SET.PS  010650         S300  = 002400
LINE7   001632         MSENAB= 000040         PR6   = 000300         SET1    025156         S3600 = 005400          ↑B
LOBITS  006234         MSPEED  025504         PR7   = 000340         SEVEN = 000020         S4800 = 006000
LOCK    001362         MTERM   025630         PS    = 177776         SEVENS= 000060         S50   = 000000
LOCKUP  025324         MTITLE  001000         PSW   = 177776         SILOAL= 020000         S600  = 003000
LOLIM   006226         MTSTN   010165         PUSHR0= 010046         SILOEN= 010000         S7200 = 006400
LP0   = 000000         MVECTO  025354         PUSH1S= 005746         SIX   = 000010         S75   = 000400
LP1   = 000001         MVECX   010135         PUSH2S= 024646         SIXS  = 000050         S9600 = 007000
LP2   = 000002         MWHICH  025562         PWRVEC= 000024         SNAP    020370         TABLE2  025220
LP3   = 000003         NEXT    001360         QUITS   024426         SPACNT  006507         TBITVE= 000014
LP4   = 000004         NUMLIN  025342         RCVON = 010000         SPEED   025340         TBUF    025352
LP5   = 000005         NUMTCR  025344         RDATA   025332         SPIN    024374       ↑B TCR0  = 000001
LP6   = 000006         ODDPAR= 000200         RDONE = 000200         STACK = 001120         TCR1  = 000002
LP7   = 000007         OFFSET  023140         RECDAT  025350         STFLG   025322         TCR2  = 000004
MAINT = 000010         ONESTO= 000000         REGIST  001400         STKLMT= 177774         TCR3  = 000010
MANT0   001512         OTHER   023366         RESREG  007104         STOP    001462         TCR4  = 000020
MANT1   001526         OUT     020716         RESTAR  011304         SV05    006244 ↑B        TCR5  = 000040
MANT10  001652         OVRRUN= 040000         RESVEC= 000010         SWR     001160         TCR6  = 000100
MANT11  001666         PAR     001366         RES05 = 104410         SWREG   000176         TCR7  = 000200
MANT12  001702         PARAM = 104405         RIE   = 000100         SW0   = 000001         TDATA   025330
MANT13  001716         PARAM1  006076         RING0 = 000001         SW00  = 000001         TD0     001422
MANT14  001732         PARER = 010000         RING1 = 000002         SW01  = 000002         TD1     001424
MANT15  001746         PARERR  006152         RING2 = 000004         SW02  = 000004         TD2     001426
MANT16  001762         PARITY= 000100         RING3 = 000010         SW03  = 000010         TD3     001430
MANT17  001776         PARMD = 104415         RING4 = 000020         SW04  = 000020         TD4     001432
MANT2   001542         PAR0    001510         RING5 = 000040         SW05  = 000040         TD5     001434
MANT3   001556         PAR1    001524         RING6 = 000100    ↑B     SW06  = 000100         TD6     001436
MANT4   001572         PAR10   001650         RING7 = 000200         SW07  = 000200         TD7     001440
MANT5   001606         PAR11   001664         RL0   = 000000         SW08  = 000400         TEIGHT  002140
MANT6   001622         PAR12   001700         RL1   = 000400         SW09  = 001000         TEMP    010544
MANT7   001636         PAR13   001714         RL2   = 001000         SW1   = 000002         TEST1   023554
```

N09

```
MD-11-DZDZA-C   MACY11 27(1006)  21-OCT-76  13:09  PAGE 95                                                         PAGE:  0117
DZDZ↑BAC.P11    21-OCT-76 13:07          SYMBOL TABLE

TEST2    024176        TST5     012656        $ATY4    005462        $FILLS   001175        $SCOPE   004654
TFIVE    002146        TST6     012750        $AUTOB   001154        $GDADR   001140        $SETUP=  000000
TIE    = 040000        TST7     013042        $BASE    001310        $GDDAT   001144        $SVLAD   005042
TKVEC  = 000060        TTST     004672        $BDADR   001142        $GET42   004600        $SVPC  = 000040
TLAST  = 022146        TWOSTO=  000040        $BDDAT   001146        $HD    = 000001        $SWR   = 164000
TLO    = 000000        TXSVC    020442        $CDW1    001314        $HIBTS   001462        $SWREG   001256
TL1    = 000400        TYPDAT   007072        $CDW2    001316        $ICNT    001124        $SWRMK=  000000
TL2    = 001000        TYPE   = 104402        $CHARC   005440        $ILLUP   007674        $TESTN   001240
TL3    = 001400        TYPMSG   006762        $CMTAG   001120        $INTAG   001155        $TIMES   001226
TL4    = 002000        T110     002106        $CM1   = 000006 ↑B       $ITEMB   001134          $TKB     001166
TL5    = 002400        T1200    002120        $CM2   = 000014        $LF      001232        $TKS     001164
TL6    = 003000        T134     002110        $CM3   = 000006        $LFLG    005707        $TMPO    001216
TL7    = 003400        T150     002112        $CM4   = 000004        $LPADR   001126        $TMP1    001220
TMTBL    002102        T1800    002122        $CPUOP   001262        $LPERR   001130        $TMP2    001222
TPVEC  = 000064        T2000    002124        $CRAP↑B = 177777         $MADR1   001266          $TMP3    001224
TRAPVE=  000034        T2400    002126        $CRLF    001231        $MADR2   001272        $TN    = 000035
TRDY   = 100000        T300     002114        $DDW0    001320        $MADR3   001276        $TPB     001172
TRTVEC=  000014        T3600    002130        $DDW1    001322        $MADR4   001302        $TPFLG   001177
TR0      001442        T4800    002132        $DDW10   001344        $MAIL    001234        $TPS     001170
TR1      001444        T50      002102        $DDW11   001346        $MAMS1   001264        $TSTM    001466
TR2      001446        T600     002116        $DDW12   001350        $MAMS2   001270        $TSTNM   001122
TR3      001450        T7200    002134        $DDW13   001352        $MAMS3   001274        $TYPE    005162
TR4      001452        T75      002104        $DDW14   001354        $MAMS4   001300        $TYPEC   005374
TR5      001454        T9600    002136        $DDW15   001356        $MBADR   001464        $TYPEX   005442
TR6      001456        VECMAP   012↑B026        $DDW2    001324          $MFLG    005706          $UNIT    001246
TR7      001460        VEC1     023204        $DDW3    001326        $MSGAD   001250        $UNITM   001472
TSEVEN   002142        VEC2     023214        $DDW4    001330        $MSGLG   001252        $USWR    001260
TSIX     002144        WCHFLG   025320        $DDW5    001332        $MSGTY   001234        $VECT1   001304
TST1     012216        WRDCNT   006504        $DDW6    001334        $MTYP1   001265        $VECT2   001306
TST10    013134        W↑BTBS.F  007060         $DDW7    001336          $MTYP2  001271          $XTSTR  004720
TST11    013272        XBEGIN   023454        $DDW8    001340        $MTYP3   001275        $Y     = 000020
TST12    013446        XBX      006650        $DDW9    001342        $MTYP4   001301        $$GET4=  000000
TST13    013564        XCSR     004624        $DEVCT   001244        $MXCNT   005116        .      = 030170
TST14    013650        XEOP     024634        $DEVM    001312        $N     = 000034        .ADVAN   006606
TST15    013740        XERR     004646        $DOAGN   004620        $NULL    001174        .BEGIN   004356
TST16    014024        XHEAD    010261        $E     = 000036        $NWTST=  000000        .CNVRT   006334
TST17    014214        XMTCNT   001376        $ENDAD   004610        $OVER    005060        .CONVR   006330
TST2     012406        XMTLIN   001374        $ENDCT   004574        $PASS    001242        .DCLAS   006554
TST20    014352        XMTSRV   022600        $ENV     001254        $PASTM   001470        .DELAY   006566
TST21  ↑B 014466         XPASS    004640          $ENVM    001255          $PWRAD  00767           .DEVIC  006534
TST22    015002        XSTART   023142        $EOP     004444        $PWRDN   007530        .ERRTA   026222
TST23    015330        XSTATQ   010350        $EOPCT   004566        $PWRMG   007664        .INSTE   006016
TST24    015606        XTCR0    025132        $ERFLG   001123        $PWRUP   007602        .INSTR   005712
TST25    016114        XTCR1    025146        $ERMAX   001135        $QUES    001230        .INST1   005732          ↑B
TST26    016450        XTSTN    007234        $ERROR   006620        $REGAD   001200        .MSG     005734
TST27    017102        XVEC     004632        $ERRPC   001136        $REG0    001202        .PARAM   006036
TST3     012472        XX     = 160210        $ERRTB   001360        $REG1    001204        .PARMD   024654
TST30    017630        YY     = 000500        $ERTTL   001132        $REG2    001206        .PAWCH   025050
TST31    020212        ZZ     = 000020        $ETABL   001254        $REG3    001210        .RESOS   006276
TST32    021020        $APTHD   001462        $ETEND   001360        $REG4    001212        .SAVOS   006236
TST33    021514        $ATYC    005470        $FATAL   001236        $REG5    001214        .SCOPE   004654
TST34    022146        $ATY1    005444        $FFLG    005710        $RTNAD   004622        .SCOP1   005120
TST4     012564        $ATY3    005452        $FILLC   001176        $SAVR6   007700        .SETFL   010362
```

```
.↑BSTART  002150          .TRPSR  006512          .TRPTA  002002          .TYPE   005144          .$X    = 001462

. ABS.  030170     000

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DZDZAC,DZDZAC/SOL+SYSMAC.SML[400,1066],DZDZAC.P11[400,2670]
 RUN-TIME: 57 74 2 SECONDS
 RUN-TIME RATIO: 292/135=2.1
 CORE USED:  50K  (100 PAGES)
```

C10

```
Spooler runtime 20 Seconds, 84 KCS, 544 disk reads, 3 disk writes, 118 pages
[redacted] Date 01-Dec-76 17[redacted]:42 Monitor IPC-0 [redacted]
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000111111111111111111111111111111110
000000001111111111222222222233333333334444444444555555555566666666667777777777888888888899999999990000000000111111111122222222223331 2
```