DZ11

ASYNC MULTIPLEXER TESTS

MD-11-DZDZA-D

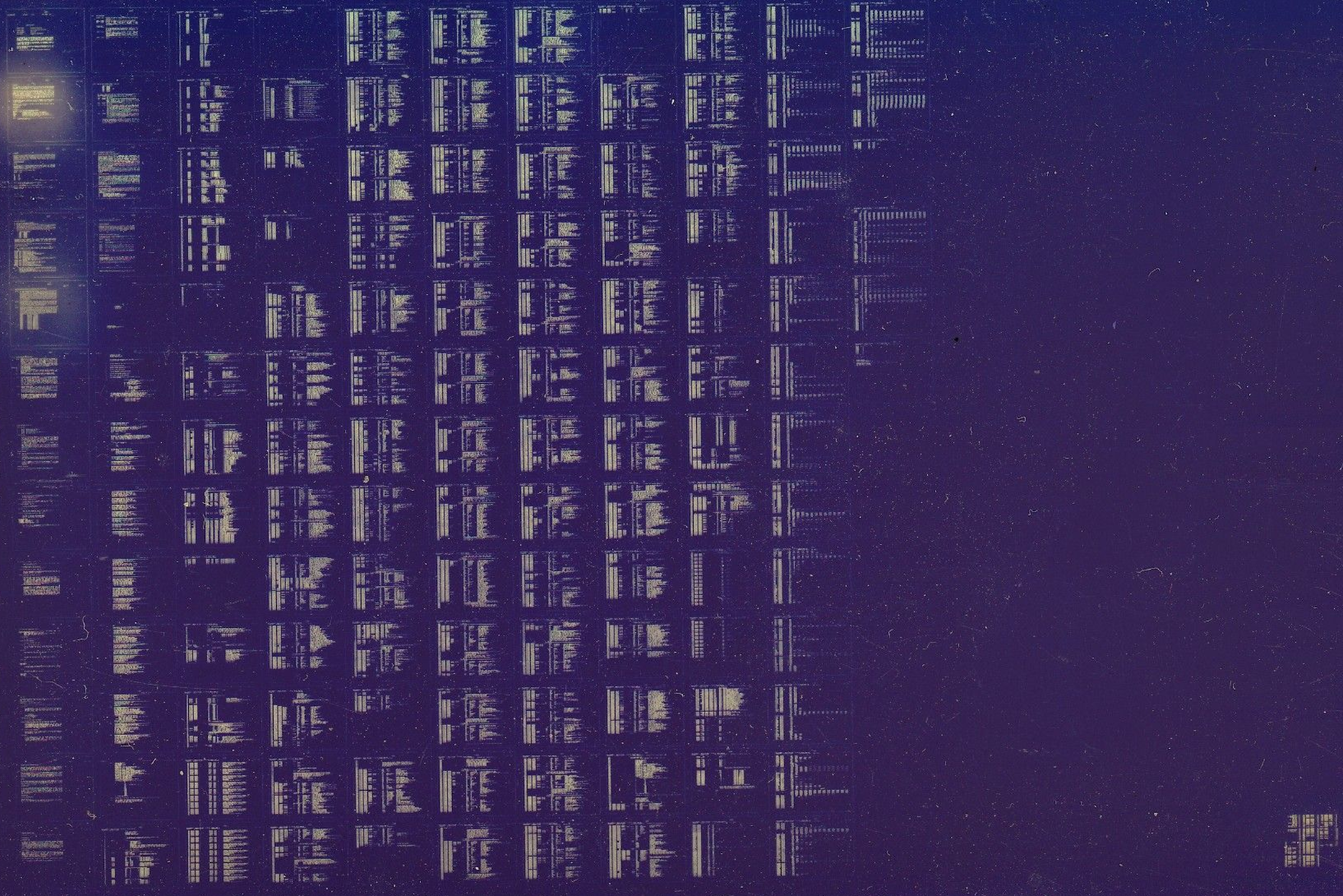
EP-DZDZA-D-DL-C

FICHE 1 OF 1

APR 1977

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZDZA-D-D
PRODUCT NAME:	DZ11 8 LINE ASYNC MUX TESTS
DATE RELEASED:	MARCH 1977
MAINTAINER:	DIAGNOSTIC ENGINEERING

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. The AUTOSIZER is designed to detect DZ11 device addresses and vectors and to determine whether the DZ11 that is detected is an EIA or 20mA board. All remaining parameters default to certain values (see SEC.8.5). Console input may be controlled at any start time through the use of SW00, SW03, SW04, and SW06 (see SEC. 4.1.1 for a detailed description of these switches).

Currently there is one standalone diagnostic (DZDZA), one system module for DEC X/11 (DZAA), and an online overlay for DZITA (ITEP) - DZDZB. (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))
H3271	Staggered turnaround connector for EIA module.
H3190	Staggered turnaround connector for 20mA module.
H325	Cable turnaround and dist pnl testing for EIA module.
H315	This may be subsituted for H325.

NOTE: A staggered turnaround connector is needed in order to test the PARITY and BREAK logic.

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 changed 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 [illegible] 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 set SW00=1 for user parameter input from console terminal. On first start if SW07=1 and SW00=0 the program will default to console parameter input (SW00=1).
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    160100
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. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. For information of status table see section 8.4 for help.

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.  If 1st start of  program after  loading the
                operator must input address and vector from console.
  SW 06   Set:  Reselect DZ11's desired active
  SW 05   Set:  Reserved
  SW 04   Set:  Select delay parameter (see SEC. 4.1.1)
  SW 03   Set:  Extra parameter input  (see SEC. 4.1.1)
  SW 02   Set:  Lock on selected test
**SW 01   Set:  Restart program at selected test
 *SW 00   Set:  Get users parameters from console
```

* For Echo or Cable tests (program started at loc. 210) this switch set to 1 allows the user to type in the Vector and the CSR for the DZ11 under test.

** For Echo or Cable test this switch set to 1 allows the selection of either the Echo or Cable test, baud rate, and the line number under test.

4.1.1 SWITCH REGISTER CONTROL OF PARAMETER INPUT FROM CONSOLE

SW 00 GET USERS PARAMETERS FROM CONSOLE. Setting this switch at start up time allows the user to input at the Console terminal the following parameters: base device address, base vector address, bus request level, declare EIA or 20mA module, mode of operation (External,Internal, or Staggered), and the number of DZ11's that are running. Using this switch alone defaults the following parameters: all 8 lines are set to be tested on each DZ11, the default baud rate is set at 19.2 kbaud, and the character length for the majority of testing is set at eight bits per character with two stop bits.

SW 03 EXTRA PARAMETER INPUT. Setting this switch at start up time provides the user with the ability to set the lines active for testing and to set the default baud rate used for the majority of the diagnostic tests. The Delay Parameter is automatically adjusted to the baud rate given by the user.

SW 04 SELECT DELAY PARAMETER. The DELAY parameter this switch controls determines the length of time the program stalls waiting for a character to be completely transmitted or received. This delay count is automatically set to provide enough delay time for the default baud rate specified when running the program on an 11/45 with bipolar memory. When running this program on a faster processor the delay parameter should be adjusted proportionally higher than the following defaulted values:

```
2450        ;time for   50 baud
1560        ;time for   75 baud
1120        ;time for  110 baud
0750        ;time for  134 baud
0660        ;time for  150 baud
0330        ;time for  300 baud
0150        ;time for  600 baud
0060        ;time for 1200 baud
0040        ;time for 1800 baud
0030        ;time for 2000 baud
0020        ;time for 2400 baud
0010        ;time for 3600 baud
0001        ;time for 4800 baud
0001        ;time for 7200 baud
0001        ;time for 9600 baud
0001        ;time for 19.2 kbaud
```

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*** 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 first 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 can't be singled out.
This switch is designed to provide an aid for a trained trouble-shooter to sample various signals on the module and is not meant to be used as a general user control switch.

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 RECOMMENDED THAT THIS SWITCH ONLY BE USED IN CONJUNCTION WITH 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. (see SEC. 4.1.1)

4.1.3 SWITCH REGISTER PRIORITIES

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'). If an '*' is printed in front of the test no. on an error report (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 the program user is technically trained to electronically isolate signal problems on the DZ11 module.
   If SW09 is not enabeled; and there is a *HARD* error (constant); SW08 is best.
2. For intermittent errors either start the program with SW01 and SW02 set which will allow the user to lock on a selected test, or else set SW14 as an error is being typed out on the terminal. SW14 will continue to loop on that test regardless of whether an error occurs.
3. SW 14 Loop on current test.

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 terminal supported by the DZ11.

NOTE: If address 000042 is non-zero the program assumes it is under 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.

```
"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.

```
"MAINTENANCE MODE
 [EXTERNAL  <H325>-EIA ONLY      (E)]
 [INTERNAL  <DZCSR03=1>          (I)]
 [STAGGERED <H3271>-EIA ONLY     (S)]
 [STAGGERED <H3190>-20mA ONLY    (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 an A325 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.  Baud rate choices are:
        "00"(  50 baud),"01"(  75 baud),"02"( 110 baud),"03"( 134 baud),
        "04"( 150 baud),"05"( 300 baud),"06"( 600 baud),"07"(1200 baud),
        "10"(1800 baud),"11"(2000 baud),"12"(2400 baud),"13"(3600 baud),
        "14"(4800 baud),"15"(7200 baud),"16"(9600 baud),"17"(19.2 kbaud)
        Low default baud rates are not suggested since they lengthen the
        time to complete a program pass dramatically.
     ***********************************

        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
        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.

Normal starting for the first time would be: LOAD ADDRESS 210; START WITH THE SWR EQUAL TO 003.

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 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 in; just load address 210 and start with SWR equal to 002. 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. Baud rate choices are: 50, 75, 110, 135, 150 300, 600 , 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600. The program will then 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 THE LINE NO. IS REQUESTED ABOVE. 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 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: type any printable character on the CONSOLE TERMINAL (not the DZ11 terminal). The program will again type "LINE: " and wait for a response.

5.4 PROGRAM AND/OR OPERATOR ACTION

The variety of program Control Switches provided in this Diagnostic Package is designed to provide the user with a wide range of trouble-shooting techniques. Before the user attempts to run this diagnostic he should become familiar with the use of these Control Switches and their restrictions. (See Sec. 4.1, 4.1.1, 4.1.2, 4.1.3)

When the program detects an error the TEST NUMBER and PC will be typed out and possibly an error message (depending on the particular error). If it is necessary to know more information concerning the error report then look in the program listing for that TEST NUMBER and then note the PC of the error report. The reason for the error report will become clearer when reading the comments in the program listing.

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 control 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 4.1.2
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=1 (INHIBIT ITERATIONS) is set to give the fastest possible execution. The actual execution time depends greatly on the PDP11 CPU configuration. An 11/40 with Core memory will take around 100 seconds to execute a pass with no iterations and about 400 seconds to execute a fully iterated pass. Any other PDP11 CPU type will execute a pass in time proportional to the execution speed of the CPU 's memory in relation to that of an 11/40.

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-D 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 for this example.

B.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. 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 DZ11 STATUS'
1500    160100
1502    000300
1504    000005
1506    000377
1510    017470
1512    000000
```

The above information will be repeated for each of up to 16 DZ11's in the system(these will follow under this table). EXPLANATION:

1500	160100	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 either EIA or 20MA. If BIT15=0 module should be an M7819, if bit15=1 module should be an M7814.
1506	000377	This is the binary representation of what lines are to be tested.
1510	017470	This is the parameter location used in most of the tests. It indicates 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 remaining parameters should be left alone. This location is used to load the DZ11 Line Parameter Register for each line. The meaning of the bits set in this location is the same as the function of the related bits in the device Line Parameter Register.
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 user; 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 'REFERENCING' 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' 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.
After identifying a DZ11 the program then attempts to set all DTR bits in Device Register 4. If any DTR bits did set the module is assumed to be an EIA module (M7819) otherwise the status map entry is set for 20mA (M7814).

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 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 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".

For all parameter 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 Mailboz-Etable'. These variables are:

$SWREG -	used if a software switch register is desired while under 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 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.

DOCUMENT
****
DZDZAD LST
****

```
2       COPYRIGHT (C) 1977
        DIGITAL EQUIPMENT CORP.
        MAYNARD, MASS. 01754

        THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
        PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.

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

27      MISCELLANEOUS DEFINITIONS

39      GENERAL PURPOSE REGISTER DEFINITIONS

51      PRIORITY LEVEL DEFINITIONS

61      "SWITCH REGISTER" SWITCH DEFINITIONS

89      DATA BIT DEFINITIONS (BIT00 TO BIT15)

117     BASIC "CPU" TRAP VECTOR ADDRESSES

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

423                             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

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

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

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

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

```
1098    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:
        1) USING A TRAP INSTRUCTION
                TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
        OR
                TYPE
                MESADR


1930    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)


2052    *********************** TEST 1 ******************************
        THIS TEST PROVES THE SLAVE SYNC RESPONSE
        DURING A READ OR WRITE TO THE FOLLOWING ADDRESS:
                DZCSR, DZRBUF, DZTCR, DZMSR

2095    *********************** TEST 2 ******************************
        THIS TEST PROVES THAT BIT "DCLR"
        CAN BE SET AND THAT IT WILL CLEAR
        BY ITSELF AFTER A PERIOD OF TIME.

2125    *********************** 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"
```

```
2157    ************************ 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"

2189    ************************ 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"

2221    ************************ 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"

2253    ************************ 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"

2285    ************************ 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

2327    ************************ TEST 11 ******************************
        THIS TESTS THAT ALL OF THE FOLLOWING
        BITS CAN BE: SET, CLEARED, CLEARED BY "RESET INSTR *NOT* DEVICE CLEAR "
        BITS TESTED ARE:

2331     DTR0, DTR1, DTR2, DTR3, DTR4, DTR5, DTR6, DTR7
        THIS TEST IS NOT DONE IF MODULE IS 20MA VERSION

2380    ************************ 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.
```

```
2417    ************************ 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

2446    ************************ TEST 14 ******************************
        THIS TEST PERFORMS RESET TESTING AND
        TESTING OF READ ONLY REGISTER DZRBUF
        AND TESTING OF WRITE ONLY REGISTER DZLPR

2472    ************************ TEST 15 ******************************
        THIS TEST PERFORMS RESET TESTING AND
        TESTING OF READ ONLY REGISTER DZMSR
        AND TESTING OF WRITE ONLY REGISTER DZTDR

2498    ************************ TEST 16 ******************************

2499    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...

2557    ************************ 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 SPECIFIED BY SWR03=1
        AND SWR00=1 AT START TIME OR ALTERING
        STATUS MAP.

2604    ************************ 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

2640    ************************ 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.

       2724   ********************** 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)
              THIS IS THE FIRST TIME THAT ALL DATA IS CHECKED

       2805   ********************** 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
              CONNECTED BY AN H325, H3271, or H3190 CONNECTOR

       2872   ********************** 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).

       2941   ********************** 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.

       3014   ********************** 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

       3124   ********************** 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.
              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.
```

```
3259    ************************ 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.

3344    ************************ TEST 31 ******************************
        THIS TEST RUNS ALL LINES FULL BORE
        BASED UPON QUALIFIED LINES
        ..THIS IS AN INTERRUPT TEST ON THE RECEIVER AND
        TRANSMITTER

3488    ************************ 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.

3587    ************************ 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!

3686    ************************ 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!
```

```
3874    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

3911    PROGRAM WILL TYPE "ECHO"  OR  "CABLE TEST" TO INDICATE THAT TESTING HAS STARTE

4097    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-D
                ;*COPYRIGHT (C) 1977
                ;*DIGITAL EQUIPMENT CORP.
                ;*MAYNARD, MASS. 01754
                ;*
                ;*
                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                ;*
000001          $TN=1
                        ;STARTING PROCEDURE
                        ;LOAD PROGRAM
                        ;LOAD ADDRESS 000200
                        ;PRESS START
                        ;PROGRAM WILL TYPE "MAINDEC-11-DZDZAD/<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 DEFINITIONS

                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1120 ***
001120          STACK=  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
177772          PIRQ=   177772                  ;;PROGRAM INTERRUPT REQUEST REGISTER
177570          DSWR=   177570                  ;;HARDWARE SWITCH REGISTER
177570          DDISP=  177570                  ;;HARDWARE DISPLAY REGISTER

                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000          R0=     %0                      ;;GENERAL REGISTER
000001          R1=     %1                      ;;GENERAL REGISTER
000002          R2=     %2                      ;;GENERAL REGISTER
000003          R3=     %3                      ;;GENERAL REGISTER
000004          R4=     %4                      ;;GENERAL REGISTER
000005          R5=     %5                      ;;GENERAL REGISTER
000006          R6=     %6                      ;;GENERAL REGISTER
000007          R7=     %7                      ;;GENERAL REGISTER
000006          SP=     %6                      ;;STACK POINTER
000007          PC=     %7                      ;;PROGRAM COUNTER

                ;*PRIORITY LEVEL DEFINITIONS
000000          PR0=    0                       ;;PRIORITY LEVEL 0
000040          PR1=    40                      ;;PRIORITY LEVEL 1
000100          PR2=    100                     ;;PRIORITY LEVEL 2
000140          PR3=    140                     ;;PRIORITY LEVEL 3
000200          PR4=    200                     ;;PRIORITY LEVEL 4
```

```
        000240                    PR5=     240              ;;PRIORITY LEVEL 5
        000300                    PR6=     300              ;;PRIORITY LEVEL 6
        000340                    PR7=     340              ;;PRIORITY LEVEL 7

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

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

```
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

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


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

        005746          PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
        005726          POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
        010046          PUSHR0=10046    ;SAVE R0 ON STACK
        012600          POPR0=12600     ;RESTORE R0 FROM STACK
        024646          PUSH2SP=24646   ;DECREMENT STACK TWICE
        022626          POP2SP=22626    ;INCREMENT STACK TWICE

                                ;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=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=BIT9          ;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 ENABLED
        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!BIT9!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 DEFINITIONS
                                        ;---------------------

        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 PRESENT 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               ↑
                                ;       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".
                                        ;-------------------------------------------------------------
                                        ;:************************************************************

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

                000010                  .=10
        000010  010766                          SET.PS                  ;FAKE "MTPS" INSTRUCTION TRAP
        000012  000340                          PR7                     ;MAKE SURE PS IS PRIORITY 7

                000020                  .=20
        000020  004772                          .SCOPE                  ;SCOPE LOOP HANDLER
        000022  000340                          PR7                     ;HANDLE AT PRIORITY 7
        000024  007646                          $PWRDN                  ;POWER FAIL HANDLER
        000026  000340                          340                     ;SERVICE AT PRIORITY LEVEL 7
        000030  006736                          $ERROR                  ;ERROR HANDLER
        000032  000340                          340                     ;SERVICE AT PRIORITY LEVEL 7
        000034  006630                          .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  004726                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                000052                          .=52
        000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
                000040                          .=$SVPC                 ;; RESTORE PC

                000174                  .=174
        000174  000000                  DISPREG:0                       ;SOFTWARE DISPLAY REGISTER 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  023364                  JMP     XSTART          ;GOTO CABLE TEST/ECHO TEST


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

```
                              .SBTTL  COMMON TAGS

                              ;;*******************************************************************
                              ;*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 USED
        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:  .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
                              ;;*******************************************************************
                              .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     ADEVCT        ;;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
        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     AVECT1        ;;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
```

```
    462  001344  000000                $DDW10: .WORD   ADDW10   ;;DEVICE DESCRIPTOR WORD#10
    463  001346  000000                $DDW11: .WORD   ADDW11   ;;DEVICE DESCRIPTOR WORD#11
    464  001350  000000                $DDW12: .WORD   ADDW12   ;;DEVICE DESCRIPTOR WORD#12
    465  001352  000000                $DDW13: .WORD   ADDW13   ;;DEVICE DESCRIPTOR WORD#13
    466  001354  000000                $DDW14: .WORD   ADDW14   ;;DEVICE DESCRIPTOR WORD#14
    467  001356  000000                $DDW15: .WORD   ADDW15   ;;DEVICE DESCRIPTOR WORD#15
    468
    469
    470  001360                        $ETEND:
    471
```

```
                                  .SBTTL  ERROR POINTER TABLE

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

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


 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                       ;COUNT 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.
```

```

                                         ;PROGRAM 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
                         ;;*****************************************************
          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:
 001462   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 REGISTER 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                      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
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  000001                      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
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 DZ11 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          DZCR16: .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 AND 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
```

B04

```
   679  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   005724                          .ADVANCE
          104401                  SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
 002004   005236                          .SCOP1
          104402                  TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
 002006   005262                          .TYPE
          104403                  INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
 002010   006030                          .INSTR
          104404                  INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
 002012   006134                          .INSTER
          104405                  PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
 002014   006154                          .PARAM
          104406                  SETFLG=TRAP+6           ;CALL TO  SET FLAG ROUTINE
 002016   010500                          .SETFLG
          104407                  SAVOS=TRAP+7            ;CALL TO REGISTER SAVE ROUTINE
 002020   006354                          .SAVOS
          104410                  RESOS=TRAP+10           ;CALL TO REGISTER RESTORE ROUTINE
 002022   006414                          .RESOS
          104411                  CONVRT=TRAP+11          ;CALL TO DATA OUTPUT ROUTINE
 002024   006446                          .CONVRT
          104412                  CNVRT=TRAP+12           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
 002026   006452                          .CNVRT
          104413                  DEVICE.CLR=TRAP+13              ;CALL TO ISSUE A DEVICE CLEAR
 002030   006652                          .DEVICE.CLR
          104414                  DELAY=TRAP+14           ;CALL TO DELAY FOR FAST CPU'S
 002032   006704                          .DELAY
          104415                  PARMD=TRAP+15           ;CONVERT DECIMAL STRING TO OCTAL
 002034   025112                          .PARMD
          104416                  PAWCH=TRAP+16           ;SET FLAG    ECHO OR  CABLE
 002036   025306                          .PAWCH
          104417                  DCLASM=TRAP+17          ;CLEAR DEVICE, SET MAINT. BIT IF I MODE
 002040   006672                          .DCLASM

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

```
                                                ;DZ11 VECTOR AND REGISTER INDIRECT POINTERS
                                                ;WORKING AREA

002042  160040                          DZCSR:  160040  ;R/W
002044  160041                          HDZCSR: 160041  ;R/W
002046  160042                          DZRBUF: 160042  ;READ ONLY
002050  160043                          HDZRBUF:        160043  ;READ ONLY
002052  160042                          DZLPR:  160042  ;WRITE ONLY
002054  160043                          HDZLPR: 160043  ;WRITE ONLY
002056  160044                          DZTCR:  160044  ;R/W
002060  160045                          HDZTCR: 160045  ;R/W
002062  160046                          DZMSR:  160046  ;READ ONLY
002064  160047                          HDZMSR: 160047  ;READ ONLY
002066  160046                          DZTDR:  160046  ;WRITE ONLY
002070  160047                          HDZTDR: 160047  ;WRITE ONLY
                                                ;DEFAULT DZ VECTORS
002072  000300                          DZRIV:  300     ;REC INTR VECTOR
002074  000302                          DZRIS:  302     ;REC INTR STATUS
002076  000304                          DZTIV:  304     ;XMIT INTR VECTOR
002100  000306                          DZTIS:  306     ;XMIT INTR STATUS
```

```
    744
    745                                         ;TIME TABLE FOR RELATIVE TIMING TESTS
    746                                         ;-------------------------------------
    747
    748  002102                          TMTBL:
    749  002102  000000                  T50:      0
    750  002104  000000                  T75:      0
    751  002106  000000                  T110:     0
    752  002110  000000                  T134:     0
    753  002112  000000                  T150:     0
    754  002114  000000                  T300:     0
    755  002116  000000                  T600:     0
    756  002120  000000                  T1200:    0
    757  002122  000000                  T1800:    0
    758  002124  000000                  T2000:    0
    759  002126  000000                  T2400:    0
    760  002130  000000                  T3600:    0
    761  002132  000000                  T4800:    0
    762  002134  000000                  T7200:    0
    763  002136  000000                  T9600:    0
    764  002140  000000                  TEIGHT:0
    765  002142  000000                  TSEVEN:   0
    766  002144  000000                  TSIX:     0
    767  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                   ;LOCK OUT INTERRUPTS
 002162  012737  007646  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     $TSTNM                 ;SET UP FOR TEST 1
 002236  012737  002150  001126                   MOV     #.START,$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$:    MOV     (SP)+,4                ;RESTORE THE BUS ERROR VECTOR
 002316  012637  000006                           MOV     (SP)+,6
 002322  105737  001415                           TSTB    INIFLG                 ;TITLE ALREADY PRINTED?
 002326  001010                                   BNE     29$                    ;BRANCH IF YES
 002330  023727  000042  004726                   CMP     @#42,#$ENDAD           ;RUNNING UNDER ACT?
 002336  001402                                   BEQ     31$                    ;IF YES DONT PRINT TITLE
 002340  104402  001000                           TYPE    ,MTITLE                ;PRINT THE DIAGNOSTIC'S TITLE
 002344  105337  001415                  31$:    DECB    INIFLG                 ;SET THE ONCE ONLY FLAG
 002350  105737  001255                  29$:    TSTB    $ENVM                  ;DETERMINE WHETHER APT SIZING SHOULD BE DONE
 002354  100006                                   BPL     30$                    ;IF NOT, GO CHECK FOR AUTO-SIZING
 002356  004737  011440                           JSR     PC,SETAPT              ;OTHERWISE, GO DO APT SIZING FROM ETABLE
 002362  105037  001416                           CLRB    HDRFLG                 ;MAKE SURE STATUS TABLE IS PRINTED
 002366  000137  004270                           JMP     16$                    ;GO PRINT DZ STATUS TABLE
 002372  032777  000001  176560          30$:    BIT     #SW00,@SWR             ;RESELECT ?
 002400  001011                                   BNE     32$                    ;IF YES, GO SET UP THE INFORMATION
 002402  122737  000377  001415                   CMPB    #377,INIFLG            ;ON 1ST START: MUST ANSWER QUESTION
 002410  001003                                   BNE     .+10                   ;IF NOT ANSWERING QUESTIONS
 002412  105777  176542                           TSTB    @SWR                   ;ARE U AUTO SIZING?
 002416  100402                                   BMI     32$                    ;NO AUTO SIZE! NO SW00=1 ON 1ST START!
 002420  000137  003114                           JMP     73$                    ;IF NO, SKIP THE INTERROGATION
 002424  012700  001500                  32$:    MOV     #DZ.MAP,R0             ;POINT TO THE BEGINNING OF THE MAP TABLE
 002430  105037  001416                           CLRB    HDRFLG                 ;MAKE SURE A MAP GETS PRINTED
 002434  005020                          65$:    CLR     (R0)+                  ;CLEAR A TABLE LOCATION
```

```
002436  020027  002000                  CMP     R0,#DZ.END          ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
002442  001374                          BNE     65$                 ;IF NOT  CLEAR THE NEXT LOCATION IN THE TABLE
002444  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

002450                          33$:
002450  104403                          INSTR                       ;CALL THE STRING INPUT ROUTINE
002452  003334                          66$                         ;POINTER TO MESSAGE TO BE PRINTED
002454  104405                          PARAM                       ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
002456  160000                          160000                      ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002460  163770                          163770                      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002462  001500                          DZCR0                       ;POINTER TO MAP LOCATION TO BE FILLED
002464     007                          .BYTE   7                   ;MASK OF INVALID BITS FOR THIS PARAMETER
002465     001                          .BYTE   1                   ;NUMBER OF PARAMETERS TO STORE
002466  013737  001500  001310          MOV     DZCR0,$BASE         ;COPY BASE ADDRESS TO ETABLE

                                        ;GET THE BASE VECTOR ADDRESS

002474                          34$:
002474  104403                          INSTR                       ;CALL THE STRING INPUT ROUTINE
002476  003400                          67$                         ;POINTER TO MESSAGE TO BE PRINTED
002500  104405                          PARAM                       ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
002502  000300                          300                         ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002504  000776                          776                         ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002506  001502                          DZVC0                       ;POINTER TO MAP LOCATION TO BE FILLED
002510     003                          .BYTE   3                   ;MASK OF INVALID BITS FOR THIS PARAMETER
002511     001                          .BYTE   1                   ;NUMBER OF PARAMETERS TO STORE
002512  013737  001502  001304          MOV     DZVC0,$VECT1        ;COPY VECTOR TO ETABLE

                                        ;GET THE BUS REQUEST LEVEL

002520  104403                          INSTR                       ;CALL THE STRING INPUT ROUTINE
002522  003441                          68$                         ;POINTER TO MESSAGE TO BE PRINTED
002524  104405                          PARAM                       ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
002526  000004                          4                           ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002530  000007                          7                           ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002532  001504                          DZLV0                       ;POINTER TO MAP LOCATION TO BE FILLED
002534     000                          .BYTE   0                   ;MASK OF INVALID BITS FOR THIS PARAMETER
002535     001                          .BYTE   1                   ;NUMBER OF PARAMETERS TO STORE
002536  113737  001504  001305          MOVB    DZLV0,$VECT1+1      ;GET BUS REQUEST LEVEL INTO ETABLE
002544  106337  001305                  ASLB    $VECT1+1            ;ALIGN THE BITS PROPERLY
002550  106337  001305                  ASLB    $VECT1+1            ;ALIGN THE BITS PROPERLY
002554  106337  001305                  ASLB    $VECT1+1            ;ALIGN THE BITS PROPERLY
002560  106337  001305                  ASLB    $VECT1+1            ;ALIGN THE BITS PROPERLY
002564  106337  001305                  ASLB    $VECT1+1            ;ALIGN THE BITS PROPERLY

                                        ;FIND OUT IF MODULE IS EIA OR 20 MA.

002570  104402  004130                  TYPE    ,74$                ;PRINT EIA MESSAGE
002574  005037  001220                  CLR     $TMP1               ;USE $TMP1
002600  105777  176360          80$:    TSTB    @$TKS               ;IS KEYBOARD DONE?
002604  100375                          BPL     80$                 ;IF NOT, WAIT FOR IT
```

```
 002606  017746  176354                  MOV     @$TKB,-(SP)        ;IF YES, PUT CHARACETR ON STACK
 002612  042716  000240                  BIC     #240,(SP)          ;STRIP DOWN CHARACTER
 002616  122726  000015                  CMPB    #15,(SP)+          ;IS IT ?
 002622  001414                          BEQ     81$                ;IF SO, GET OUT
 002624  014677  176342                  MOV     -(SP),@$TPB        ;IF NOT, PRINT CHARACTER
 002630  042737  100000  001504          BIC     #BIT15,DZLVO       ;CLEAR EIA FLAG
 002636  122726  000102                  CMPB    #102,(SP)+         ;IS IT A B?
 002642  001356                          BNE     80$                ;IF NOT, GO BACK FOR INPUT
 002644  052737  100000  001504          BIS     #BIT15,DZLVO       ;IF SO, SET FLAG
 002652  000752                          BR      80$                ;GET MORE INPUT
 002654                          81$:


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

 002654  104403                          INSTR                      ;CALL THE STRING INPUT ROUTINE
 002656  003652                          72$                        ;POINTER TO THE MESSAGE TO BE PRINTED
 002660  104406                          SETFLG                     ;CALL THE MAINTENANCE FLAG SETUP ROUTINE
 002662  001512                          MANTO                      ;THIS IS THE FLAG BEING SETUP

                                         ;GET THE NUMBER OF DZ11'S RUNNING

 002664  104403                          INSTR                      ;CALL THE STRING INPUT ROUTINE
 002666  003610                          71$                        ;POINTER TO MESSAGE TO BE PRINTED
 002670  104405                          PARAM                      ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
 002672  000001                          1                          ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
 002674  000020                          16.                        ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
 002676  001220                          $TMP1                      ;POINTER TO MAP LOCATION TO BE FILLED
 002700     000                          .BYTE   0                  ;MASK OF INVALID BITS FOR THIS PARAMETER
 002701     001                          .BYTE   1                  ;NUMBER OF PARAMETERS TO STORE

 002702  012737  000377  001506          MOV     #377,LINEO         ;SET UP DEFAULT LINES
 002710  012737  017470  001510          MOV     #17470,PARO        ;SET UP DEFAULT LPR PARAMETER
                                                                    ;RECEIVER ON; 19.2 KBAUD; 2STOP BITS; 8 BIT/CHAR
 002716  012737  000001  006722          MOV     #1,DLYCNT          ;INITIALIZE DELAY COUNT
 002724  032777  000010  176226          BIT     #SW03,@SWR         ;DO YOU WANT PARAMETERS?
 002732  001402                          BEQ     40$                ;IF NO, SKIP THE PARAMETER CALL
 002734  004737  003144                  JSR     PC,23$             ;GET PARAMETERS
 002740  012737  000001  001312  40$:    MOV     #1,$DEVM           ;INITIALIZE ACTIVE DEVICE SELECTION PARAMETER
 002746  113737  001220  001410          MOVB    $TMP1,DZNUM        ;COPY THE NUMBER OF DEVICES
 002754  113737  001220  001411          MOVB    $TMP1,SAVNUM       ;COPY A BACKUP NUMBER
 002762  005337  001220          62$:    DEC     $TMP1              ;$TMP1 CONTAINS THE COUNT OF UNINITIALIZED
 002766  001404                          BEQ     61$                ; SELECTED DEVICES
 002770  000261                          SEC                        ;SET A BIT FLAG TO INDICATE AN ACTIVE DEVICE
 002772  006137  001312                  ROL     $DEVM              ;POINT TO THE NEXT DEVICE
 002776  000771                          BR      62$                ;GO DO THIS PROCEDURE AGAIN
 003000  013737  001312  001222  61$:    MOV     $DEVM,$TMP2        ;# OF TIMES
 003006  013737  001312  001404          MOV     $DEVM,DZACTV       ;COPY THE ACTIVE DEVICE PARAMETER
 003014  012700  001500                  MOV     #DZCRO,RO          ;SET A POINTER TO THE SPECIFIED INFORMATION
 003020  012701  001514                  MOV     #DZCR1,R1          ;POINT R1 TO THE REST OF THE MAP TABLE
 003024  012702  001320                  MOV     #$DDWO,R2          ;POINT TO ETABLE'S DEVICE DESCRIPTOR WORDS
 003030  000241                          CLC                        ;INITIALIZE THE "C" BIT FOR A ROTATION
 003032  006037  001222                  ROR     $TMP2              ;SKIP MAPPING SETUP FOR DEVICE 0- IT'S DONE
 003036  006237  001222          64$:    ASR     $TMP2              ;ISOLATE A SELECTION FLAG IN THE "C" BIT
 003042  103404                          BCS     41$                ;IS THIS DEVICE SELECTED? IF YES, GO LOAD TABLE
 003044  012711  177777                  MOV     #-1,(R1)           ;TERMINATE THE LIST
 003050  000137  004244                  JMP     63$                ;GO TO THE NEXT BLOCK
```

```
003054  012011                           41$:    MOV     (R0)+,(R1)              ;ADDRESS
003056  062721  000010                           ADD     #10,(R1)+               ;POINT TO THE NEXT DZ11 ADDRESS VALUE
003062  012011                                   MOV     (R0)+,(R1)              ;VECTOR
003064  062721  000010                           ADD     #10,(R1)+               ;POINT TO THE NEXT VECTOR VALUE
003070  012021                                   MOV     (R0)+,(R1)+             ;LEVEL
003072  012021                                   MOV     (R0)+,(R1)+             ;LINES
003074  016012  177774                           MOV     -4(R0),(R2)             ;GET THE EIA FLAG FROM THE PRIORITY WORD
003100  042712  077777                           BIC     #77777,(R2)             ;ISOLATE THAT FLAG
003104  051022                                   BIS     (R0),(R2)+              ;ADD PARAMETERS TO DEVICE DESCRIPTOR WORD
003106  012021                                   MOV     (R0)+,(R1)+             ;PARAMETERS
003110  012021                                   MOV     (R0)+,(R1)+             ;MAINTENANCE MODE
003112  000751                                   BR      64$
003114  032777  000010  176036   73$:    BIT     #SW03,@SWR              ;ASK PARAMETERS ?
003122  001002                                   BNE     42$                     ;IF NO, GO DO AUTO SIZING
003124  000137  004244                           JMP     63$                     ;GO SET UP FOR AUTO SIZING
003130  004737  003144           42$:    JSR     PC,23$                  ;GO ASK PARAMETERS
003134  105337  001415                           DECB    INIFLG                  ;INSURE NO AUTO SIZE IF QUESTIONS ANSWERED
003140  000137  004270                           JMP     16$                     ;GO TO THE NEXT BLOCK

                                                 ;GET THE ACTIVE LINES PARAMETER

003144                                   23$:
003144  104403                                   INSTR                           ;CALL THE STRING INPUT ROUTINE
003146  003464                                   69$                             ;POINTER TO MESSAGE TO BE PRINTED
003150  104405                                   PARAM                           ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
003152  000001                                   1                               ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
003154  000377                                   377                             ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
003156  001506                                   LINE0                           ;POINTER TO MAP LOCATION TO BE FILLED
003160     000                                   .BYTE   0                       ;MASK OF INVALID BITS FOR THIS PARAMETER
003161     001                                   .BYTE   1                       ;NUMBER OF PARAMETERS TO STORE
003162  105037  001416                           CLRB    HDRFLG                  ;MAKE SURE THE CHANGES ARE PRINTED

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

003166  005737  001512                           TST     MANTO                   ;IS STAGGERED THE MODE OF OPERATION?
003172  100021                                   BPL     26$                     ;IF NOT, SKIP THIS SEGMENT
003174  013703  001506                           MOV     LINE0,R3                ;GET A SCRATCH COPY OF THE ACTIVE LINES
003200  006003                           24$:    ROR     R3                      ;GET A LINE SELECTION BIT(EVEN NUMBER LINE)
003202  103410                                   BCS     25$                     ;IF IT IS SELECTED, CHECK TO SEE IF THE NEXT IS TOO
003204  001414                                   BEQ     26$                     ;IF ALL HAVE BEEN CHECKED, CONTINUE PROCESSING
003206  006203                                   ASR     R3                      ;IF IT IS 0,CHECK TO SEE IF THE NEXT IS TOO
003210  103373                                   BCC     24$                     ;IF THIS ONE'S 0 TOO, GO CHECK THE NEXT PAIR
003212  104402  001230           27$:    TYPE    ,$QUES                  ;THIS IS AN INCORRECT PARAMETER
003216  104402  010424                           TYPE    ,MBADLN                 ;LET THE USER KNOW ABOUT IT
003222  000750                                   BR      23$                     ;GO GET THE CORRECT PARAMETER
003224  001772                           25$:    BEQ     27$                     ;IF ANOTHER FLAG ISN'T SET, THERE'S AN ERROR
003226  006203                                   ASR     R3                      ;GET THE NEXT FLAG
003230  103370                                   BCC     27$                     ;IF IT ISN'T SET, THERE'S AN ERROR
003232  000241                                   CLC                             ;INITIALIZE THE "C" BIT FOR TESTING OF THE NEXT PAIR
003234  000761                                   BR      24$                     ;GO TEST THE NEXT PAIR OF FLAGS

                                                 ;GET THE LINE PARAMETER REGISTER ARGUMENT

003236                                   26$:
003236  104403                                   INSTR                           ;CALL THE STRING INPUT ROUTINE
```

```
      003240  003540                          70$                      ;POINTER TO MESSAGE TO BE PRINTED
      003242  104405                          PARAM                    ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
      003244  000000                          0                        ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
      003246  000017                          17                       ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
      003250  001510                          PAR0                     ;POINTER TO MAP LOCATION TO BE FILLED
      003252     000                          .BYTE   0                ;MASK OF INVALID BITS FOR THIS PARAMETER
      003253     001                          .BYTE   1                ;NUMBER OF PARAMETERS TO STORE
      003254  012702  001506                  MOV     #LINE0,R2        ;POINT TO THE LINE SELECTION PARAMETER
      003260  012703  001510                  MOV     #PAR0,R3         ;POINT TO THE CHOSEN PARAMETERS
      003264  011304                          MOV     (R3),R4          ;USE BAUD RATE AS AN INDEX IN DELAY TABLE
      003266  006304                          ASL     R4               ;ALIGN INDEX ON WORD BOUNDARY
      003270  016437  030444  006722          MOV     DLYTBL(R4),DLYCNT ;SET THE DELAY COUNT FOR THIS BAUD RATE
      003276  000313                          SWAB    (R3)             ;PLACE IN HIGH BYTE
      003300  052713  010070                  BIS     #10070,(R3)      ;PLACE EXTRA PARAMETERS INTO LOC
      003304  011262  000014          28$:    MOV     (R2),14(R2)      ;LOAD THE LINES
      003310  011363  000014                  MOV     (R3),14(R3)      ;LOAD THE PARAMETERS
      003314  062702  000014                  ADD     #14,R2           ;POINT TO THE NEXT SET
      003320  062703  000014                  ADD     #14,R3           ; .. OF BOTH PARAMETERS
      003324  020327  001774                  CMP     R3,#PAR17        ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
      003330  001365                          BNE     28$              ;IF NOT, GO LOAD SOME MORE PARAMETERS
      003332  000207                          RTS     PC               ;RETURN TO CALLING BLOCK
      003334  030600  052123  041440  66$:    .ASCIZ  <200>/1ST CSR ADDRESS (160000:163700):  /
  (1) 003400  030600  052123  053040  67$:    .ASCIZ  <200>/1ST VECTOR ADDRESS (300:770):  /
  (1) 003441     200  051102  046040  68$:    .ASCIZ  <200>/BR LEVEL (4:6):  /
  (1) 003464  046200  047111  051505  69$:    .ASCIZ  <200>/LINES ACTIVE BY BIT <IN OCTAL>(001:377):  /
  (1) 003540  042200  043105  052501  70$:    .ASCIZ  <200>/DEFAULT BAUD RATE <IN OCTAL>(00:17):  /
  (1) 003610  021600  047440  020106  71$:    .ASCIZ  <200>/# OF DZ11'S <IN OCTAL> (1:20):  /
  (1) 003652  046600  044501  052116  72$:    .ASCII  <200>/MAINTENANCE MODE/
  (1) 003673     200  055440  054105          .ASCII  <200>/ [EXTERNAL  <H325>-EIA ONLY       (E)]/
  (1) 003741     200  055440  047111          .ASCII  <200>/ [INTERNAL  <DZCSR03=1>           (I)]/
  (1) 004007     200  055440  052123          .ASCII  <200>/ [STAGGERED <H3271>-EIA ONLY      (S)]: /
  (1) 004057     200  055440  052123          .ASCIZ  <200>/ [STAGGERED <H3190>-20MA ONLY     (S)]: /
  (1) 004130  052200  050131  020105  74$:    .ASCIZ  <200>/TYPE "A" FOR EIA MODULE OR "B" FOR 20 MA (A:B): /
  (1) 004212  042600  052116  051105  75$:    .ASCIZ  <200>/ENTER DELAY PARAMETER: /
  (1)         004244                  .EVEN
  (1) 004244                          63$:
      004244  122737  000377  001415          CMPB    #377,INIFLG      ;ONLY DO AUTO SIZE ON 1ST START
      004252  001006                          BNE     16$
      004254  032777  000200  174676          BIT     #BIT7,@SWR       ;BIT7=1??
      004262  001002                          BNE     16$              ;BR IF NO AUTO SIZE
      004264  004737  011612                  JSR     PC,AUTO.SIZE     ;GO DO THE AUTO SIZE
      004270  105737  001416          16$:    TSTB    HDRFLG           ;HAS THE TABLE BEEN TYPED YET?
      004274  001021                          BNE     1$               ;IF SO, DON'T TYPE IT AGAIN
      004276  105337  001416                  DECB    HDRFLG           ;INDICATE THAT THE TABLE WILL BE TYPED
      004302  104402  010377                  TYPE    ,XHEAD           ;TYPE MAP HEADER
      004306  012700  001500                  MOV     #DZ.MAP,R0       ;SET POINTER
      004312  010037  001220          5$:     MOV     R0,$TMP1         ;POINT TO THE MAP LOCATION
      004316  012037  001222                  MOV     (R0)+,$TMP2      ;SET DATA
      004322  022737  177777  001222          CMP     #-1,$TMP2        ;END OF LIST?
      004330  001403                          BEQ     1$               ;BR IF YES
      004332  104411                  17$:    CONVRT                   ;CALL THE OCTAL TO ASCII CONVERSION ROUTINE
      004334  010466                          XSTATQ                   ;CONVERT THE DATA AT THIS ADDRESS
      004336  000765                          BR      5$               ;GO PRINT THE NEXT PARAMETER
      004340  005737  000042          1$:     TST     @#42             ;IS PROGRAM RUNNING UNDER MONITOR
      004344  001026                          BNE     3$               ;YES
      004346  032777  000100  174604          BIT     #SW06,@SWR       ;DESELECT SPECIFIC DEVICES??
```

```
004354  001422                                  BEQ     3$              ;BR IF NO.
004356  104402  010320                          TYPE    ,MNEW           ;TYPE THE MESSAGE.
004362  005000                                  CLR     R0              ;ZERO DATA DISPLAY
004364  000000                                  HALT                    ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
004366  027737  174566  001312                  CMP     @SWR,$DEVM      ;IS THE NUMBER VALID?
004374  101404                                  BLOS    2$              ;BR IF NUMBER IS OK.
004376  104402  010172                          TYPE    ,MERR3          ;TELL USER OF INVALID NUMBER.
004402  000000                          9$:     HALT                    ;STOP EVERY THING.
004404  000776                                  BR      9$              ;RESTART THE PROGRAM AGAIN.
004406  017737  174546  001404          2$:     MOV     @SWR,DZACTV     ;GET NEW DEVICE PATTERN
004414  013700  001404                          MOV     DZACTV,R0       ;SHOW THE USER WHAT HE SELECTED.
004420  000000                                  HALT                    ;CONTINUE DYNAMIC SWITCHES.
004422  032777  000020  174530          3$:     BIT     #SW04,@SWR      ;CHECK TO SEE IF DELAY COUNT CHANGES
004430  001407                                  BEQ     18$             ;IF NOT, GO CLEAR VECTOR AREA
004432  104403                                  INSTR                   ;CALL THE STRING INPUT ROUTINE
004434  004212                                  75$                     ;POINTER TO MESSAGE TO BE PRINTED
004436  104405                                  PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
004440  000001                                  1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
004442  177777                                  177777                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
004444  006722                                  DLYCNT                  ;POINTER TO MAP LOCATION TO BE FILLED
004446     000                                  .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
004447     001                                  .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
004450  012700  000300                  18$:    MOV     #300,R0         ;PREPARE TO CLEAR THE FLOATING
004454  012701  000302                          MOV     #302,R1         ;VECTOR AREA. 300-776
004460  010120                          4$:     MOV     R1,(R0)+        ;START PUTTING "PC+2 - HALT"
004462  005021                                  CLR     (R1)+           ;IN VECTOR AREA.
004464  022021                                  CMP     (R0)+,(R1)+     ;POP POINTERS
004466  022700  001000                          CMP     #1000,R0        ;ALL DONE??
004472  001372                                  BNE     4$              ;BR IF NO.

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

004474  012706  001120                  .BEGIN: MOV     #STACK,SP       ;SET UP STACK
004500  106427  000340                          MTPS    #PR7            ;LOCK OUT INTERRUPTS
004504  005737  000042                          TST     @#42            ;IS PROGRAM UNDER MONITOR CONTROL
004510  001015                                  BNE     2$              ;BR IF YES
004512  032777  000004  174440                  BIT     #BIT2,@SWR      ;CHECK FOR LOCK ON TEST
004520  001406                                  BEQ     1$              ;BR IF NO LOCK DESIRED.
004522  104402  010216                          TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
004526  012737  000240  005010                  MOV     #NOP,TTST       ;ADJUST SCOPE ROUTINE.
004534  000403                                  BR      2$              ;CONTINUE ALONG.
004536  013737  005232  005010          1$:     MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
004544  012737  011070  001126          2$:     MOV     #CYCLE,$LPADR   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
004552  104402  010107                          TYPE    ,MR             ;TYPE "RUNNING"
004556  000177  174344                          JMP     @$LPADR         ;START TESTING
```

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

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

004562                                  $EOP:
004562    000004                                SCOPE
004564    005037    001136                      CLR     $ERRPC          ;CLEAR LAST ERROR PC
004570    105037    001123                      CLRB    $ERFLG          ;CLEAR ERROR FLAG
004574    104402    010063                      TYPE    ,MEPASS         ;TYPE END PASS
004600    104402    010245                      TYPE    ,MCSRX          ;TYPE CSR
004604    104412    004742                      CNVRT   ,XCSR           ;SHOW IT
004610    104402    010253                      TYPE    ,MVECX          ;TYPE VECTOR
004614    104412    004750                      CNVRT   ,XVEC           ;SHOW IT
004620    005237    001242                      INC     $PASS           ;RAISE PASS COUNT
004624    104402    010261                      TYPE    ,MPASSX         ;TYPE PASSES
004630    104412    004756                      CNVRT   ,XPASS          ;SHOW IT
004634    005337    001242                      DEC     $PASS           ;RESTORE PASS COUNT
004640    104402    010272                      TYPE    ,MERRX          ;TYPE ERRORS
004644    104412    004764                      CNVRT   ,XERR           ;SHOW IT
004650    105337    001411                      DECB    SAVNUM          ;ARE ALL DEVICES TESTED?
004654    001030                                BNE     $DOAGN          ;BR IF NO.
004656    113737    001410    001411            MOVB    DZNUM,SAVNUM    ;RESTORE THE COUNT
004664    005037    001226                      CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
004670    005237    001242                      INC     $PASS           ;;INCREMENT THE PASS NUMBER
004674    042737    100000    001242            BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
004702    005327                                DEC     (PC)+           ;;LOOP?
004704    000001                        $EOPCT: .WORD   1
004706    003013                                BGT     $DOAGN          ;;YES
004710    012737                                MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
004712    000001                        $ENDCT: .WORD   1
004714    004704                                $EOPCT
004716    013700    000042              $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
004722    001405                                BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
004724    000005                                RESET                   ;;CLEAR THE WORLD
004726    004710                        $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
004730    000240                                NOP                     ;;SAVE ROOM
004732    000240                                NOP                     ;;FOR
004734    000240                                NOP                     ;;ACT11
004736                                  $DOAGN:
004736    000137                                JMP     @(PC)+          ;;RETURN
004740    011070                        $RTNAD: .WORD   CYCLE

004742    000001                        XCSR:   1
004744       006       002                      .BYTE   6,2
004746    002042                                DZCSR
004750    000001                        XVEC:   1
004752       003       002                      .BYTE   3,2
004754    002072                                DZRIV
```

```
004756  000001                         XPASS:  1
004760     006       002                       .BYTE   6,2
004762  001242                                 $PASS
004764  000001                         XERR:   1
004766     006       002                       .BYTE   6,2
004770  001132                                 $ERTTL

                                               ;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

004772                                 $SCOPE:
004772  004737  007360                 .SCOPE: JSR     PC,SERV.G           ;FIND OUT IF <↑G> WAS HIT
004776  005037  001136                         CLR     $ERRPC              ;CLEAR LAST ERROR PC.
005002  022716  012376                         CMP     #TST1+2,(SP)        ;IS THIS THE SCOPE AT THE BEGINNING OF TST1?
005006  001413                                 BEQ     $XTSTR              ;IF SO, DON'T LOOP ON IT
005010  000406                         TTST:   BR      1$                  ;GOTO 1$    (IF LOCK SW02=1; THIS LOC =240)
005012  105777  174146                         TSTB    @$TKS               ;KEYBOARD DONE?
005016  100067                                 BPL     $OVER               ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
005020  017766  174142  177776                 MOV     @$TKB,-2(SP)        ;CLEAR DONE BIT
005026  032777  040000  174124         1$:     BIT     #BIT14,@SWR         ;;LOOP ON PRESENT TEST?
005034  001060                                 BNE     $OVER               ;;YES IF SW14=1
                                       ;#####START OF CODE FOR THE XOR TESTER#####
005036  000416                         $XTSTR: BR      6$                  ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                           ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
005040  013746  000004                         MOV     @#ERRVEC,-(SP)      ;;SAVE THE CONTENTS OF THE ERROR VECTOR
005044  012737  005064  000004                 MOV     #5$,@#ERRVEC        ;;SET FOR TIMEOUT
005052  005737  177060                         TST     @#177060            ;;TIME OUT ON XOR?
005056  012637  000004                         MOV     (SP)+,@#ERRVEC      ;;RESTORE THE ERROR VECTOR
005062  000436                                 BR      $SVLAD              ;;GO TO THE NEXT TEST
005064  022626                         5$:     CMP     (SP)+,(SP)+         ;;CLEAR THE STACK AFTER A TIME OUT
005066  012637  000004                         MOV     (SP)+,@#ERRVEC      ;;RESTORE THE ERROR VECTOR
005072  000441                                 BR      $OVER               ;;LOOP ON THE PRESENT TEST
005074                                 6$:;#####END OF CODE FOR THE XOR TESTER#####
005074  105737  001123                 2$:     TSTB    $ERFLG              ;;HAS AN ERROR OCCURRED?
005100  001404                                 BEQ     3$                  ;;BR IF NO
005102  105037  001123                 4$:     CLRB    $ERFLG              ;;ZERO THE ERROR FLAG
005106  005037  001226                         CLR     $TIMES              ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
005112  032777  004000  174040         3$:     BIT     #BIT11,@SWR         ;;INHIBIT ITERATIONS?
005120  001011                                 BNE     1$                  ;;BR IF YES
005122  005737  001242                         TST     $PASS               ;;IF FIRST PASS OF PROGRAM
005126  001406                                 BEQ     1$                  ;;      INHIBIT ITERATIONS
005130  005237  001124                         INC     $ICNT               ;;INCREMENT ITERATION COUNT
005134  023737  001226  001124                 CMP     $TIMES,$ICNT        ;;CHECK THE NUMBER OF ITERATIONS MADE
005142  002015                                 BGE     $OVER               ;;BR IF MORE ITERATION REQUIRED
```

```
005144  012737  000001  001124  1$:      MOV     #1,$ICNT          ;;REINITIALIZE THE ITERATION COUNTER
005152  013737  005234  001226           MOV     $MXCNT,$TIMES     ;;SET NUMBER OF ITERATIONS TO DO
005160  105237  001122          $SVLAD:  INCB    $TSTNM            ;;COUNT TEST NUMBERS
005164  113737  001122  001240           MOVB    $TSTNM,$TESTN     ;;SET TEST NUMBER IN APT MAILBOX
005172  011637  001126                   MOV     (SP),$LPADR       ;;SAVE SCOPE LOOP ADDRESS
005176  013777  001122  173756  $OVER:   MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER
005204  013716  001126                   MOV     $LPADR,(SP)       ;;FUDGE RETURN ADDRESS
005210  105037  001417          3$:      CLRB    MNTFLG            ;CLEAR THE MAINTENANCE BIT SETTER AFTER EACH TEST
005214  005737  001370                   TST     MODE              ;HAS THE MODE BEEN CHANGED?
005220  001003                           BNE     4$                ;IF NOT INTERNAL , GO DO A TEST
005222  112737  000010  001417           MOVB    #MAINT,MNTFLG     ;IF INTERNAL MODE NOW, SET THE MAINTENANCE BIT
005230  000002                  4$:      RTI                       ;GO DO THE TEST
005232  000406                  BRW:     406
005234  000005                  $MXCNT:  5                         ;;MAX. NUMBER OF ITERATIONS

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

005236  032777  001000  173714  .SCOP1:  BIT     #SW09,@SWR        ;IS SW09=1(SET)?
005244  001405                           BEQ     1$                ;BR IF NOT SET.
005246  005737  001362                   TST     LOCK              ;IS THER A TIGHT LOOP SPECIFIED?
005252  001402                           BEQ     1$                ;IF NO, RETURN
005254  013716  001362                   MOV     LOCK,(SP)         ;IF YES, GOTO THE ADDRESS IN LOCK.
005260  000002                  1$:      RTI                       ;GO BACK.

005262  032777  010000  173670  .TYPE:   BIT     #SW12,@SWR        ;INHIBIT ALL PRINTOUT??
005270  001403                           BEQ     1$                ;IF NOT, GO TYPE
005272  062716  000002                   ADD     #2,(SP)           ;SKIP OVER MESSAGE POINTER
005276  000002                           RTI                       ;RETURN TO WHERE PROCEDURE WAS INVOKED
005300                          1$:
                                .SBTTL  TYPE ROUTINE

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

005300  105737  001177          $TYPE:   TSTB    $TPFLG            ;;IS THERE A TERMINAL?
005304  100002                           BPL     1$                ;;BR IF YES
005306  000000                           HALT                      ;;HALT HERE IF NO TERMINAL
005310  000430                           BR      3$                ;;LEAVE
005312  010046                  1$:      MOV     R0,-(SP)          ;;SAVE R0
005314  017600  000002                   MOV     @2(SP),R0         ;;GET ADDRESS OF ASCIZ STRING
005320  122737  000001  001254           CMPB    #APTENV,$ENV      ;;RUNNING IN APT MODE
005326  001011                           BNE     62$               ;;NO,GO CHECK FOR APT CONSOLE
005330  132737  000100  001255           BITB    #APTSPOOL,$ENVM   ;;SPOOL MESSAGE TO APT
```

```
005336  001405                                BEQ     62$                     ;;NO,GO CHECK FOR CONSOLE
005340  010037  005350                        MOV     R0,61$                  ;;SETUP MESSAGE ADDRESS FOR APT
005344  004737  005570                        JSR     PC,$ATY3                ;;SPOOL MESSAGE TO APT
005350  000000                        61$:    .WORD   0                       ;;MESSAGE ADDRESS
005352  132737  000040  001255        62$:    BITB    #APTCSUP,$ENVM          ;;APT CONSOLE SUPPRESSED
005360  001003                                BNE     60$                     ;;YES,SKIP TYPE OUT
005362  112046                        2$:     MOVB    (R0)+,-(SP)             ;;PUSH CHARACTER TO BE TYPED ONTO STACK
005364  001005                                BNE     4$                      ;;BR IF IT ISN'T THE TERMINATOR
005366  005726                                TST     (SP)+                   ;;IF TERMINATOR POP IT OFF THE STACK
005370  012600                        60$:    MOV     (SP)+,R0                ;;RESTORE R0
005372  062716  000002                3$:     ADD     #2,(SP)                 ;;ADJUST RETURN PC
005376  000002                                RTI                             ;;RETURN
005400  122716  000011                4$:     CMPB    #HT,(SP)                ;;BRANCH IF <HT>
005404  001430                                BEQ     8$
005406  122716  000200                        CMPB    #CRLF,(SP)              ;;BRANCH IF NOT <CRLF>
005412  001006                                BNE     5$
005414  005726                                TST     (SP)+                   ;;POP  <CR><LF> EQUIV
005416  104402                                TYPE                            ;;TYPE A CR AND LF
005420  001231                                $CRLF
005422  105037  005556                        CLRB    $CHARCNT                ;;CLEAR CHARACTER COUNT
005426  000755                                BR      2$                      ;;GET NEXT CHARACTER
005430  004737  005512                5$:     JSR     PC,$TYPEC               ;;GO TYPE THIS CHARACTER
005434  123726  001176                6$:     CMPB    $FILLC,(SP)+            ;;IS IT TIME FOR FILLER CHARS.?
005440  001350                                BNE     2$                      ;;IF NO GO GET NEXT CHAR.
005442  013746  001174                        MOV     $NULL,-(SP)             ;;GET # OF FILLER CHARS. NEEDED
                                                                              ;;AND THE NULL CHAR.
005446  105366  000001                7$:     DECB    1(SP)                   ;;DOES A NULL NEED TO BE TYPED?
005452  002770                                BLT     6$                      ;;BR IF NO--GO POP THE NULL OFF OF STACK
005454  004737  005512                        JSR     PC,$TYPEC               ;;GO TYPE A NULL
005460  105337  005556                        DECB    $CHARCNT                ;;DO NOT COUNT AS A COUNT
005464  000770                                BR      7$                      ;;LOOP

                                      ;HORIZONTAL TAB PROCESSOR

005466  112716  000040                8$:     MOVB    #' ,(SP)                ;;REPLACE TAB WITH SPACE
005472  004737  005512                9$:     JSR     PC,$TYPEC               ;;TYPE A SPACE
005476  132737  000007  005556                BITB    #7,$CHARCNT             ;;BRANCH IF NOT AT
005504  001372                                BNE     9$                      ;;TAB STOP
005506  005726                                TST     (SP)+                   ;;POP SPACE OFF STACK
005510  000724                                BR      2$                      ;;GET NEXT CHARACTER
005512  105777  173452                $TYPEC: TSTB    @$TPS                   ;;WAIT UNTIL PRINTER IS READY
005516  100375                                BPL     $TYPEC
005520  116677  000002  173444                MOVB    2(SP),@$TPB             ;;LOAD CHAR TO BE TYPED INTO DATA REG.
005526  122766  000015  000002                CMPB    #CR,2(SP)               ;;IS CHARACTER A CARRIAGE RETURN?
005534  001003                                BNE     1$                      ;;BRANCH IF NO
005536  105037  005556                        CLRB    $CHARCNT                ;;YES--CLEAR CHARACTER COUNT
005542  000406                                BR      $TYPEX                  ;;EXIT
005544  122766  000012  000002        1$:     CMPB    #LF,2(SP)               ;;IS CHARACTER A LINE FEED?
005552  001402                                BEQ     $TYPEX                  ;;BRANCH IF YES
005554  105227                                INCB    (PC)+                   ;;COUNT THE CHARACTER
005556  000000                        $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
005560  000207                        $TYPEX: RTS     PC

                                      .SBTTL  APT COMMUNICATIONS ROUTINE

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

C05

```
005562  112737  000001  006026  $ATY1:  MOVB    #1,$FFLG         ;;TO REPORT FATAL ERROR
005570  112737  000001  006024  $ATY3:  MOVB    #1,$MFLG         ;;TO TYPE A MESSAGE
005576  000403                          BR      $ATYC
005600  112737  000001  006026  $ATY4:  MOVB    #1,$FFLG         ;;TO ONLY REPORT FATAL ERROR
005606                                  $ATYC:
005606  010046                          MOV     R0,-(SP)         ;;PUSH R0 ON STACK
005610  010146                          MOV     R1,-(SP)         ;;PUSH R1 ON STACK
005612  105737  006024                  TSTB    $MFLG            ;;SHOULD TYPE A MESSAGE?
005616  001450                          BEQ     5$               ;;IF NOT:  BR
005620  122737  000001  001254          CMPB    #APTENV,$ENV     ;;OPERATING UNDER APT?
005626  001031                          BNE     3$               ;;IF NOT:  BR
005630  132737  000100  001255          BITB    #APTSPOOL,$ENVM  ;;SHOULD SPOOL MESSAGES?
005636  001425                          BEQ     3$               ;;IF NOT:  BR
005640  017600  000004                  MOV     @4(SP),R0        ;;GET MESSAGE ADDR.
005644  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
005652  005737  001234          1$:     TST     $MSGTYPE         ;;SEE IF DONE W/ LAST XMISSION?
005656  001375                          BNE     1$               ;;IF NOT:  WAIT
005660  010037  001250                  MOV     R0,$MSGAD        ;;PUT ADDR IN MAILBOX
005664  105720                  2$:     TSTB    (R0)+            ;;FIND END OF MESSAGE
005666  001376                          BNE     2$
005670  163700  001250                  SUB     $MSGAD,R0        ;;SUB START OF MESSAGE
005674  006200                          ASR     R0               ;;GET MESSAGE LNGTH IN WORDS
005676  010037  001252                  MOV     R0,$MSGLGT       ;;PUT LENGTH IN MAILBOX
005702  012737  000004  001234          MOV     #4,$MSGTYPE      ;;TELL APT TO TAKE MSG.
005710  000413                          BR      5$
005712  017637  000004  005736  3$:     MOV     @4(SP),4$        ;;PUT MSG ADDR IN JSR LINKAGE
005720  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
005726  013746  177776                  MOV     177776,-(SP)     ;;PUSH 177776 ON STACK
005732  004737  005300                  JSR     PC,$TYPE         ;;CALL TYPE MACRO
005736  000000                  4$:     .WORD   0
005740                          5$:
005740  105737  006026          10$:    TSTB    $FFLG            ;;SHOULD REPORT FATAL ERROR?
005744  001416                          BEQ     12$              ;;IF NOT:  BR
005746  005737  001254                  TST     $ENV             ;;RUNNING UNDER APT?
005752  001413                          BEQ     12$              ;;IF NOT:  BR
005754  005737  001234          11$:    TST     $MSGTYPE         ;;FINISHED LAST MESSAGE?
005760  001375                          BNE     11$              ;;IF NOT:  WAIT
005762  017637  000004  001236          MOV     @4(SP),$FATAL    ;;GET ERROR #
005770  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
005776  005237  001234                  INC     $MSGTYPE         ;;TELL APT TO TAKE ERROR
006002  105037  006026          12$:    CLRB    $FFLG            ;;CLEAR FATAL FLAG
006006  105037  006025                  CLRB    $LFLG            ;;CLEAR LOG FLAG
006012  105037  006024                  CLRB    $MFLG            ;;CLEAR MESSAGE FLAG
006016  012601                          MOV     (SP)+,R1         ;;POP STACK INTO R1
006020  012600                          MOV     (SP)+,R0         ;;POP STACK INTO R0
006022  000207                          RTS     PC               ;;RETURN
006024     000                  $MFLG:  .BYTE   0                ;;MESSG. FLAG
006025     000                  $LFLG:  .BYTE   0                ;;LOG FLAG
006026     000                  $FFLG:  .BYTE   0                ;;FATAL FLAG
        006030                          .EVEN
        000200                  APTSIZE=200
        000001                  APTENV=001
        000100                  APTSPOOL=100
        000040                  APTCSUP=040

                                        ;STRING INPUT ROUTINE
```

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

006030  010346                          .INSTR: MOV    R3,-(SP)          ;SAVE R3 ON STACK
006032  010446                                  MOV    R4,-(SP)          ;SAVE R4 ON STACK
006034  017637  000004  006052                  MOV    @4(SP),.MSG       ;GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
006042  062766  000002  000004                  ADD    #2,4(SP)          ;POINT TO INSTRUCTION AFTER ADDRESS POINTER
006050  104402                          .INST1: TYPE                     ;PRINT THE MESSAGE
006052  000000                          .MSG:   0                        ;MESSAGE IS POINTED TO FROM HERE
006054  012704  010620                          MOV    #INBUF,R4         ;POINT R4 TO THE INPUT BUFFER
006060  012703  000007                          MOV    #7,R3             ;SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
006064  105777  173074                  1$:     TSTB   @$TKS             ;HAS A CHARACTER BEEN RECEIVED?
006070  100375                                  BPL    1$                ;IF NO, KEEP WAITING FOR IT
006072  117714  173070                          MOVB   @$TKB,(R4)        ;IF YES, SAVE IT IN THE INPUT BUFFER
006076  142714  000200                          BICB   #200,(R4)         ;KEEP ONLY THE 7-BIT ASCII INFORMATION
006102  122427  000015                          CMPB   (R4)+,#15         ;IS THIS CHARACTER A LINE FEED?
006106  001417                                  BEQ    INSTR2            ;IF SO, TERMINATE THE INPUT SEQUENCE
006110  105777  173054                  2$:     TSTB   @$TPS             ;IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
006114  100375                                  BPL    2$                ;IF WE CAN'T, WAIT UNTIL WE CAN
006116  017777  173044  173045                  MOV    @$TKB,@$TPB       ;ECHO THE CHARACTER BACK
006124  005303                                  DEC    R3                ;REDUCE THE NUMBER OF CHARACTERS RECEIVED
006126  001356                                  BNE    1$                ;IF WE DON'T HAVE 7, GO GET SOME MORE
006130  012604                                  MOV    (SP)+,R4          ;IF WE HAVE 7, RESTORE R4
006132  012603                                  MOV    (SP)+,R3          ;RESTORE R3
006134  010346                          .INSTE: MOV    R3,-(SP)          ;SAVE R3 ON THE STACK
006136  010446                                  MOV    R4,-(SP)          ;SAVE R4 ON THE STACK
006140  104402  001230                          TYPE   ,$QUES            ;PRINT A QUESTION MARK... WHAT'S GOING ON?
006144  000741                                  BR     .INST1            ;GO PRINT THE MESSAGE AGAIN
006146  012604                          INSTR2: MOV    (SP)+,R4          ;RESTORE R4
006150  012603                                  MOV    (SP)+,R3          ;RESTORE R3
006152  000002                                  RTI                      ;RETURN TO THE MAIN PROCEDURE

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

006154  010546                          .PARAM: MOV    R5,-(SP)          ;SAVE R5 ON THE STACK
006156  010446                                  MOV    R4,-(SP)          ;SAVE R4 ON THE STACK
006160  016605  000004                          MOV    4(SP),R5          ;GET THE SETUP INFORMATION POINTER
006164  012537  006344                          MOV    (R5)+,LOLIM       ;SET THE LOW LIMIT FOR THE INPUT
006170  012537  006346                          MOV    (R5)+,HILIM       ;SET THE HIGH LIMIT FOR THE INPUT
006174  012537  006350                          MOV    (R5)+,DEVADR      ;SAVE THE ADDRESS WHERE THE RESULT WILL BE STORED
006200  112537  006352                          MOVB   (R5)+,LOBITS      ;GET THE MASK OF THE INCORRECT BITS
006204  112537  006353                          MOVB   (R5)+,ADRCNT      ;GET THE COUNT OF ITEMS TO BE STORED
006210  010566  000004                          MOV    R5,4(SP)          ;POINT TO WHERE MAIN LINE PROGRAM WILL RESUME
006214  005005                          PARAM1: CLR    R5                ;INITIALIZE THE ASCII TO OCTAL RESULT WORD
006216  012704  010620                          MOV    #INBUF,R4         ;POINT TO THE INPUT BUFFER
006222  122714  000015                          CMPB   #15,(R4)          ;IS THIS CHARACTER A CARRIAGE RETURN?
006226  001420                                  BEQ    PARERR            ;IF SO, PRINT THE MESSAGE AGAIN
006230  121427  000060                  1$:     CMPB   (R4),#60          ;IS THIS CHARACTER BELOW THE NUMERIC RANGE?
006234  002415                                  BLT    PARERR            ;IF SO, GO PRINT THE MESSAGE AGAIN
006236  121427  000067                          CMPB   (R4),#67          ;IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
006242  003012                                  BGT    PARERR            ;IF SO, GO PRINT THE MESSAGE AGAIN
006244  142714  000060                          BICB   #60,(R4)          ;ISOLATE THE NUMBER THE CHARACTER REPRESENTS
006250  152405                                  BISB   (R4)+,R5          ;CONCATENATE THESE BITS TO THE ALREADY EXISTING STRING
006252  122714  000015                          CMPB   #15,(R4)          ;IS THE NEXT CHARACTER A CARRIAGE RETURN?
006256  001406                                  BEQ    LIMITS            ;IF SO, GO SEE IF NUMBER IS WITHIN LIMITS
006260  006305                                  ASL    R5                ;CLEAR BIT POSITION 0, MOVE EXISTING STRING TO LEFT
```

```
006262  006305                                   ASL     R5                ;CLEAR POSITION 1, MOVE STRING TO LEFT AGAIN
006264  006305                                   ASL     R5                ;MOVE THE STRING ONE MORE TIME TO MAKE ROOM FOR
                                                                           ;NEXT THREE BITS
006266  000760                                   BR      1$                ;GO GET THE NEXT CHARACTER
006270  104404                          PARERR:  INSTER                    ;THERE WAS AN ERROR... GO PRINT MESSAGE AGAIN
006272  000750                                   BR      PARAM1            ;TRY GETTING THE PARAMETERS AGAIN

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

006274  020537  006346                  LIMITS:  CMP     R5,HILIM          ;DOES RESULT EXCEED ITS MAXIMUM CORRECT VALUE?
006300  101373                                   BHI     PARERR            ;IF YES, GO PRINT THE MESSAGE AGAIN
006302  020537  006344                           CMP     R5,LOLIM          ;IS THE RESULT LOWER THAN ALLOWED?
006306  103770                                   BLO     PARERR            ;IF YES, GO PRINT THE MESSAGE AGAIN
006310  133705  006352                           BITB    LOBITS,R5         ;ARE ANY INCORRECT BITS SET IN THE RESULT?
006314  001365                                   BNE     PARERR            ;IF SO, GO PRINT THE MESSAGE AGAIN

                                                 ;STORE NUMBER AT SPECIFIED ADDRESS

006316  013704  006350                           MOV     DEVADR,R4         ;POINT TO THE LOCATION WHERE THE RESULT WILL BE STORED
006322  010524                          1$:      MOV     R5,(R4)+          ;STORE THE RESULT
006324  062705  000002                           ADD     #2,R5             ;CALCULATE THE NEXT DATUM
006330  105337  006353                           DECB    ADRCNT            ;REDUCE COUNT OF STORED RESULTS. IS IT EXCEEDED?
006334  001372                                   BNE     1$                ;IF NOT, GO STORE THE NEXT DATUM
006336  012604                                   MOV     (SP)+,R4          ;RESTORE R4
006340  012605                                   MOV     (SP)+,R5          ;RESTORE R5
006342  000002                                   RTI                       ;RETURN TO THE MAIN PROGRAM

006344  000000                          LOLIM:   0                         ;LOWEST ACCEPTABLE VALUE
006346  000000                          HILIM:   0                         ;HIGHEST ACCEPTABLE
006350  000000                          DEVADR:  0                         ;LOCATION WHERE RESULT WILL BE STORED
006352     000                          LOBITS:  .BYTE   0                 ;INCORRECT BITS MASK
006353     000                          ADRCNT:  .BYTE   0                 ;COUNT OF ITEMS TO BE STORED

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

006354  016637  000004  001402          .SAV05:  MOV     4(SP),SAVPC       ;SAVE R7 (PC)

                                                 ;SAVE R0-R5

006362  010537  001214                  SV05:    MOV     R5,$REG5          ;SAVE R5
006366  010437  001212                           MOV     R4,$REG4          ;SAVE R4
006372  010337  001210                           MOV     R3,$REG3          ;SAVE R3
006376  010237  001206                           MOV     R2,$REG2          ;SAVE R2
006402  010137  001204                           MOV     R1,$REG1          ;SAVE R1
006406  010037  001202                           MOV     R0,$REG0          ;SAVE R0
006412  000002                                   RTI                       ;LEAVE.

                                                 ;RESTORE R0-R5

006414  013700  001202                  .RES05:  MOV     $REG0,R0          ;RESTORE R0
006420  013701  001204                           MOV     $REG1,R1          ;RESTORE R1
006424  013702  001206                           MOV     $REG2,R2          ;RESTORE R2
006430  013703  001210                           MOV     $REG3,R3          ;RESTORE R3
006434  013704  001212                           MOV     $REG4,R4          ;RESTORE R4
```

```
006440  013705  001214                   MOV     $REG5,R5            ;RESTORE R5
006444  000002                           RTI                         ;LEAVE

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

006446  104402  001231           .CONVR: TYPE    ,$CRLF              ;PRINT A CARRIAGE RETURN
006452  010046                   .CNVRT: MOV     R0,-(SP)            ;SAVE R0
006454  010146                           MOV     R1,-(SP)            ;SAVE R1
006456  010346                           MOV     R3,-(SP)            ;SAVE R3
006460  010446                           MOV     R4,-(SP)            ;SAVE R4
006462  010546                           MOV     R5,-(SP)            ;SAVE R5
006464  017601  000012                   MOV     @12(SP),R1          ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
006470  062766  000002  000012           ADD     #2,12(SP)           ;POINT TO WHERE MAIN PROGRAM WILL RESUME
006476  012137  006622                   MOV     (R1)+,WRDCNT        ;GET NUMBER OF WORDS TO BE PRINTED
006502  112105                   1$:     MOVB    (R1)+,R5            ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
006504  112100                           MOVB    (R1)+,R0            ;GET THE NUMBER OF SPACES TO PRINT
006506  013104                           MOV     @(R1)+,R4           ;COPY THE WORD TO BE CONVERTED
006510  110537  006624                   MOVB    R5,CHRCNT           ;COPY THE CHARACTER COUNT
006514  010403                   3$:     MOV     R4,R3               ;COPY THE ARGUMENT WORD AGAIN
006516  042703  177770                   BIC     #↑C<7>,R3           ;ISOLATE THREE BITS TO BE TREATED AS A CHARACTER
006522  062703  000060                   ADD     #060,R3             ;MAKE AN ASCII CHARACTER OUT OF THEM
006526  110346                           MOVB    R3,-(SP)            ;SAVE THAT CHARACTER
006530  006004                           ROR     R4                  ;MOVE THE NEXT THREE BITS INTO PLACE
006532  006204                           ASR     R4                  ;MOVE THEM AGAIN
006534  006204                           ASR     R4                  ;AND FINALLY A THIRD TIME
006536  005305                           DEC     R5                  ;REDUCE CHARACTER COUNT.ARE ALL CHARACTERS
                                                                     ;BUILT?
006540  001365                           BNE     3$                  ;IF NO, GO BUILD THE NEXT ONE.
006542  012703  010724                   MOV     #MDATA,R3           ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
006546  112623                   4$:     MOVB    (SP)+,(R3)+         ;STORE THE CHARACTER, STARTING WITH THE MOST
006550  105337  006624                   DECB    CHRCNT              ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
006554  001374                           BNE     4$                  ;IF NO, GO TRANSFER ANOTHER
006556  105700                           TSTB    R0                  ;ARE ANY SPACES TO BE PRINTED?
006560  001404                           BEQ     6$                  ;IF NO, DON'T SET UP ANY
006562  112723  000040           5$:     MOVB    #040,(R3)+          ;ADD A SPACE TO THE OUTPUT BUFFER
006566  105300                           DECB    R0                  ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
006570  001374                           BNE     5$                  ;IF YES, GO ADD ANOTHER SPACE
006572  105013                   6$:     CLRB    (R3)                ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
006574  104402  010724                   TYPE    ,MDATA              ;PRINT THE STRING WE JUST BUILT
006600  005337  006622                   DEC     WRDCNT              ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
006604  001336                           BNE     1$                  ;IF YES, GO CONVERT THEM
006606  012605                           MOV     (SP)+,R5            ;RESTORE R5
006610  012604                           MOV     (SP)+,R4            ;RESTORE R4
006612  012603                           MOV     (SP)+,R3            ;RESTORE R3
006614  012601                           MOV     (SP)+,R1            ;RESTORE R1
006616  012600                           MOV     (SP)+,R0            ;RESTORE R0
006620  000002                           RTI                         ;RETURN TO THE MAIN PROGRAM
006622  000000                   WRDCNT: 0
006624     000                   CHRCNT: .BYTE                       ;NUMBER OF CHARACTERS TO PRINT
006625     000                   SPACNT: .BYTE   0                   ;NUMBER OF SPACES TO PRINT

006626  000000                   BINWRD: 0



                                         ;TRAP DISPATCH SERVICE
```

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

006630  010046                  .TRPSR: MOV     R0,-(SP)        ;SAVE R0. USE R0 TO FIND TRAP ROUTINE
006632  016600  000002                  MOV     2(SP),R0        ;GET TRAP ADDRESS
006636  005740                          TST     -(R0)           ;GET TRAP
006640  111000                          MOVB    (R0),R0         ;GET RIGHT BYTE OF TRAP(TRAP OFFSET)
006642  006300                          ASL     R0              ;POSITION OFFSET FOR TABLE INDEXING
006644  016000  002002                  MOV     .TRPTAB(R0),R0  ;PLACE INDEXED ADDRESS OF TABLE IN R0
006650  000200                          RTS     R0              ;TRANSFER TO THAT ADDRESS AND RESTORE OLD R0

                                        ;DEVICE CLEAR ROUTINE
                                        ;ISSUE A DEVICE CLEAR
                                        ;--------------------
006652                          .DEVICE.CLR:
006652  052777  000020  173162          BIS     #DCLR,@DZCSR    ;SET DCLR
006660  032777  000020  173154  1$:     BIT     #DCLR,@DZCSR    ;DID IT CLEAR?
006666  001374                          BNE     1$              ;BR IF NO
006670  000002                          RTI                     ;EXIT ROUTINE

                                        ;ROUTINE TO HANDLE MAINTENANCE BIT SETTING WITH DEVICE CLEAR
                                        ;-----------------------------------------------------------
006672  104413                  .DCLASM:DEVICE.CLR              ;ISSUE A DEVICE CLEAR
006674  153777  001417  173140          BISB    MNTFLG,@DZCSR   ;LOAD THE MAINTENANCE BIT IF IT IS I MODE
006702  000002                          RTI                     ;RETURN TO CALLING ROUTINE

006704                          .DELAY:
006704  010046                          MOV     R0,-(SP)        ;SAVE R0
006706  013700  006722                  MOV     DLYCNT,R0       ;SET COUNT
006712  005300                  1$:     DEC     R0              ;DELAY
006714  001376                          BNE     1$              ;
006716  012600                          MOV     (SP)+,R0        ;RESTORE R0
006720  000002                          RTI                     ;LEAVE ROUTINE
006722  000001                  DLYCNT: .WORD   1               ;PATCHABLE LOC FOR MORE TIME

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

006724  013716  001360          .ADVANCE:MOV    NEXT,(SP)       ;CRUNCH STACK WITH ADDRESS OF SCOPE CALL
006730  005037  001362                  CLR     LOCK            ;RESET TIGHT LOOP ADDRESS
006734  000002                          RTI                     ;CHECK TO SEE IF OLD TEST GETS REPEATED

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

006736  004737  007360          $ERROR: JSR     PC,SERV.G       ;FIND OUT IF <↑G> WAS HIT
006742  032777  010000  172210          BIT     #SW12,@SWR      ;BELL ON ERROR?
006750  001406                          BEQ     XBX             ;BR IF NO BELL
006752  105777  172212                  TSTB    @$TPS           ;TTY READY.
006756  100003                          BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
006760  112777  000207  172204          MOVB    #207,@$TPB      ;PUSH A BELL AT THE TTY.
006766  032777  020000  172164  XBX:    BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
006774  001113                          BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
006776  021637  001136                  CMP     (SP),$ERRPC     ;WAS THIS ERROR FOUND LAST TIME?
007002  001404                          BEQ     1$              ;BR IF YES
```

```
007004  011637  001136                        MOV     (SP),$ERRPC         ;RECORD BEING HERE
007010  105037  001123                        CLRB    $ERFLG              ;PREPARE HEADER
007014  104407                       1$:      SAVOS                       ;SAVE ALL PROC REGISTERS
007016  011605                                MOV     (SP),R5             ;GET THE PC OF ERROR
007020  162705  000002                        SUB     #2,R5               ;GET ADDRESS OF TRAP CALL
007024  011504                                MOV     (R5),R4             ;GET ERROR INSTRUCTION
007026  110437  001134                        MOVB    R4,$ITEMB           ;COPY TEST NUMBER FOR APT HANDLING
007032  006304                                ASL     R4                  ;MULT BY TWO
007034  061504                                ADD     (R5),R4             ;DOUBLE IT
007036  006304                                ASL     R4                  ;MULT AGAIN
007040  042704  177001                        BIC     #177001,R4          ;CLEAR JUNK
007044  062704  026460                        ADD     #.ERRTAB,R4         ;GET POINTER
007050  012437  007174                        MOV     (R4)+,ERRMSG        ;GET ERROR MESSAGE
007054  012437  007206                        MOV     (R4)+,DATAHD        ;GET DATA HEADRER
007060  011437  007220                        MOV     (R4),DATABP         ;GET DATA TABLE
007064  105737  001123                        TSTB    $ERFLG              ;TYPE HEADER
007070  001403                                BEQ     TYPMSG              ;BR IF YES
007072  005737  007220                        TST     DATABP              ;DOES DATA TABLE EXIST?
007076  001044                                BNE     TYPDAT              ;BR IF YES.
007100  104402  001231               TYPMSG:  TYPE    ,$CRLF              ;TYPE A CARRIAGE RETURN
007104  104402  001231                        TYPE    ,$CRLF              ;AND TYPE ANOTHER
007110  005737  001362                        TST     LOCK
007114  001402                                BEQ     1$
007116  104402  010315                        TYPE    ,MASTEK
007122  104402  010303               1$:      TYPE    ,MTSTN
007126  104412  007352                        CNVRT   ,XTSTN              ;SHOW IT
007132  104402  010372                        TYPE    ,MERRPC             ;TYPE PC.
007136  104412  007344                        CNVRT   ,ERTABO             ;SHOW IT
007142  104402  010245                        TYPE    ,MCSRX
007146  104412  004742                        CNVRT   ,XCSR
007152  104402  001231                        TYPE    ,$CRLF              ;GIVE A CR/LF
007156  112737  177777  001123                MOVB    #-1,$ERFLG          ;NO MORE HEADER UNLESS NO DATA TABLE.
007164  005737  007174                        TST     ERRMSG              ;IS THERE AN ERROR MESSAGE?
007170  001402                                BEQ     WTBS.FM             ;BR IF NO.
007172  104402                                TYPE                        ;TYPE
007174  000000                       ERRMSG:  0                           ;     ERROR MESSAGE
007176                               WTBS.FM:                             ;
007176  005737  007206                        TST     DATAHD              ;DATA HEADER?
007202  001402                                BEQ     TYPDAT              ;BR IF NO
007204  104402                                TYPE                        ;TYPE
007206  000000                       DATAHD:  0                           ;     DATA HEADER
007210  005737  007220               TYPDAT:  TST     DATABP              ;DATA TABLE?
007214  001402                                BEQ     RESREG              ;BR IF NO.
007216  104411                                CONVRT                      ;SHOW
007220  000000                       DATABP:  0                           ;     DATA TABLE
007222  104410                       RESREG:  RESOS                       ;RESTORE PROC REGISTERS
007224  122737  000001  001254       HALTS:   CMPB    #APTENV,$ENV        ;IS APT RUNNING?
007232  001007                                BNE     2$                  ;SKIP APT CALL IF NOT
007234  113737  001134  007246                MOVB    $ITEMB,7$           ;COPY ERROR NUMBER
007242  004737  005600                        JSR     PC,$ATY4            ;CALL APT SERVICE
007246  000000                       7$:      .WORD   0                   ;ERROR NUMBER STUCK HERE
007250  000777                       8$:      BR      8$                  ;LOCK UP HERE
007252  022737  004726  000042       2$:      CMP     #$ENDAD,@#42        ;CHECK TO SEE IF IN ACT-11 MODE
007260  001403                                BEQ     1$                  ;IF SO, HANDLE ACCORDINGLY
007262  005777  171672                        TST     @SWR                ;HALT ON ERROR?
007266  100004                                BPL     EXITER              ;BR IF NO HALT ON ERROR
```

```
007270  016677  000002  171664  1$:      MOV     2(SP),@DISPLAY  ;SHOW ERROR PC IN DATA DISPLAY
007276  000000                           HALT                    ;HALT
007300  005237  001132          EXITER:  INC     $ERTTL          ;UPDATE ERROR COUNT
007304  032777  000400  171646           BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
007312  001007                           BNE     1$              ;BR IF YES
007314  032777  002000  171636           BIT     #SW10,@SWR      ;GOTO NEXT TEST?
007322  001407                           BEQ     2$              ;BR IF NO
007324  013737  001360  001126           MOV     NEXT,$LPADR     ;SET FOR NEXT TEST
007332  012706  001120          1$:      MOV     #STACK,SP       ;RESET SP
007336  000177  171564                   JMP     @$LPADR         ;GOTO SPECIFIED TEST
007342  000002                  2$:      RTI                     ;RETURN
007344  000001                  ERTABO:  1
007346     006     002                   .BYTE   6,2
007350  001402                           $AVPC
007352  000001                  XTSTN:   1
007354     002     002                   .BYTE   2,2
007356  001122                           $TSTNM
007360  022737  177570  001160  SERV.G:  CMP     #177570,SWR     ;IS THE SWITCH REGISTER HARDWIRED?
007366  001513                           BEQ     6$              ;IF SO, IGNORE ↑G
007370  017746  171572                   MOV     @$TKB,-(SP)     ;OTHERWISE, GET THE LAST CHARACTER TYPED
007374  042716  000200                   BIC     #BIT7,(SP)      ;STRIP PARITY(EIGHTH) BIT
007400  122726  000007                   CMPB    #7,(SP)+        ;IS IT ↑G?
007404  001104                           BNE     6$              ;IF NOT, IGNORE INPUT
007406  032777  004000  171550           BIT     #4000,@$TKS     ;RX BUSY?
007414  001361                           BNE     SERV.G          ;BR IF YES
007416  017737  171536  007640           MOV     @SWR,90$        ;SAVE (SWR).
007424  013777  007640  171526  1$:      MOV     90$,@SWR        ;
007432  104402  007620                   TYPE    ,89$
007436  104412  007632                   CNVRT   ,88$
007442  104402  007642                   TYPE    ,91$
007446  105777  171512                   TSTB    @$TKS           ;WAIT FOR DONE.
007452  100375                           BPL     .-4
007454  017746  171506                   MOV     @$TKB,-(SP)     ;
007460  042716  000200                   BIC     #BIT7,(SP)      ;
007464  122726  000015                   CMPB    #15,(SP)+       ;
007470  001450                           BEQ     5$              ;
007472  005077  171462                   CLR     @SWR            ;
007476  105777  171466          2$:      TSTB    @$TPS           ;
007502  100375                           BPL     .-4
007504  016677  177776  171460           MOV     -2(SP),@$TPB    ;
007512  000241                           CLC
007514  006177  171440                   ROL     @SWR            ;
007520  006177  171434                   ROL     @SWR            ;
007524  006177  171430                   ROL     @SWR            ;
007530  103735                           BCS     1$              ;ERROR
007532  026627  177776  000060           CMP     -2(SP),#60      ;
007540  002731                           BLT     1$              ;
007542  026627  177776  000067           CMP     -2(SP),#67      ;
007550  003325                           BGT     1$              ;
007552  042766  177770  177776           BIC     #↑C<7>,-2(SP)   ;
007560  056677  177776  171372           BIS     -2(SP),@SWR     ;
007566  105777  171372                   TSTB    @$TKS           ;
007572  100375                           BPL     .-4             ;
007574  017746  171366                   MOV     @$TKB,-(SP)
007600  042716  000200                   BIC     #BIT7,(SP)
007604  122726  000015                   CMPB    #15,(SP)+
007610  001332                                  BNE     2$                      ;
007612  104402  001231                  5$:     TYPE    ,$CRLF                  ;
007616  000207                          6$:     RTS     PC

007620  020200  051450  051127          89$:    .ASCIZ  <200>? (SWR)=/?
007626  036451  000057
                                        .EVEN
007632  000001                          88$:    1
007634     006     000                          .BYTE   6,0
007636  007640                                  90$
007640  000000                          90$:    .WORD   0
007642  036457  000057                  91$:    .ASCIZ  ?/=/?
                                        .EVEN
                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;;*****************************************************************
                                        ;POWER DOWN ROUTINE
007646  012737  010012  000024          $PWRDN: MOV     #$ILLUP,@#PWRVEC  ;;SET FOR FAST UP
007654  012737  000340  000026                  MOV     #340,@#PWRVEC+2   ;;PRIO:7
007662  010046                                  MOV     R0,-(SP)          ;;PUSH R0 ON STACK
007664  010146                                  MOV     R1,-(SP)          ;;PUSH R1 ON STACK
007666  010246                                  MOV     R2,-(SP)          ;;PUSH R2 ON STACK
007670  010346                                  MOV     R3,-(SP)          ;;PUSH R3 ON STACK
007672  010446                                  MOV     R4,-(SP)          ;;PUSH R4 ON STACK
007674  010546                                  MOV     R5,-(SP)          ;;PUSH R5 ON STACK
007676  017746  171256                          MOV     @SWR,-(SP)        ;;PUSH @SWR ON STACK
007702  010637  010016                          MOV     SP,$SAVR6         ;;SAVE SP
007706  012737  007720  000024                  MOV     #$PWRUP,@#PWRVEC  ;;SET UP VECTOR
007714  000000                                  HALT
007716  000776                                  BR      .-2               ;;HANG UP

                                        ;;*****************************************************************
                                        ;POWER UP ROUTINE
007720  012737  010012  000024          $PWRUP: MOV     #$ILLUP,@#PWRVEC  ;;SET FOR FAST DOWN
007726  013706  010016                          MOV     $SAVR6,SP         ;;GET SP
007732  005037  010016                          CLR     $SAVR6            ;;WAIT LOOP FOR THE TTY
007736  005237  010016                  1$:     INC     $SAVR6            ;;WAIT FOR THE INC
007742  001375                                  BNE     1$                ;;OF  WORD
007744  012677  171210                          MOV     (SP)+,@SWR        ;;POP STACK INTO @SWR
007750  012605                                  MOV     (SP)+,R5          ;;POP STACK INTO R5
007752  012604                                  MOV     (SP)+,R4          ;;POP STACK INTO R4
007754  012603                                  MOV     (SP)+,R3          ;;POP STACK INTO R3
007756  012602                                  MOV     (SP)+,R2          ;;POP STACK INTO R2
007760  012601                                  MOV     (SP)+,R1          ;;POP STACK INTO R1
007762  012600                                  MOV     (SP)+,R0          ;;POP STACK INTO R0
007764  012737  007646  000024                  MOV     #$PWRDN,@#PWRVEC  ;;SET UP THE POWER DOWN VECTOR
007772  012737  000340  000026                  MOV     #340,@#PWRVEC+2   ;;PRIO:7
010000  104402                                  TYPE                      ;;REPORT THE POWER FAILURE
010002  010020                          $PWRMG: .WORD   MPFAIL            ;;POWER FAIL MESSAGE POINTER
010004  012716                                  MOV     (PC)+,(SP)        ;;RESTART AT RESTART
010006  011434                          $PWRAD: .WORD   RESTART           ;;RESTART ADDRESS
010010  000002                                  RTI
010012  000000                          $ILLUP: HALT                      ;;THE POWER UP SEQUENCE WAS STARTED
010014  000776                                  BR      .-2               ;; BEFORE THE POWER DOWN WAS COMPLETE
010016  000000                          $SAVR6: 0                         ;;PUT THE SP HERE
010020  050200  051127  043040          MPFAIL: .ASCIZ  <200>/PWR FAILED. RESTART AT LAST TEST /
```

```
(2)  010063     200  047105  020104  MEPASS: .ASCIZ  <200>/END PASS DZDZA-D  /
(2)  010107     200  052522  047116  MR:     .ASCIZ  <200>/RUNNING   /
(2)  010123     200  051120  043517  MERR2:  .ASCIZ  <200>/PROGRAM INDICATES NO DEVICES PRESENT./
(2)  010172  044600  051516  043125  MERR3:  .ASCIZ  <200>/INSUFFICIENT DATA!/
(2)  010216  046200  041517  020113  MLOCK:  .ASCIZ  <200>/LOCK ON SELECTED TEST/
(2)  010245     103  051123  020072  MCSRX:  .ASCIZ   /CSR: /
(2)  010253     126  041505  020072  MVECX:  .ASCIZ  /VEC: /
(2)  010261     120  051501  042523  MPASSX: .ASCIZ  /PASSES: /
(2)  010272  051105  047522  051522  MERRX:  .ASCIZ  /ERRORS: /
(2)  010303     124  051505  020124  MTSTN:  .ASCIZ  /TEST NO: /
(2)  010315     052  000040          MASTEK: .ASCIZ  /* /
(2)  010320  051600  052105  051440  MNEW:   .ASCIZ  <200>/SET SWITCH REG TO DZ11'S DESIRED ACTIVE./
(2)  010372  041520  020072     000  MERRPC: .ASCIZ  /PC: /
(2)  010377     200  040515  020120  XHEAD:  .ASCIZ  <200>/MAP OF DZ11 STATUS/<200>
(2)  010424  044600  046114  043505  MBADLN: .ASCIZ  <200>/ILLEGAL ENTRY IN STAGGERED MODE/<200>
(2)                                  .EVEN
(2)  010466  000002                  XSTATQ: 2
     010470     006     003                  .BYTE   6,3
     010472  001220                          $TMP1
     010474     006     002                  .BYTE   6,2
     010476  001222                          $TMP2
                                     .EVEN
                                             ;THIS ROUTINE ESTABLISHES WHICH MAINTENANCE MODE THE DEVICE IS IN
                                             ;-------------------------------------------------------------
                                             ;
                                             ;E=EXTERNAL LOOP BACK
                                             ;I=INTERNAL LOOP BACK
                                             ;S=STAGGERED LOOP BACK
     010500  017605  000000          .SETFLG:MOV     @(SP),R5        ;PICK UP ADDRESS OF TAG
     010504  042737  000040  010620          BIC     #40,INBUF       ;STRIP LOWER CASE
     010512  122737  000105  010620          CMPB    #'E,INBUF       ;IS IT EXTERNAL LOOP BACK ?
     010520  001005                          BNE     4$              ;NO
     010522  013715  010612                  MOV     1$,(R5)         ;YES STORE INFO
     010526  105037  001417                  CLRB    MNTFLG          ;SET MAINT BIT =0
     010532  000422                          BR      7$              ;GET OUT
     010534  122737  000111  010620  4$:     CMPB    #'I,INBUF       ;IS IT INTERNAL LOOP BACK ?
     010542  001006                          BNE     5$              ;NO
     010544  013715  010614                  MOV     2$,(R5)         ;YES STORE INFO
     010550  112737  000010  001417          MOVB    #MAINT,MNTFLG   ;SET UP THE MAINTENANCE FLAG LOADER
     010556  000410                          BR      7$              ;GET OUT
     010560  122737  000123  010620  5$:     CMPB    #'S,INBUF       ;IS IT STAGGERED LOOP BACK ?
     010566  001007                          BNE     6$              ;WHAT ?
     010570  013715  010616                  MOV     3$,(R5)         ;YES STORE INFO
     010574  105037  001417                  CLRB    MNTFLG          ;ZERO BITS
     010600  062716  000002          7$:     ADD     #2,(SP)         ;POP AROUND
     010604  000002                          RTI
     010606  104404                  6$:     INSTER                  ;RETRY
     010610  000733                          BR      .SETFLG         ;DITTO
     010612  000200                  1$:     .WORD   200             ;EXTERNAL =  E
     010614  000000                  2$:     .WORD   0               ;INTERNAL = I
     010616  100000                  3$:     .WORD   100000          ;STAGGERED = S

                                             ;BUFFERS FOR INPUT-OUTPUT

     010620  000000                  INBUF:  0
             010662                  .=.+40
     010662  000000                  TEMP:   0
```

```
                010724                        .=.+40
        010724  000000                        MDATA:  0
                010766                        .=.+40

        010766  011637  011064                SET.PS: MOV     (SP),3$
        010772  162737  000002  011064                SUB     #2,3$
        011000  017737  000060  011066                MOV     @3$,4$
        011006  022737  106427  011066                CMP     #106427,4$
        011014  001003                                BNE     1$
        011016  011637  011064                        MOV     (SP),3$
        011022  000412                                BR      2$
        011024  022737  106437  011066  1$:           CMP     #106437,4$
        011032  001401                                BEQ     .+4
        011034  000000                                HALT            ;RESERVED INSTRUCTION NOT "MTPS"
        011036  011637  011064                        MOV     (SP),3$
        011042  017737  000016  011064                MOV     @3$,3$
        011050  062716  000002          2$:           ADD     #2,(SP)
        011054  017766  000004  000002                MOV     @3$,2(SP)
        011062  000002                                RTI
        011064  000000                  3$:           0
        011066  000000                  4$:           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.
                                                      ;

011070  005737  001404           CYCLE:   TST     DZACTV           ;ARE ANY DZ11'S TO BE TESTED?
011074  001004                            BNE     1$               ;BR IF OK.
011076  104402  010123                    TYPE    ,MERR2           ;NO DZ11'S SELECTED!!
011102  000000                            HALT                     ;STOP THE SHOW.
011104  000776                            BR      .-2              ;DISQUALIFY CONT. SW.
011106  013737  005234  001226   1$:      MOV     $MXCNT,$TIMES    ;RESTORE THE NUMBER OF ITERATIONS TO MAKE
011114  033737  001406  001404            BIT     RUN,DZACTV       ;IS THIS ONE "ACTIVE"
011122  001020                            BNE     2$               ;BR IF GOOD ONE FOUND.
011124  000241                            CLC
011126  006137  001406                    ROL     RUN              ;UPDATE POINTER
011132  005537  001406                    ADC     RUN              ;CATCH CARRY FROM RUN
011136  062737  000014  001412            ADD     #14,ACTIVE       ;UPDATE ADDRESS POINTER.
011144  022737  002000  001412            CMP     #DZ.END,ACTIVE   ;HAVE WE PASSED THE END OF THE MAP?
011152  001355                            BNE     1$               ;IF NO, KEEP GOING; NOT ALL TESTED FOR.
011154  012737  001500  001412            MOV     #DZ.MAP,ACTIVE   ;RESET ADDRESS POINTER.
011162  000751                            BR      1$               ;KEEP LOOKING FOR ACTIVE DZ11
011164  000241                   2$:      CLC
011166  006137  001406                    ROL     RUN              ;UPDATE POINTER.
011172  005537  001406                    ADC     RUN              ;CATCH CARRY.
011176  013700  001412                    MOV     ACTIVE,R0        ;GET ADDRESS POINTER.
011202  062737  000014  001412            ADD     #14,ACTIVE       ;UPDATE.
011210  022737  002000  001412            CMP     #DZ.END,ACTIVE
                                                                   ;ALL DONE?
011216  001003                            BNE     3$               ;BR IF NO.
011220  012737  001500  001412            MOV     #DZ.MAP,ACTIVE   ;RESTORE POINTER.
011226  012037  001310           3$:      MOV     (R0)+,$BASE      ;LOAD SYSTEM CTRL. REG
011232  012037  002072                    MOV     (R0)+,DZRIV      ;LOAD VECTOR
011236  012037  026454                    MOV     (R0)+,DZPRT      ;LOAD PRIORITY
011242  113737  026455  001414            MOVB    DZPRT+1,EIAFLG   ;EIA OR 20MA
011250  042737  100000  026454            BIC     #BIT15,DZPRT     ;CLEAR FLAG
011256  012037  001364                    MOV     (R0)+,LINE       ;SET UP LINE DZ LINES ACTIVE
011262  012037  001366                    MOV     (R0)+,PAR        ;SET UP PARAMETERIZATION
011266  012037  001370                    MOV     (R0)+,MODE       ;SET UP MAINTENANCE MODE
011272  004737  026246                    JSR     PC,DZLEV         ;SET UP
011276  005737  000042                    TST     @#42             ;ARE WE UNDER MONITOR CONTROL?
011302  001051                            BNE     4$               ;IF YES, SKIP THIS SETUP
011304  032777  000002  167646            BIT     #SW01,@SWR       ;IF SW01=1, GET STARTING TEST #
011312  001445                            BEQ     4$               ;BR IF NO TEST IS TO BE INPUTTED
011314  104402  001231           7$:      TYPE    ,$CRLF
                                                                   ;CALL THE STRING INPUT ROUTINE
011320  104403                            INSTR
011322  010303                            MTSTN                    ;POINTER TO MESSAGE TO BE PRINTED
011324  104405                            PARAM                    ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
011326  000001                            1                        ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
011330  001000                            1000                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
011332  001122                            $TSTNM                   ;POINTER TO MAP LOCATION TO BE FILLED
011334     000                            .BYTE   0                ;MASK OF INVALID BITS FOR THIS PARAMETER
011335     001                            .BYTE   1                ;NUMBER OF PARAMETERS TO STORE
```

```
011336  012700  012374                    MOV     #TST1,R0
011342  022710  000004            5$:     CMP     #4,(R0)
011346  001020                            BNE     6$
011350  022760  012737  000002            CMP     #12737,2(R0)
011356  001014                            BNE     6$
011360  023760  001122  000004            CMP     $TSTNM,4(R0)     ;IS THIS THE TEST ?
011366  001010                            BNE     6$               ;IF NOT, DON'T PROCESS NUMBER
011370  010037  001126                    MOV     R0,$LPADR        ;SAVE PC
011374  062737  000002  001126            ADD     #2,$LPADR        ;POP OVER SCOPE
011402  104402  001231                    TYPE    ,$CRLF
011406  000412                            BR      8$
011410  005720                    6$:     TST     (R0)+
011412  020027  022364                    CMP     R0,#TLAST+10
011416  001351                            BNE     5$
011420  104402  001230                    TYPE    ,$QUES
011424  000733                            BR      7$
011426  012737  012374  001126    4$:     MOV     #TST1,$LPADR     ;PREPARE TEST ADDRESS
011434                            8$:
011434  000177  167466            RESTART:JMP     @$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.

011440  012700  001500          SETAPT: MOV     #DZ.MAP,R0              ;POINT TO THE DEVICE MAP TABLE
011444  013701  001310                  MOV     $BASE,R1                ;BUILD DEVICE ADDRESSES IN R1
011450  013702  001304                  MOV     $VECT1,R2               ;BUILD DEVICE VECTORS IN R2
011454  042702  177007                  BIC     #↑C<770>,R2             ;STRIP AWAY OTHER INFORMATION

011460  113703  001305                  MOVB    $VECT1+1,R3             ;LOAD THE INTERRUPT PRIORITY FROM R3
011464  106003                          RORB    R3                      ;ALIGN THE NUMBER
011466  106003                          RORB    R3                      ;ALIGN THE NUMBER
011470  106003                          RORB    R3                      ;ALIGN THE NUMBER
011472  106003                          RORB    R3                      ;ALIGN THE NUMBER
011474  106003                          RORB    R3                      ;ALIGN THE NUMBER
011476  042703  177770                  BIC     #↑C<7>,R3               ;REMOVE ALL BUT BUS LEVEL NUMBER
011502  012704  001320                  MOV     #$DDW0,R4               ;POINT TO THE BEGINNING OF DEVICE PARAMETERS
011506  013705  001312                  MOV     $DEVM,R5                ;GET THE MAP OF ACTIVE DEVICES
011512  010537  001404                  MOV     R5,DZACTV               ;SAVE THE BIT MAP
011516  006005                  1$:     ROR     R5                      ;GET A DEVICE SELECTION BIT
011520  103407                          BCS     3$                      ;IF IT IS SELECTED, GO SET UP A MAP
011522  001425                          BEQ     5$                      ;IF NO MORE ARE SELECTED, GET OUT OF SETUP
011524  005724                          TST     (R4)+                   ;POINT TO NEXT DEVICE DESCRIPTOR
011526  062701  000010          2$:     ADD     #10,R1                  ;SET UP THE NEXT ADDRESS
011532  062702  000010                  ADD     #10,R2                  ;SET UP THE NEXT VECTOR GROUP
011536  000767                          BR      1$                      ;GO SEE IF MORE DEVICES REMAIN
011540  010120                  3$:     MOV     R1,(R0)+                ;LOAD DEVICE ADDRESS
011542  010220                          MOV     R2,(R0)+                ;LOAD THE VECTOR ADDRESS
011544  010320                          MOV     R3,(R0)+                ;LOAD THE INTERRUPT PRIORITY LEVEL
011546  013720  001314                  MOV     $CDW1,(R0)+             ;GET THE NUMBER OF LINES IN OPERATION
011552  012420                          MOV     (R4)+,(R0)+             ;LOAD DEVICE PARAMETERS
011554  100006                          BPL     4$                      ;IF 20MA MODE SELECTED, SET IT UP
011556  052760  100000  177772          BIS     #100000,-6(R0)          ;SET THE 20MA FLAG IN DZLVN
011564  042760  100000  177776          BIC     #100000,-2(R0)          ;CLEAR THE FLAG IN DZPARN
011572  005020                  4$:     CLR     (R0)+                   ;DEFAULT OPERATION TO INTERNAL MAINTENANCE MODE
011574  000754                          BR      2$                      ;GO BUILD THE NEXT ADDRESS
011576  012710  177777          5$:     MOV     #-1,(R0)                ;TERMINATE THE DEVICE MAP
011602  012737  001256  001160          MOV     #$SWREG,SWR             ;SET TO SOFTWARE APT SWITCH REGISTER
011610  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)
                                                ;*

011612                          AUTO.SIZE:
011612  000005                          RESET                           ;INSURE A BUS INIT.
011614  105337  001415                  DECB    INIFLG                  ;SHOW THAT I WAS HERE
011620  012702  001500          CSRMAP: MOV     #DZ.MAP,R2              ;LOAD MAP POINTER.
011624  012703  001320                  MOV     #$DDW0,R3               ;POINT TO ETABLE DEVICE DESCRIPTOR WORDS
011630  005022                  1$:     CLR     (R2)+                   ;ZERO ENTIRE MAP
011632  022702  002000                  CMP     #DZ.END,R2              ;ALL DONE?
```

```
011636  001374                            BNE     1$                  ;BR IF NO
011640  105037  001410                    CLRB    DZNUM               ;SET OCTAL NUMBER OF DZ11'S TO 0
011644  012702  001500                    MOV     #DZ.MAP,R2
011650  012701  160000                    MOV     #160000,R1          ;SET FOR FIRST ADDRESS TO BE TESTED
011654  012737  012174  000004            MOV     #6$,@#4             ;SET FOR NON-EXISTENT DEVICE TIME OUT
011662  052711  000040          2$:       BIS     #BIT5,(R1)          ;TRY TO SET MASTER SCAN ENABLE
011666  052761  000200  000004            BIS     #BIT7,4(R1)         ;TRY TO TRANSMIT ON LINE 7
011674  005000                            CLR     R0                  ;USE R0 AS A COUNTER
011676  005711                  7$:       TST     (R1)                ;HAS TRANSMITTER READY COME UP?
011700  100403                            BMI     8$                  ;IF SO, GO GET A FINAL CHECK
011702  005300                            DEC     R0                  ;REDUCE COUNT. TIME UP?
011704  001374                            BNE     7$                  ;IF NOT, KEEP WAITING
011706  000463                            BR      3$                  ;ASSUME IT'S NOT A DZ11
011710  032761  000200  000004  8$:       BIT     #BIT7,4(R1)         ;IS LINE 7 ENABLE STILL SET? IT SHOULD BE
011716  001457                            BEQ     3$                  ;IF IT'S NOT, ASSUME IT'S NOT A DZ11
011720  032711  000040                    BIT     #BIT5,(R1)          ;IS MASTER SCAN ENABLE STILL SET?
011724  001454                            BEQ     3$                  ;IF NOT, ASSUME IT'S NOT A DZ11
011726  005000                            CLR     R0
011730  052711  000020                    BIS     #20,(R1)            ;SET DEVICE CLEAR
011734  032711  000020                    BIT     #20,(R1)            ;SHOULD STAY SET FOR A WHILE IF DZ
011740  001446                            BEQ     3$                  ;BR IF NOT DZ11
011742  032711  000020                    BIT     #20,(R1)            ;WAIT FOR BIT TO CLEAR
011746  001404                            BEQ     .+12                ;BR WHEN CLEARED
011750  104414                            DELAY
011752  005200                            INC     R0
011754  001372                            BNE     .-12
011756  000437                            BR      3$                  ;BIT NOT CLEARED! MUST NOT BE DZ11
011760  005011                            CLR     (R1)                ;GET RID OF MASTER SCAN ENABLE
011762  005061  000004                    CLR     4(R1)               ;GET RID OF LINE 7 ENABLE
                                ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DZ11 CSR ADDRESS.
011766  010122                            MOV     R1,(R2)+            ;STORE CSR IN CORE TABLE.
011770  005722                            TST     (R2)+               ;POP OVER VECTOR STORE AREA
011772  012722  000005                    MOV     #5,(R2)+            ;SET THE DEFAULT BUS LEVEL
011776  052761  177400  000004            BIS     #177400,4(R1)       ;TRY TO SET ALL DTR BITS
012004  032761  177400  000004            BIT     #177400,4(R1)       ;IF ANY SET ASSUME EIA BOARD
012012  001003                            BNE     9$                  ;IF NONE SET ASSUME BOARD IS
012014  052762  100000  177776            BIS     #BIT15,-2(R2)       ;20 MA, SET 20 MA FLAG
012022  012722  000377          9$:       MOV     #377,(R2)+          ;SET THE DEFAULT LINE SELECTION PARAMETER
012026  012712  017470                    MOV     #17470,(R2)         ;SET THE DEFAULT PARAMETERS
012032  012223                            MOV     (R2)+,(R3)+         ;COPY PARAMETERS INTO ETABLE DESCRIPTOR
012034  005022                            CLR     (R2)+               ;SET THE DEFAULT MODE OF OPERATION
012036  012712  177777                    MOV     #-1,(R2)            ;TERMINATE LIST
012042  105237  001410                    INCB    DZNUM               ;UPDATE DEVICE COUNTER
012046  122737  000020  001410            CMPB    #20,DZNUM           ;ARE MAX. NO. OF DEV FOUND?
012054  001405                            BEQ     100$                ;YES DON'T LOOK FOR ANY MORE.
012056  062701  000010          3$:       ADD     #10,R1              ;UPDATE CSR POINTER ADDRESS
012062  022701  163700                    CMP     #163700,R1
012066  001275                            BNE     2$                  ;BR IF MORE ADDRESS TO CHECK.
012070                          100$:
012070  105737  001410                    TSTB    DZNUM               ;WERE ANY DZ11'S FOUND AT ALL?
012074  001432                            BEQ     5$                  ;ERROR AUTO SIZER FOUND NO DZ11'S IN THIS SYS.
012076  113701  001410                    MOVB    DZNUM,R1
012102  110137  001411                    MOVB    R1,SAVNUM           ;SAVE NUMBER OF DEVICES
012106  012737  000001  001404            MOV     #1,DZACTV
012114  005301                  4$:       DEC     R1
012116  001404                            BEQ     98$
```

```
012120  000261                                  SEC
012122  006137  001404                          ROL     DZACTV
012126  000772                                  BR      4$
012130  013737  001500  001310  98$:            MOV     DZCR0,$BASE             ;POINT TO THE ADDRESS OF FIRST DEVICE
012136  013737  001512  001314                  MOV     MANT0,$CDW1             ;INDICATE TO ETABLE WHAT MODE IS BEING USED
012144  012737  000006  000004  99$:            MOV     #6,@#4                  ;RESTORE TRAP VECTOR
012152  013737  001404  001312                  MOV     DZACTV,$DEVM            ;SAVE ACTIVE REGISTER
012160  000410                                  BR      VECMAP                  ;GO FIND THE VECTOR NOW.
012162  104402  010123          5$:             TYPE    ,MERR2                  ;NOTIFY OPR THAT NO DZ11'S FOUND.
012166  005000                                  CLR     R0                      ;MAKE DATA DISPLAY ZERO
012170  000000                                  HALT                            ;STOP THE SHOW
012172  000776                                  BR      .-2                     ;DISABLE CONT. SW.
012174  012716  012056          6$:             MOV     #3$,(SP)                ;ENTERED BY NON-EXISTENT TIME-OUT
012200  000002                                  RTI                             ;RETURN TO MAINSTREAM

012202  012737  000340  000022  VECMAP:         MOV     #340,@#22               ;SET IOT TRAP PRIORITY TO 7
012210  012737  012326  000020                  MOV     #4$,@#20                ;SET IOT TRAP VECTOR
012216  012702  001500                          MOV     #DZ.MAP,R2              ;SET SOFTWARE POINTER
012222  012700  000300                          MOV     #300,R0                 ;FLOATING VECTORS START HERE.
012226  012701  000302                          MOV     #302,R1                 ;PC OF IOT INSTR.
012232  010120                  1$:             MOV     R1,(R0)+                ;START FILLING VECTOR AREA
012234  012721  000004                          MOV     #4,(R1)+                ;WITH .+2; IOT
012240  022021                                  CMP     (R0)+,(R1)+             ;ADD 2 TO R0 +R1
012242  020127  001000                          CMP     R1,#1000                ;HAS THE VECTOR AREA BEEN EXCEEDED?
012246  101771                                  BLOS    1$                      ;BR IF MORE TO FILL
012250  013704  001404                          MOV     DZACTV,R4               ;STORE TEMPORARILY
012254  000241                  2$:             CLC
012256  006004                                  ROR     R4                      ;BRING OUT A BIT
012260  103036                                  BCC     5$                      ;BR IF ALL DONE
012262  106427  000000                          MTPS    #0                      ;ZERO CPU PRIO
012266  012772  040040  000000                  MOV     #BIT14+BIT5,@(R2)
012274  011201                                  MOV     (R2),R1                 ;GET CSR
012276  112761  000200  000004                  MOVB    #BIT7,4(R1)             ;SET THE TCR BIT!
                                                                                ;ATTEMPT TO FORCE AN INTERRUPT
012304  005200                                  INC     R0                      ;STALL
012306  001376                                  BNE     .-2                     ;        FOR TIME TO INTERRUPT
012310  012762  000300  000002                  MOV     #300,2(R2)              ;NO INTERRUPT ASSUME 300 AND FIX DZ11 LATER
012316  000005                  3$:             RESET                           ;INIT
012320  062702  000014                          ADD     #14,R2                  ;POP SOFTWARE POINTER
012324  000753                                  BR      2$                      ;KEEP GOING
012326  011662  000002          4$:             MOV     (SP),2(R2)              ;GET VECTOR ADDRESS
012332  162762  000010  000002                  SUB     #10,2(R2)               ;POINT BACK TO THE CORRECT VECTOR
012340  042762  000007  000002                  BIC     #7,2(R2)                ;CLEAR JUNK
012346  022626                                  POP2SP                          ;POP IOT JUNK OFF STACK
012350  012716  012316                          MOV     #3$,(SP)                ;SET FOR RETURN
012354  000002                                  RTI
012356  013737  001502  001304  5$:             MOV     DZVC0,$VECT1            ;COPY VECTOR OF FIRST DEVICE INTO ETABLE
012364  012737  004772  000020                  MOV     #.SCOPE,IOTVEC          ;RESTORE THE SCOPE TRAP
012372  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
                                        ;;*************************************************************
012374  000004                          TST1:   SCOPE
012376  012737  000001  001122                  MOV     #1,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
012404  012737  012564  001360                  MOV     #TST2,NEXT      ;POINT TO THE START OF THE NEXT TEST
012412  012737  012552  000004                  MOV     #5$,4           ;SET TRAP VECTOR
012420  012737  000340  000006                  MOV     #PR7,6          ;SET PRIORITY TO LEVEL 7
012426  012737  012434  001362                  MOV     #1$,LOCK        ;SET RETURN IF SW09=11
012434  013700  002042                  1$:     MOV     DZCSR,R0        ;SET ADDRESS TO TEST
012440  011001                                  MOV     (R0),R1         ;READ THE ADDRESS
012442  000240                                  NOP                     ;WASTE TIME
012444  005010                                  CLR     (R0)            ;WRITE THE ADDRESS
012446  000240                                  NOP                     ;WASTE TIME
012450  012737  012456  001362                  MOV     #2$,LOCK        ;SET RETURN ADDRESS FOR SW09
012456  013700  002046                  2$:     MOV     DZRBUF,R0       ;SET ADDRESS TO TEST
012462  011001                                  MOV     (R0),R1         ;READ THE ADDRESS
012464  000240                                  NOP
012466  005010                                  CLR     (R0)            ;WRITE THE ADDRESS
012470  000240                                  NOP                     ;WASTE TIME
012472  012737  012500  001362                  MOV     #3$,LOCK        ;SET RETURN ADDRESS FOR SW09
012500  013700  002056                  3$:     MOV     DZTCR,R0        ;SET ADDRESS TO TEST
012504  011001                                  MOV     (R0),R1         ;READ THE ADDRESS
012506  000240                                  NOP
012510  005010                                  CLR     (R0)            ;WRITE THE ADDRESS
012512  000240                                  NOP
012514  012737  012522  001362                  MOV     #4$,LOCK        ;SET RETURN ADDRESS
012522  013700  002062                  4$:     MOV     DZMSR,R0        ;SET ADDRESS TO TEST
012526  011001                                  MOV     (R0),R1         ;READ FROM ADDRESS
012530  000240                                  NOP
012532  005010                                  CLR     (R0)            ;WRITE THE ADDRESS
012534  000240                                  NOP
012536  012737  000006  000004                  MOV     #6,4            ;SET TRAP CATCHER BACK TO NORMAL
012544  005037  000006                          CLR     6
012550  104400                                  ADVANCE                 ;SCOPE THIS TEST
012552  011601                          5$:     MOV     (SP),R1         ;SAVE PC OF TRAP
012554  022626                                  CMP     (SP)+,(SP)+     ;POP TRAP OFF STACK
012556  104001                                  ERROR   1               ;*NO SLAVE SYNC RESPONSE.
012560  104401                                  SCOP1                   ;SW09=1?
012562  000111                                  JMP     (R1)            ;RTI
                                                ;************************ TEST 2 ******************************
                                                ;*THIS TEST PROVES THAT BIT "DCLR"
                                                ;*CAN BE SET AND THAT IT WILL CLEAR
                                                ;*BY ITSELF AFTER A PERIOD OF TIME.
                                        ;;*  TEST 2
                                        ;;*************************************************************
012564  000004                          TST2:   SCOPE
012566  012737  000002  001122                  MOV     #2,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
012574  012737  012650  001360                  MOV     #TST3,NEXT      ;POINT TO THE START OF THE NEXT TEST
012602  013700  002042                          MOV     DZCSR,R0        ;SET POINTER
012606  012705  000020                          MOV     #DCLR,R5        ;SET DCLR
012612  010510                                  MOV     R5,(R0)         ;WRITE DCLR INTO DZCSR
```

```
 012614  011004                              MOV     (R0),R4         ;READ BACK DZCSR
 012616  020504                              CMP     R5,R4           ;DZCSR OK?
 012620  001401                              BEQ     1$              ;IF IT IS SET SKIP THE ERROR CALL
 012622  104002                              ERROR   2               ;*DCLR SHOULD BE SET..MOMENTARILY
                                 ;NOW LETS WATCH IT DISAPPEAR
 012624  005002                      1$:     CLR     R2              ;SET COUNTER TO 0
 012626  005005                              CLR     R5              ;SET EXPECTED TO 0
 012630  005003                              CLR     R3              ;DUAL LOOP COUNTER
 012632  011004                      2$:     MOV     (R0),R4         ;IS DCLR CLEAR?
 012634  001405                              BEQ     3$              ;IF YES , GO TO THE NEXT TEST
 012636  005203                              INC     R3              ;IF NO,COUNT 1 OF 65535 TICKS
                                                                     ;THE WORD CREATED BY THE IMMEDIATE 0 WILL BE
                                                                     ;THE COUNTER
 012640  001374                              BNE     2$              ;HAS THE TIME EXPIRED? IF NO, GO TEST BIT AGAIN
 012642  005302                              DEC     R2              ;HAS THE TOTAL TIME EXPIRED?
 012644  001372                              BNE     2$              ;IF NO, CHECK THE BIT AGAIN
 012646  104002                              ERROR   2               ;*DCLR FAILED TO CLEAR
 012650                              3$:
                                             ;*********************** TEST 3 ****************************
                                             ;*TEST TO VERIFY THAT BIT "MAINT" CAN
                                             ;*BE SET. THEN VERIFY THAT BIT "MAINT" CAN
                                             ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                             ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                                             ;*CLEARED BY A "DEVICE CLEAR"
                                     ;;*  TEST 3
                                     ;;**********************************************************
 012650  000004                      TST3:   SCOPE
 012652  012737  000003  001122              MOV     #3,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
 012660  012737  012742  001360              MOV     #TST4,NEXT      ;POINT TO THE START OF THE NEXT TEST
 012666  013700  002042                      MOV     DZCSR,R0        ;GET BASE ADDRESS
 012672  012705  000010                      MOV     #MAINT,R5       ;SET BIT
 012676  010510                              MOV     R5,(R0)         ;SET SET IN DEVICE
 012700  011004                              MOV     (R0),R4         ;READ THE BIT FROM DEVICE
 012702  020504                              CMP     R5,R4           ;WAS BIT SET?
 012704  001401                              BEQ     1$              ;BR IF YES
 012706  104002                              ERROR   2               ;*BIT R/W FAILURE
 012710  040510                      1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
 012712  011004                              MOV     (R0),R4         ;READ DEVICE
 012714  001404                              BEQ     2$              ;BR IF BITS WERE CLEARED.
 012716  010546                              MOV     R5,-(SP)        ;SAVE THE BIT
 012720  005005                              CLR     R5              ;SET EXPECTED RESULTS TO 0
 012722  104002                              ERROR   2               ;*BIT FAILED TO CLEAR
 012724  012605                              MOV     (SP)+,R5        ;RESTORE THE BIT.
 012726  010510                      2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
 012730  104413                              DEVICE.CLR              ;ISSUE DEVICE CLEAR
 012732  011004                              MOV     (R0),R4         ;READ THE BIT.
 012734  001402                              BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
 012736  005005                              CLR     R5              ;SET EXPECTED TO ZERO
 012740  104002                              ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
 012742                              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 CAN BE
                                             ;*CLEARED BY A "DEVICE CLEAR"
```

```
                                        ;;*  TEST 4
                                        ;;*************************************************************
012742  000004                          TST4:   SCOPE
012744  012737  000004  001122                  MOV     #4,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
012752  012737  013034  001360                  MOV     #TST5,NEXT      ;POINT TO THE START OF THE NEXT TEST
012760  013700  002042                          MOV     DZCSR,R0        ;GET BASE ADDRESS
012764  012705  000040                          MOV     #MSENAB,R5      ;SET BIT
012770  010510                                  MOV     R5,(R0)         ;SET SET IN DEVICE
012772  011004                                  MOV     (R0),R4         ;READ THE BIT FROM DEVICE
012774  020504                                  CMP     R5,R4           ;WAS BIT SET?
012776  001401                                  BEQ     1$              ;BR IF YES
013000  104002                                  ERROR   2               ;*BIT R/W FAILURE
013002  040510                          1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
013004  011004                                  MOV     (R0),R4         ;READ DEVICE
013006  001404                                  BEQ     2$              ;BR IF BITS WERE CLEARED.
013010  010546                                  MOV     R5,-(SP)        ;SAVE THE BIT
013012  005005                                  CLR     R5              ;SET EXPECTED RESULTS TO 0
013014  104002                                  ERROR   2               ;*BIT FAILED TO CLEAR
013016  012605                                  MOV     (SP)+,R5        ;RESTORE THE BIT.
013020  010510                          2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
013022  104413                                  DEVICE.CLR              ;ISSUE DEVICE CLEAR
013024  011004                                  MOV     (R0),R4         ;READ THE BIT.
013026  001402                                  BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
013030  005005                                  CLR     R5              ;SET EXPECTED TO ZERO
013032  104002                                  ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
013034                                  3$:
                                                ;************************ TEST 5 ****************************
                                                ;*TEST TO VERIFY THAT BIT "SILOEN" CAN
                                                ;*BE SET. THEN VERIFY THAT BIT "SILOEN" CAN
                                                ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                                ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                                                ;*CLEARED BY A "DEVICE CLEAR"
                                        ;;*  TEST 5
                                        ;;*************************************************************
013034  000004                          TST5:   SCOPE
013036  012737  000005  001122                  MOV     #5,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
013044  012737  013126  001360                  MOV     #TST6,NEXT      ;POINT TO THE START OF THE NEXT TEST
013052  013700  002042                          MOV     DZCSR,R0        ;GET BASE ADDRESS
013056  012705  010000                          MOV     #SILOEN,R5      ;SET BIT
013062  010510                                  MOV     R5,(R0)         ;SET SET IN DEVICE
013064  011004                                  MOV     (R0),R4         ;READ THE BIT FROM DEVICE
013066  020504                                  CMP     R5,R4           ;WAS BIT SET?
013070  001401                                  BEQ     1$              ;BR IF YES
013072  104002                                  ERROR   2               ;*BIT R/W FAILURE
013074  040510                          1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
013076  011004                                  MOV     (R0),R4         ;READ DEVICE
013100  001404                                  BEQ     2$              ;BR IF BITS WERE CLEARED.
013102  010546                                  MOV     R5,-(SP)        ;SAVE THE BIT
013104  005005                                  CLR     R5              ;SET EXPECTED RESULTS TO 0
013106  104002                                  ERROR   2               ;*BIT FAILED TO CLEAR
013110  012605                                  MOV     (SP)+,R5        ;RESTORE THE BIT.
013112  010510                          2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
013114  104413                                  DEVICE.CLR              ;ISSUE DEVICE CLEAR
013116  011004                                  MOV     (R0),R4         ;READ THE BIT.
013120  001402                                  BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
013122  005005                                  CLR     R5              ;SET EXPECTED TO ZERO
```

```
013124  104002                                ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
013126                                3$:
                                              ;*********************** TEST 6 *****************************
                                              ;*TEST TO VERIFY THAT BIT "RIE" CAN
                                              ;*BE SET. THEN VERIFY THAT BIT "RIE" CAN
                                              ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                              ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                                              ;*CLEARED BY A "DEVICE CLEAR"
                                      ;;*  TEST 6
                                      ;;*************************************************************
013126  000004                        TST6:   SCOPE
013130  012737  000006  001122                MOV     #6,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
013136  012737  013220  001360                MOV     #TST7,NEXT      ;POINT TO THE START OF THE NEXT TEST
013144  013700  002042                        MOV     DZCSR,R0        ;GET BASE ADDRESS
013150  012705  000100                        MOV     #RIE,R5 ;SET BIT
013154  010510                                MOV     R5,(R0)         ;SET SET IN DEVICE
013156  011004                                MOV     (R0),R4         ;READ THE BIT FROM DEVICE
013160  020504                                CMP     R5,R4           ;WAS BIT SET?
013162  001401                                BEQ     1$              ;BR IF YES
013164  104002                                ERROR   2               ;*BIT R/W FAILURE
013166  040510                        1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
013170  011004                                MOV     (R0),R4         ;READ DEVICE
013172  001404                                BEQ     2$              ;BR IF BITS WERE CLEARED.
013174  010546                                MOV     R5,-(SP)        ;SAVE THE BIT
013176  005005                                CLR     R5              ;SET EXPECTED RESULTS TO 0
013200  104002                                ERROR   2               ;*BIT FAILED TO CLEAR
013202  012605                                MOV     (SP)+,R5        ;RESTORE THE BIT.
013204  010510                        2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
013206  104413                                DEVICE.CLR              ;ISSUE DEVICE CLEAR
013210  011004                                MOV     (R0),R4         ;READ THE BIT.
013212  001402                                BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
013214  005005                                CLR     R5              ;SET EXPECTED TO ZERO
013216  104002                                ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
013220                                3$:
                                              ;*********************** TEST 7 *****************************
                                              ;*TEST TO VERIFY THAT BIT "TIE" CAN
                                              ;*BE SET. THEN VERIFY THAT BIT "TIE" CAN
                                              ;*BE CLEARED (WRITTEN TO A ZERO). AND FINALLY
                                              ;*VERIFY THAT AFTER BEING SET AGAIN IT CAN BE
                                              ;*CLEARED BY A "DEVICE CLEAR"
                                      ;;*  TEST 7
                                      ;;*************************************************************
013220  000004                        TST7:   SCOPE
013222  012737  000007  001122                MOV     #7,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
013230  012737  013312  001360                MOV     #TST10,NEXT     ;POINT TO THE START OF THE NEXT TEST
013236  013700  002042                        MOV     DZCSR,R0        ;GET BASE ADDRESS
013242  012705  040000                        MOV     #TIE,R5 ;SET BIT
013246  010510                                MOV     R5,(R0)         ;SET SET IN DEVICE
013250  011004                                MOV     (R0),R4         ;READ THE BIT FROM DEVICE
013252  020504                                CMP     R5,R4           ;WAS BIT SET?
013254  001401                                BEQ     1$              ;BR IF YES
013256  104002                                ERROR   2               ;*BIT R/W FAILURE
013260  040510                        1$:     BIC     R5,(R0)         ;CLEAR THE BIT.
013262  011004                                MOV     (R0),R4         ;READ DEVICE
013264  001404                                BEQ     2$              ;BR IF BITS WERE CLEARED.
013266  010546                                MOV     R5,-(SP)        ;SAVE THE BIT
```

```
013270  005005                          CLR     R5              ;SET EXPECTED RESULTS TO 0
013272  104002                          ERROR   2               ;*BIT FAILED TO CLEAR
013274  012605                          MOV     (SP)+,R5        ;RESTORE THE BIT.
013276  010510                  2$:     MOV     R5,(R0)         ;SET THE BIT AGAIN
013300  104413                          DEVICE.CLR              ;ISSUE DEVICE CLEAR
013302  011004                          MOV     (R0),R4         ;READ THE BIT.
013304  001402                          BEQ     3$              ;BR IF BIT CLEARED BY INIT (DEVICE CLEAR)
013306  005005                          CLR     R5              ;SET EXPECTED TO ZERO
013310  104002                          ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
013312                          3$:
                                        ;************************ TEST 10 *****************************
                                        ;*THIS TESTS THAT ALL OF THE FOLLOWING
                                        ;*BITS CAN BE: SET, CLEARED, CLEARED BY "DEVICE CLEAR "
                                        ;*BITS TESTED ARE:
                                        ;* TCR0, TCR1, TCR2, TCR3, TCR4, TCR5, TCR6, TCR7
                                ;;*  TEST 10
                                ;;**********************************************************
013312  000004                  TST10:  SCOPE
013314  012737  000010  001122          MOV     #10,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
013322  012737  013450  001360          MOV     #TST11,NEXT     ;POINT TO THE START OF THE NEXT TEST
013330  013700  002056                  MOV     DZTCR,R0        ;SET DEVICE ADDRESS
013334  012705  000001                  MOV     #TCR0,R5        ;SET EXPECTED RESULTS
013340  012737  013346  001362          MOV     #1$,LOCK        ;SET FOR SW09
013346  010510                  1$:     MOV     R5,(R0)         ;SET THE BIT
013350  011004                          MOV     (R0),R4         ;READ THE BIT FROM THE DEVICE
013352  042704  177400                  BIC     #↑C<377>,R4     ;CLEAR HIGH BYTE
013356  020504                          CMP     R5,R4           ;WAS BIT OK?
013360  001401                          BEQ     2$              ;BR IF YES
013362  104002                          ERROR   2               ;*BIT FAILED TO SET.
013364  040510                  2$:     BIC     R5,(R0)         ;CLEAR THE BIT
013366  011004                          MOV     (R0),R4         ;READ THE REGISTER
013370  042704  177400                  BIC     #↑C<377>,R4     ;CLEAR HIGH BYTE
013374  005704                          TST     R4              ;BITS CLEAR?
013376  001404                          BEQ     3$              ;BR IF YES
013400  010546                          MOV     R5,-(SP)        ;SAVE GOOD RESULTS
013402  005005                          CLR     R5              ;SET EXPECTED TO 0
013404  104002                          ERROR   2               ;*REPORT BIT NOT CLEAR
013406  012605                          MOV     (SP)+,R5        ;RESTORE R5
013410  010510                  3$:     MOV     R5,(R0)         ;SET THE BIT AGAIN.
013412  104413                          DEVICE.CLR              ;ISSUE DEVICE CLEAR
013414  011004                          MOV     (R0),R4         ;READ THE REGISTER
013416  042704  177400                  BIC     #↑C<377>,R4     ;CLEAR HIGH BYTE
013422  005704                          TST     R4              ;BITS CLEAR?
013424  001404                          BEQ     4$              ;BR IF YES
013426  010546                          MOV     R5,-(SP)        ;SAVE GOOD RESULTS
013430  005005                          CLR     R5              ;SET EXPECTED TO 0
013432  104002                          ERROR   2               ;*REPORT BIT NOT CLEAR
013434  012605                          MOV     (SP)+,R5        ;RESTORE R5
013436  104401                  4$:     SCOP1                   ;LOCK ON BIT? SET SW09=1
013440  106305                          ASLB    R5              ;CHANGE TO NEXT BIT
013442  001341                          BNE     1$              ;CONTINUE TESTING
013444  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
                                         ;:*******************************************************************
013450  000004                           TST11:  SCOPE
013452  012737  000011  001122                   MOV     #11,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
013460  012737  013632  001360                   MOV     #TST12,NEXT         ;POINT TO THE START OF THE NEXT TEST
013466  013700  002056                           MOV     DZTCR,R0            ;SET DEVICE ADDRESS
013472  012705  000400                           MOV     #DTR0,R5            ;SET EXPECTED RESULTS
013476  012737  013514  001362                   MOV     #1$,LOCK            ;SET FOR SW09
013504  105737  001414                           TSTB    EIAFLG              ;20MA OR EIA
013510  100001                                   BPL     1$                  ;BR IF EIA
013512  104400                                   ADVANCE                     ;EXIT TEST
013514  010510                           1$:     MOV     R5,(R0)             ;SET THE BIT
013516  011004                                   MOV     (R0),R4             ;READ THE BIT FROM THE DEVICE
013520  105004                                   CLRB    R4                  ;CLEAR LOW BYTE
013522  020504                                   CMP     R5,R4               ;WAS BIT OK?
013524  001401                                   BEQ     2$                  ;BR IF YES
013526  104002                                   ERROR   2                   ;*BIT FAILED TO SET.
013530  040510                           2$:     BIC     R5,(R0)             ;CLEAR THE BIT
013532  011004                                   MOV     (R0),R4             ;READ THE REGISTER
013534  105004                                   CLRB    R4                  ;CLEAR LOW BYTE
013536  005704                                   TST     R4                  ;BITS CLEAR?
013540  001404                                   BEQ     3$                  ;BR IF YES
013542  010546                                   MOV     R5,-(SP)            ;SAVE GOOD RESULTS
013544  005005                                   CLR     R5                  ;SET EXPECTED TO 0
013546  104002                                   ERROR   2                   ;*REPORT BIT NOT CLEAR
013550  012605                                   MOV     (SP)+,R5            ;RESTORE R5
013552  010510                           3$:     MOV     R5,(R0)             ;SET THE BIT AGAIN.
013554  104413                                   DEVICE.CLR                  ;ISSUE DEVICE CLEAR
013556  011004                                   MOV     (R0),R4             ;READ THE REGISTER
013560  105004                                   CLRB    R4                  ;CLEAR LOW BYTE
013562  030510                                   BIT     R5,(R0)             ;WAS BIT CLEARED BY DEVICE.CLR?
013564  001001                                   BNE     4$                  ;BR IF NO (IT SHOULDN'T BE CLEAR)
013566  104002                                   ERROR   2                   ;*BIT CLEARED BY DEVICE.CLR
013570  104401                           4$:     SCOP1                       ;LOCK ON BIT? SW09=1
013572  006305                                   ASL     R5                  ;CHANGE TO NEXT BIT
013574  001347                                   BNE     1$                  ;IF NOT DONE LOOP
013576  012710  177400                           MOV     #177400,(R0)        ;SET ALL DTR BITS
013602  005005                                   CLR     R5                  ;CLEAR LOCATION FOR ERROR PRINTOUT
013604  005227  000000                   5$:     INC     #0                  ;ACT DELAY LOOP FOR
013610  001375                                   BNE     5$                  ;RESET INSTRUCTION
013612  000005                                   RESET                       ;ISSUE A BUS INIT
013614  011004                                   MOV     (R0),R4             ;READ REGISTER
013616  105004                                   CLRB    R4                  ;CLEAR LOW BYTE
013620  005704                                   TST     R4                  ;DTR BITS CLEAR?
013622  001401                                   BEQ     .+4                 ;IF YES CONTINUE
013624  104002                                   ERROR   2                   ;IF NO PRINT ERROR
013626  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
                                        ;;*************************************************************
 013632  000004                         TST12:  SCOPE
 013634  012737  000012  001122                 MOV     #12,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
 013642  012737  013750  001360                 MOV     #TST13,NEXT         ;POINT TO THE START OF THE NEXT TEST
 013650  013700  002042                         MOV     DZCSR,R0            ;SET ADDRESS TO R0
 013654  005005                                 CLR     R5                  ;SET EXPECTED TO 0
 013656  012710  027607                         MOV     #RDONE+BIT11+BIT10+BIT9+BIT8+BIT2+BIT1+BIT0+SILOAL,(R0)
                                                                            ;WRITE THE BITS
 013662  011004                                 MOV     (R0),R4             ;READ BACK THE BITS
 013664  001401                                 BEQ     1$                  ;BR IF NONE ARE SET.
 013666  104002                                 ERROR   2                   ;*BITS WERE SET.
 013670  012710  100000                 1$:     MOV     #TRDY,(R0)          ;ATTEMPT TO WRITE TRDY
 013674  011004                                 MOV     (R0),R4             ;READ TRDY
 013676  001401                                 BEQ     2$                  ;BR IF NOT SET
 013700  104002                                 ERROR   2                   ;*
 013702  012705  100000                 2$:     MOV     #TRDY,R5            ;SET EXPECTED BIT
 013706  005077  166140                         CLR     @DZLPR              ;LOAD LINE 0
 013712  052777  000001  166136                 BIS     #TCR0,@DZTCR        ;SET TCR BIT
 013720  052710  000040                         BIS     #MSENAB,(R0)        ;
 013724  052705  000040                         BIS     #MSENAB,R5          ;SET SCAN ENABLE
 013730  005002                                 CLR     R2                  ;SET COUNTER TO ZERO
 013732  011004                         3$:     MOV     (R0),R4             ;READ THE REGISTER
 013734  020504                                 CMP     R5,R4               ;BIT SET?
 013736  001404                                 BEQ     4$                  ;BR IF YES
 013740  104414                                 DELAY                       ;STALL TIME
 013742  005202                                 INC     R2                  ;UPDATE COUNTER
 013744  001372                                 BNE     3$                  ;BR IF COUNTER NOT DONE.
 013746  104002                                 ERROR   2                   ;*TRDY NOT SET!
 013750                                 4$:
                                                ;*********************** TEST 13 *****************************
                                                ;*THIS TEST PERFORMS RESET TESTING AND
                                                ;*TESTING OF READ ONLY AND WRITE ONLY BITS
                                                ;* IN REGISTER DZCSR
                                                ;*VERIFY THAT "TIE", "SILOEN", "RIE", "MSENAB", "MAINT"
                                                ;*ARE THE ONLY R/W BITS IN THE DZCSR.
                                                ;*THEN SET "DCLR" AND VERIFY THEY ARE CLEARED
                                        ;;*  TEST 13
                                        ;;*************************************************************
 013750  000004                         TST13:  SCOPE
 013752  012737  000013  001122                 MOV     #13,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
 013760  012737  014034  001360                 MOV     #TST14,NEXT         ;POINT TO THE START OF THE NEXT TEST
 013766  104413                                 DEVICE.CLR
 013770  013700  002042                         MOV     DZCSR,R0            ;SET UP FOR ERROR MESSAGE
 013774  012710  177757                         MOV     #^C<DCLR>,(R0)      ;TRY TO WRITE
 014000  012705  050150                         MOV     #TIE!SILOEN!RIE!MSENAB!MAINT,R5 ;MAKE EXPECTED
 014004  011004                                 MOV     (R0),R4             ;ACTUAL
 014006  020405                                 CMP     R4,R5               ;CMP EXPECTED VS ACTUAL
 014010  001401                                 BEQ     1$                  ;YES
 014012  104002                                 ERROR   2                   ;*NO
 014014  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.
 014020  052710  000020                         BIS     #DCLR,(R0)          ;DEVICE MASTER RESET
 014024  011004                                 MOV     (R0),R4             ;ACTUAL
 014026  020405                                 CMP     R4,R5               ;CMP ACTUAL VS EXPECTED
```

```
014030  001401                                  BEQ     2$                      ;YES
014032  104002                                  ERROR   2                       ;*NO
014034                                  2$:
                                                ;*********************** TEST 14 ******************************
                                                ;*THIS TEST PERFORMS RESET TESTING AND
                                                ;*TESTING OF READ ONLY REGISTER DZRBUF
                                                ;*AND TESTING OF WRITE ONLY REGISTER DZLPR
                                        ;;*  TEST 14
                                        ;;*************************************************************
014034  000004                          TST14:  SCOPE
014036  012737  000014  001122                  MOV     #14,$TSTNM              ;LOAD THE NUMBER OF THIS TEST
014044  012737  014124  001360                  MOV     #TST15,NEXT             ;POINT TO THE START OF THE NEXT TEST
014052  104413                                  DEVICE.CLR                      ;CLEAR DZ11
014054  013700  002046                          MOV     DZRBUF,R0                       ;SET UP FOR ERROR MESSAGE
014060  011005                                  MOV     (R0),R5                 ;SET EXPECTED
014062  012777  177777  165762                  MOV     #-1,@DZLPR              ;TRY TO WRITE ALL 1'S
014070  011004                                  MOV     (R0),R4                 ;ACTUAL
014072  042705  104000                          BIC     #DVALID!BIT11,R5        ;DITTO
014076  020405                                  CMP     R4,R5                   ;CMP ACTUAL VS EXPECTED
014100  001401                                  BEQ     1$                      ;IF YES,GO CONTINUE PROCESSING
014102  104002                                  ERROR   2                       ;*ERROR- BIT PATTERN NOT CORRECT
014104  010403                          1$:     MOV     R4,R3                   ;GET A COPY OF THE ACTUAL BIT PATTERN
014106  005103                                  COM     R3                      ;GET THE LOGICAL INVERSE OF THE BIT PATTERN
014110  010377  165736                          MOV     R3,@DZLPR               ;TRY TO WRITE
014114  011004                                  MOV     (R0),R4                 ;ACTUAL
014116  020405                                  CMP     R4,R5                   ;CMP ACTUAL VS EXPECTED
014120  001401                                  BEQ     2$                      ;IF YES, GET OUT OF THIS TEST
014122  104002                                  ERROR   2                       ;*NO
014124                                  2$:
                                                ;*********************** TEST 15 ******************************
                                                ;*THIS TEST PERFORMS RESET TESTING AND
                                                ;*TESTING OF READ ONLY REGISTER DZMSR
                                                ;*AND TESTING OF WRITE ONLY REGISTER DZTDR
                                        ;;*  TEST 15
                                        ;;*************************************************************
014124  000004                          TST15:  SCOPE
014126  012737  000015  001122                  MOV     #15,$TSTNM              ;LOAD THE NUMBER OF THIS TEST
014134  012737  014210  001360                  MOV     #TST16,NEXT             ;POINT TO THE START OF THE NEXT TEST
014142  104413                                  DEVICE.CLR                      ;CLEAR DZ11
014144  013700  002062                          MOV     DZMSR,R0                        ;SET UP FOR ERROR MESSAGE
014150  011005                                  MOV     (R0),R5                 ;SET EXPECTED
014152  012777  177777  165706                  MOV     #-1,@DZTDR              ;TRY TO WRITE ALL 1'S
014160  011004                                  MOV     (R0),R4                 ;ACTUAL
014162  020405                                  CMP     R4,R5                   ;CMP ACTUAL VS EXPECTED
014164  001401                                  BEQ     1$                      ;IF YES,GO CONTINUE PROCESSING
014166  104002                                  ERROR   2                       ;*ERROR- BIT PATTERN NOT CORRECT
014170  010403                          1$:     MOV     R4,R3                   ;GET A COPY OF THE ACTUAL BIT PATTERN
014172  005103                                  COM     R3                      ;GET THE LOGICAL INVERSE OF THE BIT PATTERN
014174  010377  165666                          MOV     R3,@DZTDR               ;TRY TO WRITE
014200  011004                                  MOV     (R0),R4                 ;ACTUAL
014202  020405                                  CMP     R4,R5                   ;CMP ACTUAL VS EXPECTED
014204  001401                                  BEQ     2$                      ;IF YES, GET OUT OF THIS TEST
014206  104002                                  ERROR   2                       ;*NO
014210                                  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
                                              ;;*****************************************************************
014210  000004                                TST16:  SCOPE
014212  012737  000016  001122                        MOV     #16,$TSTNM              ;LOAD THE NUMBER OF THIS TEST
014220  012737  014404  001360                        MOV     #TST17,NEXT             ;POINT TO THE START OF THE NEXT TEST
014226  012737  014300  001362                        MOV     #1$,LOCK                ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
014234  105737  001414                                TSTB    EIAFLG                  ;EIA OR 20MA?
014240  100001                                        BPL     10$                     ;BR IF EIA
014242  104400                                        ADVANCE                         ;EXIT TEST
014244  013700  002062                        10$:    MOV     DZMSR,R0                ;SET REGISTER
014250  104413                                        DEVICE.CLR                      ;INIT DZ11
014252  005003                                        CLR     R3                      ;ZERO LINE NUMBER
014254  012702  000001                                MOV     #1,R2                   ;SET POINTER
014260  005737  001370                                TST     MODE                    ;ARE WE IN STAGGERED MODE?
014264  100405                                        BMI     1$                      ;YES WE ARE!
014266  013737  001360  001126                        MOV     NEXT,$LPADR             ;LEAVE THIS TEST! NOT STAGGERED
014274  000177  164626                                JMP     @$LPADR                 ;EXIT
014300  130237  001364                        1$:     BITB    R2,LINE                 ;TEST THIS LINE?
014304  001004                                        BNE     3$                      ;YES
014306  005203                                2$:     INC     R3                      ;LINE #
014310  106302                                        ASLB    R2                      ;GET NEXT LINE
014312  103372                                        BCC     1$                      ;KEEP TESTING
014314  104400                                        ADVANCE                         ;ADVANCE THIS TEST
014316  010204                                3$:     MOV     R2,R4                   ;SAVE BINARY BIT FOR LINE #
014320  032703  000001                                BIT     #BIT0,R3                ;GET STAGGERED COMPANION LINE
014324  001402                                        BEQ     4$                      ;BR IF LINE EVEN
014326  006204                                        ASR     R4                      ;ADJUST LINE
014330  000401                                        BR      5$                      ;
014332  006304                                4$:     ASL     R4                      ;ADJUST LINE
014334  005005                                5$:     CLR     R5                      ;SET EXPECTED
014336  150405                                        BISB    R4,R5                   ;
014340  000305                                        SWAB    R5                      ;
014342  150405                                        BISB    R4,R5                   ;
014344  150277  165510                                BISB    R2,@HDZTCR              ;SET DTR
014350  104414                                        DELAY                           ;CABLE DELAY
014352  011004                                        MOV     (R0),R4                 ;READ MSR REGISTER
014354  020504                                        CMP     R5,R4                   ;OK?
014356  001401                                        BEQ     6$                      ;YES
014360  104002                                        ERROR   2                       ;*ERROR IN RING OR CARRIER
014362  140277  165472                        6$:     BICB    R2,@HDZTCR              ;CLEAR DTR
014366  104414                                        DELAY                           ;CABLE DELAY
014370  011004                                        MOV     (R0),R4                 ;READ MSR
014372  001402                                        BEQ     7$                      ;BR IF THEY CLEARED
014374  005005                                        CLR     R5                      ;SET EXPECTED TO 0
014376  104002                                        ERROR   2                       ;*BITS NOT CLEARED
014400  104401                                7$:     SCOP1                           ;LOCK ON SIGNAL?
```

```
014402  000741                                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 SPECIFIED BY SWR03=1
                                      ;*AND SWR00=1 AT START TIME OR ALTERING
                                      ;*STATUS MAP.
                              ;;*  TEST 17
                              ;;*****************************************************
014404  000004                TST17:  SCOPE
014406  012737  000017  001122        MOV     #17,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
014414  012737  014542  001360        MOV     #TST20,NEXT     ;POINT TO THE START OF THE NEXT TEST
014422  012737  014456  001362        MOV     #3$,LOCK        ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
014430  105737  001370                TSTB    MODE            ;EXTERNAL?
014434  100401                        BMI     2$              ;BR IF YES
014436  104400                1$:     ADVANCE                 ;EXIT TEST
014440  105737  001414        2$:     TSTB    EIAFLG          ;YOU BETTER BE IN
014444  100774                        BMI     1$              ;EIA MODE FOR THIS TEST.
014446  013700  002062                MOV     DZMSR,R0        ;SET REGISTER
014452  012702  000001                MOV     #1,R2           ;SET LINE POINTER
014456  130237  001364        3$:     BITB    R2,LINE         ;LINE SELECTED?
014462  001003                        BNE     5$              ;BR IF YES
014464  106302                4$:     ASLB    R2              ;NEXT LINE
014466  103373                        BCC     3$              ;CONTINUE TEST
014470  104400                        ADVANCE                 ;ADVANCE THIS TEST
014472  005005                5$:     CLR     R5              ;SET EXPECTED
014474  150205                        BISB    R2,R5           ;
014476  000305                        SWAB    R5              ;
014500  150205                        BISB    R2,R5           ;
014502  150277  165352                BISB    R2,@HDZTCR      ;SET DTR
014506  104414                        DELAY                   ;CABLE DELAY
014510  011004                        MOV     (R0),R4         ;READ MSR
014512  020504                        CMP     R5,R4           ;BITS OK?
014514  001401                        BEQ     6$              ;BR IF YES
014516  104002                        ERROR   2               ;CARRIER OR RING ERROR
014520  140277  165334        6$:     BICB    R2,@HDZTCR      ;CLEAR DTR
014524  104414                        DELAY                   ;CABLE DELAY
014526  011004                        MOV     (R0),R4         ;READ MSR
014530  001402                        BEQ     7$              ;BR IF BITS CLEARED
014532  005005                        CLR     R5              ;CLEAR EXPECTED LOC.
014534  104002                        ERROR   2               ;BITS NOT CLEARED.
014536  104401                7$:     SCOP1                   ;LOCK ON LINE?
014540  000751                        BR      4$              ;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
                              ;;*****************************************************
```

```
014542  000004                    TST20:  SCOPE
014544  012737  000020  001122            MOV     #20,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
014552  012737  014666  001360            MOV     #TST21,NEXT         ;POINT TO THE START OF THE NEXT TEST
014560  104413                            DEVICE.CLR                  ;ISSUE A "DEVICE CLEAR" (RESET)
014562  013700  002042                    MOV     DZCSR,R0            ;SET POINTER
014566  012705  100040                    MOV     #MSENAB!TRDY,R5     ;START THE EXPECTED LINE NUMBER AT 0
014572  005037  001372                    CLR     SAVLIN              ;SET UP FOR ERROR PRINTOUTS
014576  012702  000001                    MOV     #1,R2               ;USING R2 AS A BIT POINTER, POINT TO LINE 0
014602  130237  001364            1$:     BITB    R2,LINE             ;IS THIS LINE SELECTED?
014606  001420                            BEQ     5$                  ;IF NO, SKIP THE STARTUP
014610  050277  165242            2$:     BIS     R2,@DZTCR           ;SET THE GO BIT FOR THIS LINE
014614  052710  000040                    BIS     #MSENAB,(R0)        ;START THE SCANNER
014620  005004                            CLR     R4                  ;SET FOR DELAY
014622  032710  100000            3$:     BIT     #TRDY,(R0)          ;TX READY?
014626  001004                            BNE     4$                  ;BR IF YES
014630  104414                            DELAY                       ;DELAY
014632  005204                            INC     R4                  ;COUNTER
014634  001372                            BNE     3$                  ;BR IF <>0!
014636  104003                            ERROR   3                   ;*TX NOT READY!
014640  011004                    4$:     MOV     (R0),R4             ;GET THE LINE POINTED TO BY THE SCANNER
014642  020405                            CMP     R4,R5               ;IS THE LINE NUMBER WHAT IT SHOULD BE?
014644  001401                            BEQ     5$                  ;IF YES,GO WORK ON THE NEXT LINE
014646  104002                            ERROR   2                   ;*LINE NUMBER DID NOT MATCH TCR BIT
014650  062705  000400            5$:     ADD     #400,R5             ;POINT TO THE NEXT EXPECTED LINE
014654  104413                            DEVICE.CLR                  ;ISSUE A "DEVICE CLEAR" (RESET)
014656  005237  001372                    INC     SAVLIN              ;ADJUST FOR NEXT LINE
014662  106302                            ASLB    R2                  ;POINT TO THE NEXT LINE.ARE ALL LINES TESTED?
014664  103346                            BCC     1$                  ;IF NOT, GO DO THE NEXT LINE
014666                            6$:
                                          ;*********************** TEST 21 *****************************
                                          ;*TEST TO TRANSMIT ONE CHAR AND
                                          ;*RECEIVE ONE CHAR ON ONE LINE
                                          ;*AT A TIME. THE CHAR IS "252" AND
                                          ;*ALL SELECTED LINES WILL BE TURNED ON
                                          ;*ONE AT A TIME. THIS IS THE FIRST TIME ANY
                                          ;*DATA IS CHECKED IN THE RECEIVER.
                                          ;*USING SWITCH NINE WITH THIS TEST CREATES A TIGHT SCOPE LOOP
                                          ;*WHICH TRANSMITS A STEADY STREAM OF CHARACTERS.
                                  ;;*  TEST 21
                                  ;;***************************************************************
014666  000004                    TST21:  SCOPE
014670  012737  000021  001122            MOV     #21,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
014676  012737  015202  001360            MOV     #TST22,NEXT         ;POINT TO THE START OF THE NEXT TEST
014704  012737  015160  001362            MOV     #16$,LOCK           ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
014712  104417                            DCLASM                      ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
014714  013701  001366                    MOV     PAR,R1              ;PICK UP PARAMETERS
014720  012702  000001                    MOV     #1,R2               ;PICK UP INIT POINTER
014724  030237  001364            1$:     BIT     R2,LINE             ;SHOULD THIS LINE BE SET UP ?
014730  001402                            BEQ     2$                  ;NO
014732  010177  165114                    MOV     R1,@DZLPR           ;SET UP LINE PARAMETERS
014736  005201                    2$:     INC     R1                  ;POSITION POINTER TO THE NEXT LINE
014740  106302                            ASLB    R2                  ;GOT 'EM ALL ?
014742  103370                            BCC     1$                  ;IF NO, GO SET UP THE NEXT LINE
014744  005037  001372                    CLR     SAVLIN              ;CLEAR LINE # INDICATOR
014750  012702  000001                    MOV     #1,R2               ;LINE POINTER
014754  052777  000040  165060            BIS     #MSENAB,@DZCSR      ;START SCANNER
```

```
014762  030237  001364                  3$:     BIT     R2,LINE                 ;VALID LINE ?
014766  001462                                  BEQ     14$                     ;NO SET UP NEXT LINE
014770  010277  165062                          MOV     R2,@DZTCR               ;SET TCR BIT
014774  032777  000200  165040          4$:     BIT     #RDONE,@DZCSR           ;IS REC DONE = 0 ?
015002  001401                                  BEQ     5$                      ;IF YES, ALLOW TIME FOR TRDY TO SET
015004  104020                                  ERROR   20                      ;*REC DONE SHOULD = 0
015006  005005                          5$:     CLR     R5
015010  032777  100000  165024          6$:     BIT     #TRDY,@DZCSR
015016  001004                                  BNE     7$
015020  104414                                  DELAY
015022  105205                                  INCB    R5
015024  001371                                  BNE     6$
015026  104003                                  ERROR   3                       ;*TRDY FAILED TO SET!
015030  112777  000252  165030          7$:     MOVB    #252,@DZTDR             ;LOAD CHARACTER
015036  013705  001372                          MOV     SAVLIN,R5               ;MAKE EXPECTED LINE #
015042  105737  001371                          TSTB    MODE+1                  ;IS THIS TEST IN STAGGERED MODE?
015046  001406                                  BEQ     10$                     ;IF NOT, SKIP STAGGERED SETUP

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

015050  006205                                  ASR     R5                      ;GET THE LAST BIT INTO THE CARRY BIT
015052  103402                                  BCS     8$                      ;IF IT IS SET, GO CLEAR IT
015054  000261                                  SEC                             ;IF IT IS CLEAR SET IT HERE
015056  000401                                  BR      9$                      ;SKIP THE CLEARING
015060  000241                          8$:     CLC                             ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
015062  006105                          9$:     ROL     R5                      ;GET THE NEW BIT BACK INTO R5
015064  000305                          10$:    SWAB    R5                      ;MOVE THE LINE NUMBER TO THE UPPER BYTE
015066  152705  000252                          BISB    #252,R5                 ;ADD CHARACTER
015072  052705  100000                          BIS     #DVALID,R5              ;ADD DATA VALID
015076  005003                                  CLR     R3
015100  032777  000200  164734          11$:    BIT     #RDONE,@DZCSR
015106  001004                                  BNE     12$
015110  104414                                  DELAY
015112  005203                                  INC     R3
015114  001371                                  BNE     11$
015116  104004                                  ERROR   4                       ;*RDONE FAILED TO SET!
015120  017704  164722                  12$:    MOV     @DZRBUF,R4              ;LOAD THE VALUE ACTUALLY RECEIVED
015124  020405                                  CMP     R4,R5                   ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
015126  001401                                  BEQ     13$                     ;IF YES, GO DO THE NEXT LINE
015130  104006                                  ERROR   6                       ;*NO DATA/CONTENTS DID NOT COMPARE
015132  104401                          13$:    SCOP1                           ;CHECK TO SEE IF SWITCH NINE IS SET
015134  040277  164716                  14$:    BIC     R2,@DZTCR               ;CLEAR TCR BIT FOR THAT LINE.
015140  005237  001372                  15$:    INC     SAVLIN                  ;INC EXPECTED LINE
015144  013700  001372                          MOV     SAVLIN,R0               ;SET UP CHARACTER OFFSET
015150  006300                                  ASL     R0                      ;MAKE THE OFFSET A POWER OF TWO
015152  106302                                  ASLB    R2                      ;SHIFT THE LINE POINTER. ARE WE ALL DONE?
015154  103302                                  BCC     3$                      ;IF NO, GO AROUND AGAIN FOR NEXT LINE
015156  104400                                  ADVANCE                         ;GO TO NEXT TEST

                                        ;TIGHT SCOPE LOOP FOR THIS TEST. LOOP TRANSMITS CHARACTERS ONLY

015160  032777  100000  164654          16$:    BIT     #TRDY,@DZCSR            ;IS TRANSMITTER READY?
015166  001774                                  BEQ     16$                     ;IF NOT, WAIT FOR IT
015170  112777  000252  164670                  MOVB    #252,@DZTDR             ;LOAD THE CHARACTER
015176  104401                                  SCOP1                           ;LOOP AGIN IF SW09=1
015200  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   BASED UPON VALID LINES)
                                                    ;*THIS IS THE FIRST TIME THAT ALL DATA IS CHECKED
                                            ;;*  TEST 22
                                            ;;***********************************************************
015202  000004                              TST22:  SCOPE
015204  012737  000022  001122                      MOV     #22,$TSTNM              ;LOAD THE NUMBER OF THIS TEST
015212  012737  015530  001360                      MOV     #TST23,NEXT             ;POINT TO THE START OF THE NEXT TEST
015220  012737  015334  001362                      MOV     #4$,LOCK                ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
015226  104417                                      DCLASM                          ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
015230  013701  001366                              MOV     PAR,R1                  ;PICK UP PARAMETERS
015234  012702  000001                              MOV     #1,R2                   ;PICK UP INIT POINTER
015240  030237  001364                      1$:     BIT     R2,LINE                 ;SHOULD THIS LINE BE SET UP ?
015244  001402                                      BEQ     2$                      ;NO
015246  010177  164600                              MOV     R1,@DZLPR               ;SET UP LINE PARAMETERS
015252  005201                              2$:     INC     R1                      ;POSITION POINTER TO THE NEXT LINE
015254  106302                                      ASLB    R2                      ;GOT 'EM ALL ?
015256  103370                                      BCC     1$                      ;IF NO, GO SET UP THE NEXT LINE
015260  005037  001372                              CLR     SAVLIN                  ;CLEAR LINE # INDICATOR
015264  012700  001422                              MOV     #TD0,R0                 ;POINT TO THE DATA AREA
015270  005020                                      CLR     (R0)+                   ;CLEAR A DATA WORD
015272  022700  001462                              CMP     #STOP,R0                ;FINISHED ?
015276  001374                                      BNE     .-6                     ;NO
015300  005000                                      CLR     R0                      ;CLEAR OFFSET
015302  013737  002046  001400                      MOV     DZRBUF,REGIST           ;SAVE FOR ERROR MSG
015310  012702  000001                              MOV     #1,R2                   ;LINE POINTER
015314  052777  000040  164520                      BIS     #MSENAB,@DZCSR          ;START SCANNER
015322  030237  001364                      3$:     BIT     R2,LINE                 ;VALID LINE ?
015326  001465                                      BEQ     14$                     ;NO SET UP NEXT LINE
015330  010277  164522                              MOV     R2,@DZTCR               ;SET TCR BIT
015334  032777  000200  164500              4$:     BIT     #RDONE,@DZCSR           ;IS REC DONE = 0 ?
015342  001401                                      BEQ     5$                      ;IF YES, ALLOW TIME FOR TRDY TO SET
015344  104020                                      ERROR   20                      ;*REC DONE SHOULD = 0
015346  005005                              5$:     CLR     R5
015350  032777  100000  164464              6$:     BIT     #TRDY,@DZCSR
01[illegible]  001004                                      BNE     7$
01[illegible]  104414                                      DELAY
015362  105205                                      INCB    R5
015364  001371                                      BNE     6$
015366  104003                                      ERROR   3                       ;*TRDY FAILED TO SET!
015370  116077  001422  164470              7$:     MOVB    TD0(R0),@DZTDR          ;LOAD CHARACTER
015376  013705  001372                              MOV     SAVLIN,R5               ;MAKE EXPECTED LINE #
015402  105737  001371                              TSTB    MODE+1                  ;IS THIS TEST IN STAGGERED MODE?
015406  001406                                      BEQ     10$                     ;IF NOT, SKIP STAGGERED SETUP

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

015410  006205                                      ASR     R5                      ;GET THE LAST BIT INTO THE CARRY BIT
015412  103402                                      BCS     8$                      ;IF IT IS SET, GO CLEAR IT
015414  000261                                      SEC                             ;IF IT IS CLEAR SET IT HERE
015416  000401                                      BR      9$                      ;SKIP THE CLEARING
015420  000241                              8$:     CLC                             ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
015422  006105                              9$:     ROL     R5                      ;GET THE NEW BIT BACK INTO R5
```

```
015424  000305                          10$:    SWAB    R5                      ;MOVE THE LINE NUMBER TO THE UPPER BYTE
015426  156005  001422                          BISB    TDO(R0),R5              ;ADD CHARACTER
015432  052705  100000                          BIS     #DVALID,R5              ;ADD DATA VALID
015436  005003                                  CLR     R3
015440  032777  000200  164374          11$:    BIT     #RDONE,@DZCSR
015446  001004                                  BNE     12$
015450  104414                                  DELAY
015452  005203                                  INC     R3
015454  001371                                  BNE     11$
015456  104004                                  ERROR   4                       ;*RDONE FAILED TO SET!
015460  017704  164362                  12$:    MOV     @DZRBUF,R4              ;LOAD THE VALUE ACTUALLY RECEIVED
015464  020405                                  CMP     R4,R5                   ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
015466  001401                                  BEQ     13$                     ;IF YES, GO DO THE NEXT LINE
015470  104006                                  ERROR   6                       ;*NO DATA/CONTENTS DID NOT COMPARE
015472  104401                          13$:    SCOP1                           ;CHECK TO SEE IF SWITCH NINE IS SET
015474  105260  001422                          INCB    TDO(R0)                 ;INCREMENT BINARY PATTERN FOR THIS LINE
015500  001315                                  BNE     4$                      ;GO 'ROUND AGAIN FOR NEXT CHARACTER
015502  040277  164350                  14$:    BIC     R2,@DZTCR               ;CLEAR TCR BIT FOR THAT LINE.
015506  005237  001372                  15$:    INC     SAVLIN                  ;INC EXPECTED LINE
015512  013700  001372                          MOV     SAVLIN,R0               ;SET UP CHARACTER OFFSET
015516  006300                                  ASL     R0                      ;MAKE THE OFFSET A POWER OF TWO
015520  106302                                  ASLB    R2                      ;SHIFT THE LINE POINTER. ARE WE ALL DONE?
015522  103277                                  BCC     3$                      ;IF NO, GO AROUND AGAIN FOR NEXT LINE
015524  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 FLAG "PARITY ERRORS"
                                                ;*ONLY ONE LINE AT A TIME WILL BE EXERCISED.
                                                ;*THIS TEST WILL NOT BE  EXERCISED UNLESS
                                                ;*CONNECTED BY AN H325, H3271, OR H3190 CONNECTOR
                                        ;;*  TEST 23
                                        ;;*********************************************************
015530  000004                          TST23:  SCOPE
015532  012737  000023  001122                  MOV     #23,$TSTNM              ;LOAD THE NUMBER OF THIS TEST
015540  012737  016006  001360                  MOV     #TST24,NEXT             ;POINT TO THE START OF THE NEXT TEST
015546  012737  015644  001362                  MOV     #3$,LOCK                ;SET FOR LOOP
015554  005737  001370                          TST     MODE                    ;ARE WE RUNNING IN INTERNAL MODE?
015560  001510                                  BEQ     12$                     ;IF SO, SKIP THIS TEST
015562  104417                                  DCLASM                          ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
015564  013701  001366                          MOV     PAR,R1                  ;PICK UP PARAMETERS
015570  052701  000300                          BIS     #ODDPAR!PARITY,R1  ;FORCE ODD PARITY
015574  012700  000001                          MOV     #1,R0                   ;PICK UP INIT POINTER
015600  030037  001364                  1$:     BIT     R0,LINE                 ;SHOULD THIS LINE BE SET UP ?
015604  001402                                  BEQ     2$                      ;IF NOT,DON'T SET IT UP
015606  010177  164240                          MOV     R1,@DZLPR               ;OTHERWISE, SET UP LINE PARAMETERS
015612  005201                          2$:     INC     R1
015614  106300                                  ASLB    R0                      ;GOT 'EM ALL ?
015616  103370                                  BCC     1$                      ;NO
015620  005037  001372                          CLR     SAVLIN                  ;CLEAR LINE #
015624  012702  000001                          MOV     #1,R2                   ;LINE POINTER
015630  052777  000040  164204                  BIS     #MSENAB,@DZCSR          ;SET MASTER SCAN ENABLE
015636  013737  002046  001400                  MOV     DZRBUF,REGIST           ;SAVE FOR ERRR MESSAGE
```

```
015644  030237  001364          3$:     BIT     R2,LINE
015650  001446                          BEQ     11$
015652  010277  164200                  MOV     R2,@DZTCR       ;SET TCR BIT
015656  110277  164206                  MOVB    R2,@HDZTDR      ;SET BREAK BIT
015662  112777  000377  164176  4$:     MOVB    #377,@DZTDR     ;LOAD CHARACTER
015670  013705  001372                  MOV     SAVLIN,R5       ;MAKE EXPECTED DATA
015674  105737  001371                  TSTB    MODE+1          ;IS THIS TEST IN STAGGERED MODE?
015700  001406                          BEQ     7$              ;IF NOT, SKIP STAGGERED SETUP

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

015702  006205                          ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
015704  103402                          BCS     5$              ;IF IT IS SET, GO CLEAR IT
015706  000261                          SEC                     ;IF IT IS CLEAR SET IT HERE
015710  000401                          BR      6$              ;SKIP THE CLEARING
015712  000241                  5$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
015714  006105                  6$:     ROL     R5              ;GET THE NEW BIT BACK INTO R5
015716  000305                  7$:     SWAB    R5              ;PUT LINE NUMBER IN UPPER BYTE
015720  052705  130000                  BIS     #DVALID!PARER!FRMERR,R5 ;ADD EXPECTED
015724  005004                          CLR     R4
015726  032777  000200  164106  8$:     BIT     #RDONE,@DZCSR
015734  001004                          BNE     9$
015736  104414                          DELAY
015740  005204                          INC     R4
015742  001371                          BNE     8$
015744  104004                          ERROR   4               ;*RDONE FAILED TO SET!
015746  017704  164074          9$:     MOV     @DZRBUF,R4      ;ACTUAL
015752  020405                          CMP     R4,R5           ;CMP ACTUAL VS EXPECTED. DO THEY MATCH?
015754  001401                          BEQ     10$             ;IF YES, GO CLEAN UP
015756  104006                          ERROR   6               ;*DATA/CONTENTS FAILED TO COMPARE
015760  105077  164104          10$:    CLRB    @HDZTDR         ;CLEAR BREAK BITS
015764  104401                          SCOP1                   ;LOOP?
015766  005237  001372          11$:    INC     SAVLIN          ;INC LINE #
015772  040277  164060                  BIC     R2,@DZTCR       ;CLEAR TCR BIT
015776  106302                          ASLB    R2
016000  103321                          BCC     3$
016002  005037  001362          12$:    CLR     LOCK            ;MAKE SURE LOCK IS CLEAR FOR NEXT TEST
                                ;************************ TEST 24 ****************************
                                ;* THIS TEST VERIFIES THAT THE DEVICE DOES NOT INTERRUPT
                                ;*WHILE THE PROCESSOR STATUS IS SET EXACTLY
                                ;*TO WHAT THE DZ11 PRIORITY IS SET TO.
                                ;*DEFAULT PRIORITY IS AT  5 (240).
                        ;;*  TEST 24
                        ;;*************************************************************
016006  000004                  TST24:  SCOPE
016010  012737  000024  001122          MOV     #24,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
016016  012737  016316  001360          MOV     #TST25,NEXT     ;POINT TO THE START OF THE NEXT TEST
016024  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
016026  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
016032  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
016036  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
016042  001402                          BEQ     2$              ;NO
016044  010177  164002                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
016050  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
016052  106302                          ASLB    R2              ;GOT 'EM ALL ?
016054  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
```

```
016056  005037  001372                   CLR     SAVLIN              ;CLEAR LINE # INDICATOR
016062  106437  026454                   MTPS    @#DZPRT             ;SET CPU STATUS TO DZ11 PRIO,
016066  113777  001364  163762           MOVB    LINE,@DZTCR         ;ENABLE THE VALID LINES
016074                          3$:
016074  012777  016164  163774           MOV     #6$,@DZTIV          ;SET UP THE TRANSMITTER INTERRUPT VECTOR
016102  012777  016172  163762           MOV     #7$,@DZRIV          ;SET UP THE RECEIVER INTERRUPT VECTOR
016110  013777  026454  163756           MOV     DZPRT,@DZRIS        ;SET THE INTERRUPT VECTOR STATUS
016116  013777  026454  163754           MOV     DZPRT,@DZTIS        ;SET TRANSMITTER INTERRUPT PRIORITY
016124  052777  040040  163710           BIS     #TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
016132  005005                           CLR     R5
016134  032777  100000  163700  4$:      BIT     #TRDY,@DZCSR
016142  001403                           BEQ     5$
016144  000240                           NOP
016146  000240                           NOP
016150  000412                           BR      8$
016152  104414                  5$:      DELAY
016154  005205                           INC     R5
016156  001366                           BNE     4$
016160  104003                           ERROR   3                   ;*TRDY NOT SET!
016162  000405                           BR      8$
016164  104010                  6$:      ERROR   10                  ;*TRANSMITTER SHOULD NOT INTERRUPT
016166  022626                           CMP     (SP)+,(SP)+         ;POP FOR FAKE RTI
016170  000402                           BR      8$                  ;CONTINUE TEST
016172  104012                  7$:      ERROR   12                  ;*RECEIVER SHOULD NOT INTERRUPT
016174  022626                           CMP     (SP)+,(SP)+         ;POP FOR FAKE RTI
016176  042777  040000  163636  8$:      BIC     #TIE,@DZCSR         ;RESET TRANSMITTER INTERRUPT ENABLE
016204  012777  016302  163664           MOV     #11$,@DZTIV         ;SET UP THE TRANSMITTER INTERRUPT VECTOR
016212  012777  016310  163652           MOV     #12$,@DZRIV         ;SET UP THE RECEIVER INTERRUPT VECTOR
016220  013777  026454  163646           MOV     DZPRT,@DZRIS        ;SET THE INTERRUPT VECTOR STATUS
016226  013777  026454  163644           MOV     DZPRT,@DZTIS        ;SET TRANSMITTER INTERRUPT PRIORITY
016234  052777  000140  163600           BIS     #RIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
016242  113777  001422  163616           MOVB    TDO,@DZTDR          ;PUT ANY RANDOM CHARACTER IN TRANSMITTER BUFFER
016250  005005                           CLR     R5
016252  032777  000200  163562  9$:      BIT     #RDONE,@DZCSR
016260  001403                           BEQ     10$
016262  000240                           NOP
016264  000240                           NOP
016266  000412                           BR      13$
016270  104414                  10$:     DELAY
016272  005205                           INC     R5
016274  001366                           BNE     9$
016276  104004                           ERROR   4                   ;*NO RX DONE! (NOT SET)
016300  000405                           BR      13$                 ;CONTINUE TEST
016302  104010                  11$:     ERROR   10                  ;*TRANSMITTER SHOULD NOT INTERRUPT
016304  022626                           CMP     (SP)+,(SP)+         ;POP FOR FAKE RTI
016306  000402                           BR      13$                 ;CONT TEST
016310  104012                  12$:     ERROR   12                  ;*RECEIVER SHOULD NOT INTERRUPT
016312  022626                           CMP     (SP)+,(SP)+         ;POP FOR FAKE RTI
016314                          13$:
016314  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
```

H07

```
                                       ;;**************************************************************
016316  000004                         TST25:  SCOPE
016320  012737  000025  001122                 MOV     #25,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
016326  012737  016644  001360                 MOV     #TST26,NEXT         ;POINT TO THE START OF THE NEXT TEST
016334  104417                                 DCLASM                      ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
016336  013701  001366                         MOV     PAR,R1              ;PICK UP PARAMETERS
016342  012702  000001                         MOV     #1,R2               ;PICK UP INIT POINTER
016346  030237  001364                 1$:     BIT     R2,LINE             ;SHOULD THIS LINE BE SET UP ?
016352  001402                                 BEQ     2$                  ;NO
016354  010177  163472                         MOV     R1,@DZLPR           ;SET UP LINE PARAMETERS
016360  005201                         2$:     INC     R1                  ;POSITION POINTER TO THE NEXT LINE
016362  106302                                 ASLB    R2                  ;GOT 'EM ALL ?
016364  103370                                 BCC     1$                  ;IF NO, GO SET UP THE NEXT LINE
016366  005037  001372                         CLR     SAVLIN              ;CLEAR LINE # INDICATOR
016372  106437  026456                         MTPS    @#LESS1             ;MAKE CPU ONE LEVEL LOWER THAN DZ11
016376  113777  001364  163452                 MOVB    LINE,@DZTCR         ;ENABLE THE VALID LINES
016404                                 3$:
016404  012777  016476  163464                 MOV     #6$,@DZTIV          ;SET UP THE TRANSMITTER INTERRUPT VECTOR
016412  012777  016514  163452                 MOV     #7$,@DZRIV          ;SET UP THE RECEIVER INTERRUPT VECTOR
016420  013777  026454  163446                 MOV     DZPRT,@DZRIS        ;SET THE INTERRUPT VECTOR STATUS
016426  013777  026454  163444                 MOV     DZPRT,@DZTIS        ;SET TRANSMITTER INTERRUPT PRIORITY
016434  052777  040040  163400                 BIS     #TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
016442  005005                                 CLR     R5
016444  032777  100000  163370         4$:     BIT     #TRDY,@DZCSR
016452  001404                                 BEQ     5$
016454  000240                                 NOP
016456  000240                                 NOP
016460  104007                                 ERROR   7                   ;*TRANSMITTER FAILED TO INTERRUPT
016462  000416                                 BR      8$
016464  104414                         5$:     DELAY
016466  005205                                 INC     R5
016470  001365                                 BNE     4$
016472  104003                                 ERROR   3                   ;*TRDY NOT SET!
016474  000411                                 BR      8$
016476  022626                         6$:     POP2SP                      ;REMOVE THE INTERRUPT FROM THE STACK
016500  042777  040000  163334                 BIC     #TIE,@DZCSR         ;DON'T LET ANY MORE INTERRUPTS OCCUR
016506  106437  026456                         MTPS    @#LESS1             ;MAKE CPU ONE LEVEL LOWER THAN DZ11
016512  000402                                 BR      8$                  ;RETURN TO THE NORMAL FLOW
016514  104012                         7$:     ERROR   12                  ;*RECEIVER SHOULD NOT INTERRUPT
016516  022626                                 CMP     (SP)+,(SP)+         ;POP FOR FAKE RTI
016520  042777  040000  163314         8$:     BIC     #TIE,@DZCSR         ;RESET TRANSMITTER INTERRUPT ENABLE
016526  012777  016626  163342                 MOV     #11$,@DZTIV         ;SET UP THE TRANSMITTER INTERRUPT VECTOR
016534  012777  016634  163330                 MOV     #12$,@DZRIV         ;SET UP THE RECEIVER INTERRUPT VECTOR
016542  013777  026454  163324                 MOV     DZPRT,@DZRIS        ;SET THE INTERRUPT VECTOR STATUS
016550  013777  026454  163322                 MOV     DZPRT,@DZTIS        ;SET TRANSMITTER INTERRUPT PRIORITY
016556  052777  000140  163256                 BIS     #RIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
016564  113777  001422  163274                 MOVB    TDO,@DZTDR          ;PUT ANY RANDOM CHARACTER IN TRANSMITTER BUFFER
016572  005005                                 CLR     R5
016574  032777  000200  163240         9$:     BIT     #RDONE,@DZCSR
016602  001404                                 BEQ     10$
016604  000240                                 NOP
016606  000240                                 NOP
016610  104011                                 ERROR   11                  ;*RECEIVER FAILED TO INTERRUPT
016612  000413                                 BR      13$
016614  104414                         10$:    DELAY
016616  005205                                 INC     R5
```

```
016620  001365                            BNE     9$
016622  104004                            ERROR   4               ;*NO RX DONE! (NOT SET)
016624  000406                            BR      13$             ;CONTINUE TEST
016626  104010                    11$:    ERROR   10              ;*TRANSMITTER SHOULD NOT INTERRUPT
016630  022626                            CMP     (SP)+,(SP)+     ;POP FOR FAKE RTI
016632  000403                            BR      13$             ;CONT TEST
016634  022626                    12$:    POP2SP                  ;REMOVE THE INTERRUPT FROM THE STACK
016636  005077  163200                    CLR     @DZCSR          ;DON'T ALLOW ANY MORE INTERRUPTS
016642                            13$:
016642  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
                                  ;;************************************************************
016644  000004                    TST26:  SCOPE
016646  012737  000026  001122            MOV     #26,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
016654  012737  017276  001360            MOV     #TST27,NEXT     ;POINT TO THE START OF THE NEXT TEST
016662  104417                            DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
016664  013701  001366                    MOV     PAR,R1          ;PICK UP PARAMETERS
016670  012702  000001                    MOV     #1,R2           ;PICK UP INIT POINTER
016674  030237  001364            1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
016700  001402                            BEQ     2$              ;NO
016702  010177  163144                    MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
016706  005201                    2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
016710  106302                            ASLB    R2              ;GOT 'EM ALL ?
016712  103370                            BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
016714  005037  001372                    CLR     SAVLIN          ;CLEAR LINE # INDICATOR
016720  012777  017150  163144            MOV     #8$,@DZRIV      ;SETUP INTERRUPT STUFF
016726  013777  026454  163140            MOV     DZPRT,@DZRIS    ;
016734  012777  017240  163134            MOV     #12$,@DZTIV     ;
016742  013777  026454  163130            MOV     DZPRT,@DZTIS    ;
016750  052777  000040  163064            BIS     #MSENAB,@DZCSR
016756  012702  000001                    MOV     #1,R2           ;LINE POINTER
016762  030237  001364            3$:     BIT     R2,LINE         ;VALID LINE ?
016766  001004                            BNE     4$
016770  005237  001372                    INC     SAVLIN
016774  106302                            ASLB    R2
016776  000771                            BR      3$
017000  106427  000340            4$:     MTPS    #PR7
017004  000240                            NOP
017006  000240                            NOP
017010  110277  163042                    MOVB    R2,@DZTCR       ;SET TCR BIT
017014  005777  163026                    TST     @DZRBUF         ;VALID DATA?
017020  100001                            BPL     .+4             ;IT BETTER NOT BE SET
017022  104017                            ERROR   17              ;DATA VALID SHOULD NOT BE SET
017024  105777  163012            5$:     TSTB    @DZCSR          ;RECEIVER DONE ?
017030  100001                            BPL     .+4
017032  104020                            ERROR   20              ;RECEIVER DONE BIT SHOULD NOT BE SET
017034  005005                            CLR     R5
```

```
017036  005004                                   CLR     R4
017040  005777  162776                   99$:    TST     @DZCSR               ;WAIT FOR TRDY
017044  100404                                   BMI     100$                 ;BR IF READY
017046  104414                                   DELAY                        ;STALL TIME
017050  005204                                   INC     R4                   ;
017052  001372                                   BNE     99$
017054  104003                                   ERROR   3                    ;TRDY FAILED TO SET
017056  105077  163004                   100$:   CLRB    @DZTDR
017062  005004                                   CLR     R4
017064  032777  000200  162750           6$:     BIT     #RDONE,@DZCSR
017072  001004                                   BNE     7$
017074  104414                                   DELAY
017076  005204                                   INC     R4
017100  001371                                   BNE     6$
017102  104004                                   ERROR   4                    ;*RDONE FAILED TO SET!
017104  005777  162732                   7$:     TST     @DZCSR               ;TRANS DONE BIT = 1 ?
017110  100401                                   BMI     .+4                  ;YES
017112  104003                                   ERROR   3                    ;*NO   TRANS DONE FAILED TO SET
                                                 ;NOW THAT BOTH TRANSMITTER AND RECEIVER DONE BIT =1
                                                 ;SET INTERRUPT ENABLES AND WATCH THE FUR FLY
017114  052777  040000  162720                   BIS     #TIE,@DZCSR
017122  052777  000100  162712                   BIS     #RIE,@DZCSR
017130  106427  000000                           MTPS    #0
017134  000240                                   NOP
017136  000240                                   NOP
017140  104007                                   ERROR   7                    ;*TRANSMITTER FAILED TO INTERRUPT
017142  104011                                   ERROR   11                   ;*RECEIVER FAILED TO INTERRUPT
                                                                              ;CHECK BR LEVEL
017144  000137  017244                           JMP     13$      ;GET OUT

                                                 ;RECEIVER INTERRUPT ROUTINE
017150  017704  162672                   8$:     MOV     @DZRBUF,R4                     ;ACTUAL
017154  010403                                   MOV     R4,R3
017156  000303                                   SWAB    R3
017160  042703  177770                           BIC     #^C<7>,R3            ;STRIP JUNK
017164  105737  001371                           TSTB    MODE+1               ;IS THIS TEST IN STAGGERED MODE?
017170  001406                                   BEQ     11$                  ;IF NOT, SKIP STAGGERED SETUP

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

017172  006203                                   ASR     R3                   ;GET THE LAST BIT INTO THE CARRY BIT
017174  103402                                   BCS     9$                   ;IF IT IS SET, GO CLEAR IT
017176  000261                                   SEC                          ;IF IT IS CLEAR SET IT HERE
017200  000401                                   BR      10$                  ;SKIP THE CLEARING
017202  000241                           9$:     CLC                          ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
017204  006103                           10$:    ROL     R3                   ;GET THE NEW BIT BACK INTO R3
017206  020337  001372                   11$:    CMP     R3,SAVLIN            ;IS THIS A VALID LINE
017212  001401                                   BEQ     .+4                  ;YES
017214  104015                                   ERROR   15                   ;*INVALID LINE
017216  042704  177400                           BIC     #^C<377>,R4          ;STRIP JUNK
017222  120504                                   CMPB    R5,R4                ;DATA COMPARE ?
017224  001401                                   BEQ     .+4                  ;YES
017226  104005                                   ERROR   5                    ;*DATA DOES NOT COMPARE
017230  040277  162622                           BIC     R2,@DZTCR            ;CLEAR TCR BIT
017234  022626                                   POP2SP                       ;REMOVE HE INTERRUPT VECTOR FROM THE STACK
017236  000402                                   BR      13$                  ;GO GET OUT OF INTERRUPT MODE
```

```
                                                ;TRANSMITTER INTERRUPT SVC ROUTINE
 017240  104011                         12$:    ERROR   11              ;THE RECEIVER INTERRUPT FAILED
                                                                        ;TO OVERRIDE THE TRANSMITTER
 017242  022626                                 POP2SP                  ;REMOVE THE INTERRUPT VECTOR FROM THE STACK
 017244  042777  040100  162570         13$:    BIC     #TIE!RIE,@DZCSR ;CLEAR INTERRUPT ENABLES
 017252  013777  002074  162612                 MOV     DZRIS,@DZRIV    ;RESTORE TRAPCATCHER
 017260  005077  162610                         CLR     @DZRIS
 017264  013777  002100  162604                 MOV     DZTIS,@DZTIV
 017272  005077  162602                         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.
                                                ;*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
                                        ;;*****************************************************************
 017276  000004                         TST27:  SCOPE
 017300  012737  000027  001122                 MOV     #27,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
 017306  012737  020024  001360                 MOV     #TST30,NEXT     ;POINT TO THE START OF THE NEXT TEST
 017314  012737  017730  001362                 MOV     #18$,LOCK       ;SET FOR LOOP
 017322  104417                                 DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
 017324  013701  001366                         MOV     PAR,R1          ;PICK UP PARAMETERS
 017330  012702  000001                         MOV     #1,R2           ;PICK UP INIT POINTER
 017334  030237  001364                 1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
 017340  001402                                 BEQ     2$              ;NO
 017342  010177  162504                         MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
 017346  005201                         2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
 017350  106302                                 ASLB    R2              ;GOT 'EM ALL ?
 017352  103370                                 BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
 017354  005037  001372                         CLR     SAVLIN          ;CLEAR LINE # INDICATOR
 017360  012700  001422                         MOV     #TDO,RO         ;POINT TO THE DATA AREA
 017364  005020                                 CLR     (RO)+           ;CLEAR A DATA WORD
 017366  022700  001462                         CMP     #STOP,RO        ;FINISHED ?
 017372  001374                                 BNE     .-6             ;NO
 017374  005000                                 CLR     RO              ;CLEAR OFFSET
 017376  012702  000001                         MOV     #1,R2           ;LINE POINTER
 017402  052777  010040  162432                 BIS     #MSENAB!SILOEN,@DZCSR   ;START SCANNER & SET SILO ENABLE
 017410  030237  001364                 3$:     BIT     R2,LINE         ;VALID LINE?
 017414  001002                                 BNE     .+6             ;YES
 017416  000137  017712                         JMP     22$             ;TRY NEXT LINE
 017422  013700  001372                         MOV     SAVLIN,RO       ;MAKE OFFSET
 017426  006300                                 ASL     RO              ;MAKE POWER OF TWO
 017430  010277  162422                         MOV     R2,@DZTCR       ;SET TCR BIT
 017434  105777  162402                 4$:     TSTB    @DZCSR          ;REC DONE = 1 ?
 017440  100001                                 BPL     .+4
 017442  104020                                 ERROR   20              ;REC DONE SHOULD NOT = 1
 017444  005003                                 CLR     R3              ;SET CHARACTER COUNT
 017446  005004                         5$:     CLR     R4
 017450  032777  100000  162364         6$:     BIT     #TRDY,@DZCSR
 017456  001004                                 BNE     7$
```

```
017460  104414                          DELAY
017462  105204                          INCB    R4
017464  001371                          BNE     6$
017466  104003                          ERROR   3                ;*TRDY FAILED TO SET
017470  116077  001422  162370  7$:     MOVB    TDO(R0),@DZTDR   ;LOAD A CHARACTER
017476  005260  001422                  INC     TDO(R0)          ;SET UP NEXT CHARACTER
017502  020327  000017                  CMP     R3,#15.          ;16 CHARACTERS ?
017506  103006                          BHIS    8$
017510  032777  020000  162324          BIT     #SILOAL,@DZCSR   ;SILO ALARM = 0 ?
017516  001401                          BEQ     .+4              ;YES
017520  104013                          ERROR   13               ;*SILO ALARM SHOULD NOT = 1
                                                                 ;UNTIL 16. DATA CHARACTERS
017522  000411                          BR      10$
017524  005004                  8$:     CLR     R4
017526  032777  020000  162306  9$:     BIT     #SILOAL,@DZCSR
017534  001004                          BNE     10$
017536  104414                          DELAY
017540  005204                          INC     R4
017542  001371                          BNE     9$
017544  104014                          ERROR   14               ;*SILO ALARM FAILED TO SET!
                                                                 ;SILO ALARM SHOULD =1 AFTER 16.
                                                                 ;DATA CHARACTERS
017546  005203                  10$:    INC     R3               ;INC CHAR COUNT
017550  022703  000102                  CMP     #66.,R3          ;FINISHED SENDING CHARACTERS ?
017554  001334                          BNE     5$               ;NO
017556  005004                          CLR     R4
017560  104414                          DELAY
017562  105204                          INCB    R4
017564  001375                          BNE     .-4
                                        ;NOW LETS READ THE SILO
017566  013705  001372                  MOV     SAVLIN,R5        ;MAKE EXPECTED LINE #
017572  105737  001371                  TSTB    MODE+1           ;IS THIS TEST IN STAGGERED MODE?
017576  001406                          BEQ     13$              ;IF NOT, SKIP STAGGERED SETUP

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

017600  006205                          ASR     R5               ;GET THE LAST BIT INTO THE CARRY BIT
017602  103402                          BCS     11$              ;IF IT IS SET, GO CLEAR IT
017604  000261                          SEC                      ;IF IT IS CLEAR SET IT HERE
017606  000401                          BR      12$              ;SKIP THE CLEARING
017610  000241                  11$:    CLC                      ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
017612  006105                  12$:    ROL     R5               ;GET THE NEW BIT BACK INTO R5
017614  000305                  13$:    SWAB    R5               ;PUT IN UPPER BYTE
017616  052705  100000                  BIS     #DVALID,R5       ;ADD DATA VALID
017622  017704  162220          14$:    MOV     @DZRBUF,R4       ;ACTUAL
017626  020405                          CMP     R4,R5            ;ACTUAL VS. EXPECTED
017630  001401                          BEQ     15$              ;YES
017632  104006                          ERROR   6                ;*DATA/CONTENTS DID NOT COMPARE
017634  032777  020000  162200  15$:    BIT     #SILOAL,@DZCSR   ;SILO ALARM= 0 ?
017642  001401                          BEQ     16$              ;YES
017644  104016                          ERROR   16               ;READING DZRBUF DID NOT CLEAR SILO ALARM
017646  005205                  16$:    INC     R5               ;UP CHARACTER
017650  120527  000077                  CMPB    R5,#63.          ;LAST SILO CHAR ?....64TH CHAR
017654  101762                          BLOS    14$
017656  005205                          INC     R5               ;ADD 1 MORE FOR THE CLOBBERED CHAR
017660  052705  040000                  BIS     #OVRRUN,R5       ;ADD OVERRUN TO EXPECTED
```

```
017664  120527  000101                  CMPB    R5,#65.         ;LAST CHARACTER ?
017670  001754                          BEQ     14$
017672  017704  162150                  MOV     @DZRBUF,R4      ;FOR GOOD MEASURE
017676  005704                          TST     R4              ;DATA VALID SHOULD = 0
017700  100001                          BPL     17$             ;YES
017702  104017                          ERROR   17              ;DATA VALID SHOULD = 0
017704  040277  162146          17$:    BIC     R2,@DZTCR       ;CLR TCR BIT
017710  104401                          SCOP1                   ;LOOP?
017712  005237  001372          22$:    INC     SAVLIN          ;INC EXPECTED LINE
017716  106302                          ASLB    R2              ;NEXT LINE
017720  103402                          BCS     .+6             ;NO
017722  000137  017410                  JMP     3$              ;YES
017726  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.

017730  052777  010040  162104  18$:    BIS     #MSENAB!SILOEN,@DZCSR   ;SETUP DEVICE
017736  012777  020014  162132          MOV     #20$,@DZTIV     ;SETUP TRANSMITTER VECTOR
017744  012737  000024  001216          MOV     #20.,$TMPO      ;TEMPORARY COUNT OF CHARACTER BURST
017752  050277  162100                  BIS     R2,@DZTCR       ;ENABLE LINE
017756  052777  040000  162056          BIS     #TIE,@DZCSR     ;ENABLE INTERRUPTS
017764  106427  000000                  MTPS    #0              ;LOWER PRIORITY
017770  000001                  19$:    WAIT                    ;ALLOW INTERRUPTS
017772  005337  001216                  DEC     $TMPO           ;REDUCE COUNT. ALL CHARACTERS SENT?
017776  001374                          BNE     19$             ;IF NO, WAIT FOR MORE
020000  042777  050040  162034          BIC     #SILOEN!MSENAB!TIE,@DZCSR ;RESET SILO COUNTER, CLEAR STROBE
020006  104401                          SCOP1                   ;LOOP AGAIN?
020010  000137  017704                  JMP     17$             ;IF NOT, RETURN TO WHERE YOU LEFT OFF
020014  112777  000252  162044  20$:    MOVB    #252,@DZTDR     ;SEND A CHARACTER
020022  000002                          RTI                     ;ALLOW MORE CHARACTERS TO COME
                                        ;*********************** TEST 30 ******************************
                                        ;*THIS TEST THAT "SILO ENABLE" WILL INHIBIT
                                        ;*RECEIVER INTERRUPTS AND THAT ON THE
                                        ;*16TH CHAR THAT "SILO ALARM" WILL CAUSE AN
                                        ;*INTERRUPT WITH "RIE" SET.
                                        ;*THIS WILL DO ALL SELECTED LINES ONE AT A TIME.
                                ;;*  TEST 30
                                ;;***************************************************************
020024  000004                  TST30:  SCOPE
020026  012737  000030  001122          MOV     #30,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
020034  012737  020406  001360          MOV     #TST31,NEXT     ;POINT TO THE START OF THE NEXT TEST
020042  012737  020130  001362          MOV     #3$,LOCK        ;SET FOR LOOP
020050  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
020052  013701  001366                  MOV     PAR,R1          ;PICK UP PARAMETERS
020056  012702  000001                  MOV     #1,R2           ;PICK UP INIT POINTER
020062  030237  001364          1$:     BIT     R2,LINE         ;SHOULD THIS LINE BE SET UP ?
020066  001402                          BEQ     2$              ;NO
020070  010177  161756                  MOV     R1,@DZLPR       ;SET UP LINE PARAMETERS
020074  005201                  2$:     INC     R1              ;POSITION POINTER TO THE NEXT LINE
020076  106302                          ASLB    R2              ;GOT 'EM ALL ?
020100  103370                          BCC     1$              ;IF NO, GO SET UP THE NEXT LINE
020102  005037  001372                  CLR     SAVLIN          ;CLEAR LINE # INDICATOR
020106  012700  001422                  MOV     #TD0,R0         ;POINT TO THE DATA AREA
020112  005020                          CLR     (R0)+           ;CLEAR A DATA WORD
```

```
020114  022700  001462                       CMP     #STOP,R0             ;FINISHED ?
020120  001374                               BNE     .-6                  ;NO
020122  005000                               CLR     R0                   ;CLEAR OFFSET
020124  012702  000001                       MOV     #1,R2                ;LINE POINTER
020130  012777  020350  161734  3$:          MOV     #11$,@DZRIV          ;SET FOR UNEXPECTED INTER.
020136  012777  000340  161730               MOV     #PR7,@DZRIS          ;SET PRIO.
020144  052777  010140  161670               BIS     #MSENAB!SILOEN!RIE,@DZCSR
                                                                          ;START SCANNER & SET SILO ENABLE
020152  030237  001364                       BIT     R2,LINE              ;VALID LINE?
020156  001002                               BNE     .+6                  ;YES
020160  000137  020366                       JMP     22$                  ;TRY NEXT LINE
020164  005777  161656                       TST     @DZRBUF              ;EMPTY THE SILO
020170  100775                               BMI     .-4                  ;BR IF DATA VALID IS SET!
020172  106427  000000                       MTPS    #0                   ;SET PROCESSOR PRIORITY TO 0
020176  013700  001372                       MOV     SAVLIN,R0            ;MAKE OFFSET
020202  006300                               ASL     R0                   ;MAKE POWER OF TWO
020204  010277  161646                       MOV     R2,@DZTCR            ;SET TCR BIT
020210  005004                       5$:     CLR     R4
020212  032777  100000  161622  6$:          BIT     #TRDY,@DZCSR
020220  001004                               BNE     7$
020222  104414                               DELAY
020224  005204                               INC     R4
020226  001371                               BNE     6$
020230  104003                               ERROR   3                    ;*TRDY FAILED TO SET
020232  116077  001422  161626  7$:          MOVB    TDO(R0),@DZTDR       ;LOAD A CHARACTER
020240  005260  001422                       INC     TDO(R0)              ;SET UP NEXT CHARACTER
020244  022760  000017  001422               CMP     #15.,TDO(R0)         ;15 CHARS YET?
020252  001406                               BEQ     8$
020254  032777  020000  161560               BIT     #SILOAL,@DZCSR       ;SILO ALARM = 0 ?
020262  001401                               BEQ     .+4                  ;YES
020264  104013                               ERROR   13                   ;*SILO ALARM SHOULD NOT = 1
                                                                          ;UNTIL 16. DATA CHARACTERS
020266  000750                               BR      5$
020270  012777  020356  161574  8$:          MOV     #12$,@DZRIV          ;SET NEW VECTOR
020276  032777  100000  161536               BIT     #TRDY,@DZCSR         ;READY FOR 16TH CHAR
020304  001774                               BEQ     .-6
020306  016077  001422  161552               MOV     TDO(R0),@DZTDR       ;LOAD THE 16TH CHAR.
020314  005004                               CLR     R4
020316  032777  020000  161516  9$:          BIT     #SILOAL,@DZCSR
020324  001005                               BNE     10$
020326  104414                               DELAY
020330  005204                               INC     R4
020332  001371                               BNE     9$
020334  104014                               ERROR   14                   ;*SILO ALARM FAILED TO SET!
020336  000410                               BR      17$                  ;SILO ALARM SHOULD =1 AFTER 16.
                                                                          ;DATA CHARACTERS
020340  000240                       10$:    NOP                          ;STALL
020342  000240                               NOP
020344  104000                               ERROR                        ;SILO ALARM NOT INTERRUPTING.
020346  000404                               BR      17$                  ;CONTINUE TEST.
020350  022626                       11$:    CMP     (SP)+,(SP)+          ;FAKE RTI
020352  104012                               ERROR   12                   ;RX SHOULD NOT INTERRUPT
020354  000401                               BR      17$                  ;CONTINUE
020356  022626                       12$:    CMP     (SP)+,(SP)+          ;GOOD INTERRUPT TO HERE.
020360  040277  161472               17$:    BIC     R2,@DZTCR            ;CLR TCR BIT
020364  104401                               SCOP1                        ;LOOP?
```

```
  3339  020366  005237  001372          22$:    INC     SAVLIN          ;INC EXPECTED LINE
  3340  020372  106302                          ASLB    R2              ;NEXT LINE
  3341  020374  103402                          BCS     .+6             ;NO
  3342  020376  000137  020130                  JMP     3$              ;YES
  3343  020402  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
                                                ;;************************************************************
        020406  000004                          TST31:  SCOPE
        020410  012737  000031  001122                  MOV     #31,$TSTNM              ;LOAD THE NUMBER OF THIS TEST
        020416  012737  021214  001360                  MOV     #TST32,NEXT             ;POINT TO THE START OF THE NEXT TEST
        020424  104417                                  DCLASM                          ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
        020426  013737  001364  021212                  MOV     LINE,RXTCR              ;SET IMAGE OF TCR BITS
        020434  013701  001366                  RSTART: MOV     PAR,R1                  ;PICK UP PARAMETER
        020440  012700  000001                          MOV     #1,R0                   ;;PICK UP INIT POINTER
        020444  030037  001364                  INIT:   BIT     R0,LINE                 ;SHOULD THIS LINE BE SET UP
        020450  001402                                  BEQ     1$                      ;NO
        020452  010177  161374                          MOV     R1,@DZLPR               ;SET UP LINE PARAM REGISTER
        020456  005201                          1$:     INC     R1
        020460  106300                                  ASLB    R0                      ;GOT 'EM ALL ?
        020462  103370                                  BCC     INIT                    ;NO
        020464  012700  001422                          MOV     #TD0,R0                 ;CLEAR TRANS DATA POINTER & REC POINTERS
        020470  005020                          INIT1:  CLR     (R0)+
        020472  022700  001462                          CMP     #STOP,R0                        ;FINISHED ?
        020476  001374                                  BNE     INIT1                   ;NO, CONTINUE CLEARING
        020500  012777  020734  161364                  MOV     #RXSVC,@DZRIV           ;SET UP REC INTR VECTOR
        020506  012777  000340  161360                  MOV     #PR7,@DZRIS             ;STATUS
        020514  012777  020636  161354                  MOV     #TXSVC,@DZTIV           ;SET UP TRANS INTR VECTOR
        020522  012777  000340  161350                  MOV     #PR7,@DZTIS             ;STATUS
        020530  052777  000100  161304                  BIS     #RIE,@DZCSR                     ;SET REC INTR ENABLE
        020536  052777  040000  161276                  BIS     #TIE,@DZCSR                     ;SET TRANS INTR ENABLE
        020544  052777  000040  161270                  BIS     #MSENAB,@DZCSR          ;SET MASTER SCAN ENABLE
        020552  113777  001364  161276                  MOVB    LINE,@DZTCR             ;SET TCR BITS...UP UP AND AWAY !
        020560  106437  026456                          MTPS    @#LESS1                         ;ALLOW INTERRUPTS


        020564  005037  020634                  SNAP:   CLR     66$
        020570  013727  006722                  67$:    MOV     DLYCNT,(PC)+            ;SET FOR DELAY
        020574  000000                          68$:    0
        020576  005337  020574                          DEC     68$
        020602  001375                                  BNE     .-4
        020604  105737  021212                          TSTB    RXTCR                   ;WAIT FOR ALL RECIEVERS TO FINISH
        020610  001002                                  BNE     3$                      ;
        020612  000137  021112                          JMP     OUT
        020616  005237  020634                  3$:     INC     66$
        020622  001362                                  BNE     67$
        020624  104007                                  ERROR   7                       ;*TRANSMITTER FAILED TO INTERRUPT
        020626  104011                                  ERROR   11                      ;*RECEIVER FAILED TO INTERRUPT
        020630  000137  021164                          JMP     FINI
        020634  000000                          66$:    0

                                                ;TRANS INTR SVC ROUTINE
        020636  005777  161200                  TXSVC:  TST     @DZCSR                  ;TRANS INTR ?
        020642  100401                                  BMI     .+4
        020644  104003                                  ERROR   3                       ;*TRANSMITTER FAILED
        020646  117703  161172                          MOVB    @HDZCSR,R3              ;SAVE IT
                                                ;NOW TEST FOR LINE # ETC
```

```
020652  042703  177770                  BIC     #↑C<7>,R3               ;STRIP JUNK
020656  010304                          MOV     R3,R4                   ;SAVE
020660  010337  001372                  MOV     R3,SAVLIN               ;ADJUST LOCATION FOR ERROR PRINTOUT
020664  012702  000001                  MOV     #1,R2                   ;SET UP POSITION POINTER
020670  105303                  3$:     DECB    R3                      ;IS IT THIS LINE ?
020672  100402                          BMI     4$                      ;YES
020674  006302                          ASL     R2                      ;UP THE LINE #
020676  000774                          BR      3$                      ;GO 'ROUND AGAIN
020700  030237  001364          4$:     BIT     R2,LINE                 ;VALID LINE?
020704  001001                          BNE     .+4                     ;YES
020706  104010                          ERROR   10                      ;NO,INVALID LINE!!!!
020710  006304                          ASL     R4                      ;MAKE POWER OF 2
020712  116477  001422  161146          MOVB    TDO(R4),@DZTDR          ;LOAD CHARACTER
020720  105264  001422                  INCB    TDO(R4)                 ;SET UP NEXT CHARACTER
020724  001002                          BNE     5$                      ;LAST CHARACTER ?
020726  040277  161124                  BIC     R2,@DZTCR               ;YES ,CLEAR TCR BIT
020732  000002                  5$:     RTI



                                ;REC INTR SVC ROUTINE
020734  105777  161102          RXSVC:  TSTB    @DZCSR                  ;REC DONE ?
020740  100401                          BMI     .+4                     ;YES
020742  104004                          ERROR   4                       ;FALSE INTERRUPT
020744  017704  161076                  MOV     @DZRBUF,R4              ;SAVE IT
020750  010403                          MOV     R4,R3
020752  000303                          SWAB    R3
020754  042703  177770                  BIC     #↑C<7>,R3               ;STRIP JUNK
020760  010337  001372                  MOV     R3,SAVLIN               ;SAVE LINE NUMBER
020764  032777  020000  161050          BIT     #SILOAL,@DZCSR          ;SILO ALARM?
020772  001401                          BEQ     .+4                     ;NO
020774  104000                          ERROR                           ;SILO ALARM SHOULD NOT =1
020776  005704                          TST     R4                      ;DATA VALID SET?
021000  100401                          BMI     .+4                     ;YES
021002  104023                          ERROR   23                      ;YOU LOSE  ...DATA VALID WAS'NT SET
021004  032704  070000                  BIT     #OVRRUN!FRMERR!PARER,R4
021010  001401                          BEQ     .+4
021012  104000                          ERROR                           ;RECEIVER ERROR FLAG/S WERE SET
021014  012702  000001                  MOV     #1,R2                   ;SET UP POSITION POINTER
021020  105303                  5$:     DECB    R3
021022  100402                          BMI     6$
021024  006302                          ASL     R2                      ;RE POSITION POINTER
021026  000774                          BR      5$                      ;GO 'ROUND AGAIN
021030  030237  001364          6$:     BIT     R2,LINE                 ;LINE VALID ?
021034  001001                          BNE     .+4                     ;YES
021036  104011                          ERROR   11                      ;INVALID LINE #
021040  013703  001372                  MOV     SAVLIN,R3               ;GET THE LINE NUMBER AGAIN
021044  006303                          ASL     R3                      ;USE R3 AS A POINTER IN THE DATA TABLE
021046  126304  001442                  CMPB    TRO(R3),R4              ;DOES THE DATA CHARACTER COMPARE ?
021052  001405                          BEQ     2$                      ;YES
021054  016305  001442                  MOV     TRO(R3),R5              ;SAVE EXPECTED
021060  042704  177400                  BIC     #↑C<377>,R4             ;CLEAR JUNK
                                                                        ;R2 = LINE # BY BIT POSITION
                                                                        ;R4 = ACTUAL DATA
                                                                        ;R5 = EXPECTED DATA
021064  104005                          ERROR   5                       ;*NO, DATA DOES NOT COMPARE
021066  005263  001442          2$:     INC     TRO(R3)                 ;SET UP FOR NEXT CHARACTER
```

```
021072  105763  001442                  TSTB    TRO(R3) ;ALL CHARS DONE?
021076  001002                          BNE     .+6
021100  040237  021212                  BIC     R2,RXTCR          ;ZERO LINE DONE INDICATOR.
021104  012716  020564                  MOV     #SNAP,(SP)        ;RESET THE BACKGROUND TIMING LOOP
021110  000002                          RTI



                                        ;FINISH UP ROUTINE
021112  106427  000340          OUT:    MTPS    #PR7                        ;STOP ALL INTERRUPTS
021116  104413                          DEVICE.CLR                ;CLEAR ALL INTERRUPTS AWAY
021120  005003                          CLR     R3
021122  005037  001372                  CLR     SAVLIN
021126  012702  000001                  MOV     #1,R2
021132  030237  001364          1$:     BIT     R2,LINE           ;VALID LINE ?
021136  001405                          BEQ     2$                ;NO
021140  022763  000400  001442          CMP     #400,TRO(R3)      ;RECEIVED A BINARY COUNT PATTERN ?
021146  001401                          BEQ     .+4               ;YES
021150  104027                          ERROR   27                ;THE LINE FAILED TO RECEIVE A FULL
                                                                  ;BINARY COUNT PATTERN
021152  005237  001372          2$:     INC     SAVLIN            ;SET UP FOR NEXT LINE
021156  005723                          TST     (R3)+             ;ADD 2
021160  106302                          ASLB    R2                ;SET UP NEXT LINE POINTER
021162  103363                          BCC     1$                ;FINISHED ?
021164                          FINI:
021164  013777  002074  160700          MOV     DZRIS,@DZRIV      ;RESTORE TRAPCATCHER
021172  005077  160676                  CLR     @DZRIS
021176  013777  002100  160672          MOV     DZTIS,@DZTIV
021204  005077  160670                  CLR     @DZTIS
021210  104400                          ADVANCE                   ;GO TO THE NEXT TEST
021212  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
                                ;;*************************************************************
021214  000004                  TST32:  SCOPE
021216  012737  000032  001122          MOV     #32,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
021224  012737  000002  001226          MOV     #2,$TIMES
021232  012737  021724  001360          MOV     #TST33,NEXT      ;POINT TO THE START OF THE NEXT TEST
021240  012737  021364  001362          MOV     #3$,LOCK         ;SET FOR LOOP
021246  005037  023362                  CLR     OFFSET           ;RESET THIS VARIABLE
```

```
021252  005037  001372                          CLR     SAVLIN                  ;RESET LINE NUMBER INDICATOR
021256  005037  001374                          CLR     XMTLIN                  ;USE THIS WORD TO TELL WHAT LINE TRANSMITTED
021262  012737  000001  001216                  MOV     #1,$TMPO                ;USE $TMPO AS A BIT POINTER
021270  012737  010070  021722                  MOV     #RCVON!S5O!EIGHT!TWOSTOP,7$ ;BUILD TEMPORARY PARAMETERS
021276  033737  001216  001364  1$:             BIT     $TMPO,LINE              ;IS THIS LINE ACTIVE?
021304  001027                                  BNE     3$                      ;IF SO, GO GET STARTED
021306  012737  010070  021722  2$:             MOV     #RCVON!S5O!EIGHT!TWOSTOP,7$ ;LOAD PARAMETERS TEMPORARILY
021314  012700  001422                          MOV     #TD0,R0                 ;POINT TO THE DATA AREA
021320  005020                                  CLR     (R0)+                   ;CLEAR A DATA WORD
021322  022700  001462                          CMP     #STOP,R0                ;FINISHED ?
021326  001374                                  BNE     .-6                     ;NO
021330  005237  001374                          INC     XMTLIN                  ;POINT TO THE NEXT LINE TO TRANSMIT
021334  042737  000007  021722                  BIC     #7,7$                   ;MAKE SURE TEMPORARY PARAMETERS POINT TO 0
021342  053737  001374  021722                  BIS     XMTLIN,7$               ;ADD DESIRED LINE NUMBER
021350  005037  023362                          CLR     OFFSET
021354  106337  001216                          ASLB    $TMPO                   ;POINT TO THE NEXT LINE
021360  103346                                  BCC     1$                      ;PROCESS THE NEXT LINE
021362  104400                                  ADVANCE                         ;TEST TO SEE IF THIS TEST GETS REPEATED
021364                                  3$:
021364  104417                                  DCLASM                          ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
021366  042737  010000  021722                  BIC     #RCVON,7$               ;ZERO PARAMTERS FOR TX LINE
021374  013777  021722  160450                  MOV     7$,@DZLPR               ;LOAD PARAMTERS FOR TX
021402  005737  001370                          TST     MODE                    ;STAGGERED?
021406  100011                                  BPL     100$                    ;BR IF NO
021410  000241                                  CLC                             ;SET UP LINE
021412  006037  021722                          ROR     7$                      ;
021416  103002                                  BCC     98$                     ;BR IF LINE WAS EVEN
021420  000241                                  CLC                             ;PREPARE TO MKE LINE EVEN
021422  000401                                  BR      99$                     ;CONTINUE
021424  000261                          98$:    SEC                             ;PREPARE TO MAKE LINE ODD
021426  006137  021722                  99$:    ROL     7$                      ;SET ALTERED LINE
021432  052737  010000  021722          100$:   BIS     #RCVON,7$               ;SET RX ON
021440  013777  021722  160404                  MOV     7$,@DZLPR               ;LOAD RX PARAMETERS
021446  013737  021722  001372                  MOV     7$,SAVLIN               ;ADJUST LOCATION FOR ERROR PRINTOUT
021454  042737  177770  001372                  BIC     #↑C<7>,SAVLIN           ;STRIP JUNK
021462  042737  000007  021722                  BIC     #7,7$                   ;CLEAR OLD LINE #
021470  053737  001374  021722                  BIS     XMTLIN,7$               ;SET LINE UP AGAIN
021476  013737  021722  001400                  MOV     7$,REGIST               ;SAVE PARAMETERS FOR PRINTOUT
021504  012700  001422                          MOV     #TD0,R0                 ;POINT TO THE DATA AREA
021510  005020                                  CLR     (R0)+                   ;CLEAR A DATA WORD
021512  022700  001462                          CMP     #STOP,R0                ;FINISHED ?
021516  001374                                  BNE     .-6                     ;NO
021520  005002                                  CLR     R2                      ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
021522  005003                                  CLR     R3                      ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
021524  005037  001220                          CLR     $TMP1                   ;INITIALIZE THE TIMER
021530  005037  001224                          CLR     $TMP3                   ;INITIALIZE THESE BITS ALSO
021534  012737  000020  001376                  MOV     #20,XMTCNT              ;SET HOW MANY CHARACTERS TO TRANSMIT
021542  012777  023004  160326                  MOV     #XMTSRV,@DZTIV
021550  012777  023150  160314                  MOV     #RXISR1,@DZRIV
021556  013777  026454  160310                  MOV     DZPRT,@DZRIS
021564  013777  026454  160306                  MOV     DZPRT,@DZTIS
021572  113777  001216  160256                  MOVB    $TMPO,@DZTCR            ;START THE VALID LINE
021600  052777  040140  160234                  BIS     #TIE!RIE!MSENAB,@DZCSR
021606  106427  000000                          MTPS    #0                      ;LOWER THE PRIORITY TO ALLOW INTERRUPTS
021612  032777  000100  160222  4$:             BIT     #RIE,@DZCSR             ;IS ROUTINE DONE?
021620  001407                                  BEQ     5$                      ;WHEN ALL IS DONE RX IE IS CLEARED IN ISR.
```

```
021622  005237  001220                   INC     $TMP1              ;COUNT TIME
021626  001371                           BNE     4$                 ;CONTINUE TEST
021630  105237  001224                   INCB    $TMP3              ;DOUBLE COUNT
021634  001366                           BNE     4$                 ;CONTINUE TEST
021636  104011                           ERROR   11                 ;INTERRUPTS NOT FINISHED
021640  004737  007360          5$:      JSR     PC,SERV.G          ;<↑G>?
021644  104401                           SCOP1                      ;LOOP?
021646  062737  000002  023362           ADD     #2,OFFSET
021654  013700  021722                   MOV     7$,R0
021660  042700  170377                   BIC     #↑C<17*400>,R0
021664  022700  007400                   CMP     #<17*400>,R0
021670  001010                           BNE     6$
021672  032737  000030  021722           BIT     #BIT4+BIT3,7$
021700  001602                           BEQ     2$
021702  162737  000010  021722           SUB     #BIT3,7$
021710  000625                           BR      3$
021712  062737  000400  021722  6$:      ADD     #400,7$
021720  000621                           BR      3$
021722  000000                  7$:      0
                                         ;************************ TEST 33 ****************************
                                         ;* THIS TEST VERIFIES THAT EVEN PARITY WORKS
                                         ;* FOR ALL ODD LINES SELECTED AND THAT ODD PARITY WORKS FOR ALL
                                         ;* EVEN LINES SELECTED.
                                         ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
                                         ;*THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
                                         ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
                                         ;*YOU ARE IN "STAGGERED" MODE.
                                         ;*40(8) CHARS ARE USED FOR THIS TEST.
                                         ;*ALL SELECTED LINES WILL BE ENABLED
                                         ;*AT THE SAME TIME!
                                ;:*  TEST 33
                                ;:*************************************************************************
021724  000004                  TST33:   SCOPE
021726  012737  000033  001122           MOV     #33,$TSTNM         ;LOAD THE NUMBER OF THIS TEST
021734  012737  022354  001360           MOV     #TST34,NEXT        ;POINT TO THE START OF THE NEXT TEST
021742  005737  001370                   TST     MODE               ;IS THIS STAGGERED MODE?
021746  100111                           BPL     6$                 ;IF NOT, DON'T DO THIS TEST
021750  104417                           DCLASM                     ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
021752  013701  001366                   MOV     PAR,R1             ;USE R1 TO BUILD PARAMETERS TO BE LOADED
021756  042701  000200                   BIC     #ODDPAR,R1         ;MAKE SURE ODD PARITY ISN'T SET
021762  052701  000100                   BIS     #PARITY,R1         ;MAKE SURE PARITY IS TURNED ON
021766  012702  000001                   MOV     #1,R2              ;USE R2 AS A LINE POINTER
021772  030237  001364          1$:      BIT     R2,LINE            ;IS THIS A VALID LINE?
021776  001411                           BEQ     3$                 ;IF NOT, SKIP TO THE NEXT LINE
022000  032701  000001                   BIT     #BIT0,R1           ;IS THIS LINE AN ODD LINE?
022004  001002                           BNE     2$                 ;IF IT'S ODD, USE EVEN PARITY
022006  052701  000200                   BIS     #ODDPAR,R1         ;IF IT'S EVEN, USE ODD PARITY
022012  010177  160034          2$:      MOV     R1,@DZLPR          ;LOAD THE LINE PARAMETER REGISTER
022016  042701  000200                   BIC     #ODDPAR,R1         ;SET UP THE NEXT PARITY TO EVEN
022022  005201                  3$:      INC     R1                 ;POINT TO THE NEXT LINE
022024  106302                           ASLB    R2                 ;MOVE THE BIT POINTER IN R2 TO THE NEXT LINE
022026  103361                           BCC     1$                 ;IF WE'RE NOT DONE, GO CHECK THE NEXT LINE
022030  005037  001372                   CLR     SAVLIN             ;CLEAR THE LINE NUMBER INDICATOR
022034  005002                           CLR     R2                 ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
022036  005003                           CLR     R3                 ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
022040  012737  000040  001376           MOV     #40,XMTCNT         ;TRANSMIT A BINARY COUNT PATTERN(00-40)
```

```
022046  012700  001422                   MOV     #TD0,R0           ;POINT TO THE DATA AREA
022052  005020                           CLR     (R0)+             ;CLEAR A DATA WORD
022054  022700  001462                   CMP     #STOP,R0          ;FINISHED ?
022060  001374                           BNE     .-6               ;NO
022062  005000                           CLR     R0                ;CLEAR OFFSET
022064  012777  023004  160004           MOV     #XMTSRV,@DZTIV    ;SET UP THE TRANSMITTER INTERRUPT VECTOR
022072  012777  022200  157772           MOV     #9$,@DZRIV        ;SET UP THE RECEIVER INTERRUPT VECTOR
022100  013777  026454  157766           MOV     DZPRT,@DZRIS      ;SET THE INTERRUPT VECTOR STATUS
022106  013777  026454  157764           MOV     DZPRT,@DZTIS      ;SET TRANSMITTER INTERRUPT PRIORITY
022114  052777  040140  157720           BIS     #RIE!TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
022122  113777  001364  157726           MOVB    LINE,@DZTCR       ;ENABLE ALL SELECTED LINES
022130  106427  000000                   MTPS    #0                ;ALLOW INTERRUPTS
022134  005037  022174          4$:      CLR     7$
022140  005037  022176                   CLR     8$
022144  032777  000100  157670  5$:      BIT     #RIE,@DZCSR       ;WHEN RX DONE; RIE WILL =0
022152  001407                           BEQ     6$                ;BR IF ALL DONE
022154  005237  022174                   INC     7$
022160  001371                           BNE     5$
022162  105237  022176                   INCB    8$
022166  100366                           BPL     5$
022170  104011                           ERROR   11                ;*RX FAILED TO FINISH (INTERRUPT)
022172  104400                  6$:      ADVANCE                   ;ADVANCE LOOP
022174  000000                  7$:      0
022176  000000                  8$:      0



                                         ;RECEIVER SERVICE ROUTINE

022200  017704  157642          9$:      MOV     @DZRBUF,R4        ;GET THE CHARACTER
022204  010401                           MOV     R4,R1             ;COPY THE RECEIVED INFORMATION
022206  000301                           SWAB    R1                ;GET THE LINE NUMBER IN THE LOWER BYTE
022210  042701  177770                   BIC     #↑C<7>,R1         ;ISOLATE THE LINE NUMBER
022214  010137  001372                   MOV     R1,SAVLIN         ;FILL LOC. FOR ERROR PRINTOUT
022220  005704                           TST     R4                ;WAS DATA VALID?
022222  100401                           BMI     10$               ;BRANCH IF YES
022224  104023                           ERROR   23                ;ERROR - DATA VALID NOT SET!
022226  006301                  10$:     ASL     R1                ;ALIGN IT ON A WORD BOUNDARY
022230  032704  010000                   BIT     #PARER,R4         ;PARITY ERROR SHOULD BE SET. IS IT?
022234  001013                           BNE     11$               ;IF SO, GO CHECK CHARACTER
022236  013737  002046  001400           MOV     DZRBUF,REGIST     ;SET UP FOR THE ERROR MESSAGE
022244  010405                           MOV     R4,R5
022246  042705  000377                   BIC     #377,R5
022252  156105  001442                   BISB    TR0(R1),R5        ;GET THE CORRECT CHARACTER
022256  052705  110000                   BIS     #DVALID!PARER,R5  ;BUILD WHAT WAS EXPECTED
022262  104006                           ERROR   6                 ;*ERROR- DID NOT GET CORRECT INFORMATION
022264  126104  001442          11$:     CMPB    TR0(R1),R4        ;CHECK THE CHARACTER. IS IT CORRECT?
022270  001407                           BEQ     12$               ;IF SO, GO SET UP NEXT CHARACTER
022272  116105  001442                   MOVB    TR0(R1),R5        ;LOAD THE CHARACTER FOR ERROR REPORTING
022276  042705  177400                   BIC     #↑C<377>,R5       ;CLEAR SIGN EXTEND
022302  042704  177400                   BIC     #↑C<377>,R4       ;REMOVE THE JUNK FROM R4, THE ACTUAL CHARACTER
022306  104005                           ERROR   5                 ;DATA ERROR
022310  005261  001442          12$:     INC     TR0(R1)           ;SET UP THE NEXT CHARACTER
022314  005203                           INC     R3                ;ADD TO THE TOTAL RECEIVED COUNT
022316  032777  040000  157516           BIT     #TIE,@DZCSR       ;ARE TRANSMISSIONS DONE?
022324  001010                           BNE     [illegible]$      ;IF NO, GO RECEIVE SOME MORE
022326  020203                           CMP     R2,R3             ;ARE ALL CHARACTERS RECEIVED?
```

```
022330  001006                             BNE     13$              ;IF NO, GO RECEIVE SOME MORE
022332  042777  000100  157502             BIC     #RIE,@DZCSR      ;DISABLE RECEIVER INTERRUPTS
022340  012716  022172                     MOV     #6$,(SP)         ;CRUNCH THE STACK
022344  000002                             RTI                      ;RETURN AND FINISH
022346  012716  022134           13$:      MOV     #4$,(SP)         ;CRUNCH THE STACK
022352  000002                             RTI                      ;GO BACK TO RECEIVER WAIT LOOP
                                          ;********************** TEST 34 ***************************
                                          ;*THIS TEST VERIFIES THAT ODD PARITY WORKS FOR ALL ODD LINES
                                          ;* SELECTED AND THAT EVEN PARITY WORKS FOR ALL EVEN LINES SELECTED
                                          ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
                                          ;*THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
                                          ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
                                          ;*YOU ARE IN "STAGGERED" MODE.
                                          ;*40(8) CHARS ARE USED FOR THIS TEST.
                                          ;*ALL SELECTED LINES WILL BE ENABLED
                                          ;*AT THE SAME TIME!
                                 ;;*  TEST 34
                                 ;;*****************************************************************
022354  000004                   TST34:    SCOPE
022356  012737  000034  001122             MOV     #34,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
022364  012737  004562  001360             MOV     #$EOP,NEXT       ;POINT TO THE END-OF-PASS HANDLER
022372  005737  001370                     TST     MODE             ;IS THIS STAGGERED MODE?
022376  100111                             BPL     6$               ;IF NOT, DON'T DO THIS TEST
022400  104417                             DCLASM                   ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
022402  013701  001366                     MOV     PAR,R1           ;USE R1 TO BUILD PARAMETERS TO BE LOADED
022406  042701  000200                     BIC     #ODDPAR,R1       ;MAKE SURE ODD PARITY ISN'T SET
022412  052701  000100                     BIS     #PARITY,R1       ;MAKE SURE PARITY IS TURNED ON
022416  012702  000001                     MOV     #1,R2            ;USE R2 AS A LINE POINTER
022422  030237  001364           1$:       BIT     R2,LINE          ;IS THIS A VALID LINE?
022426  001411                             BEQ     3$               ;IF NOT, SKIP TO THE NEXT LINE
022430  032701  000001                     BIT     #BIT0,R1         ;IS THIS LINE AN ODD LINE?
022434  001402                             BEQ     2$               ;IF IT'S EVEN, USE EVEN PARITY
022436  052701  000200                     BIS     #ODDPAR,R1       ;IF IT'S ODD, USE ODD PARITY
022442  010177  157404           2$:       MOV     R1,@DZLPR        ;LOAD THE LINE PARAMETER REGISTER
022446  042701  000200                     BIC     #ODDPAR,R1       ;SET UP THE NEXT PARITY TO EVEN
022452  005201                   3$:       INC     R1               ;POINT TO THE NEXT LINE
022454  106302                             ASLB    R2               ;MOVE THE BIT POINTER IN R2 TO THE NEXT LINE
022456  103361                             BCC     1$               ;IF WE'RE NOT DONE, GO CHECK THE NEXT LINE
022460  005037  001372                     CLR     SAVLIN           ;CLEAR THE LINE NUMBER INDICATOR
022464  005002                             CLR     R2               ;USE R2 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
022466  005003                             CLR     R3               ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
022470  012737  000040  001376             MOV     #40,XMTCNT       ;TRANSMIT A BINARY COUNT PATTERN(00-40)
022476  012700  001422                     MOV     #TD0,R0          ;POINT TO THE DATA AREA
022502  005020                             CLR     (R0)+            ;CLEAR A DATA WORD
022504  022700  001462                     CMP     #STOP,R0         ;FINISHED ?
022510  001374                             BNE     .-6              ;NO
022512  005000                             CLR     R0               ;CLEAR OFFSET
022514  012777  023004  157354             MOV     #XMTSRV,@DZTIV   ;SET UP THE TRANSMITTER INTERRUPT VECTOR
022522  012777  022630  157342             MOV     #9$,@DZRIV       ;SET UP THE RECEIVER INTERRUPT VECTOR
022530  013777  026454  157336             MOV     DZPRT,@DZRIS     ;SET THE INTERRUPT VECTOR STATUS
022536  013777  026454  157334             MOV     DZPRT,@DZTIS     ;SET TRANSMITTER INTERRUPT PRIORITY
022544  052777  040140  157270             BIS     #RIE!TIE!MSENAB,@DZCSR ;ENABLE THE DEVICE
022552  113777  001364  157276             MOVB    LINE,@DZTCR      ;ENABLE ALL SELECTED LINES
022560  106427  000000                     MTPS    #0               ;ALLOW INTERRUPTS
022564  005037  022624           4$:       CLR     7$
022570  005037  022626                     CLR     8$
```

```
022574  032777  000100  157240  5$:     BIT     #RIE,@DZCSR     ;WHEN RX DONE, RIE WILL =0
022602  001407                          BEQ     6$              ;BR IF ALL DONE
022604  005237  022624                  INC     7$
022610  001371                          BNE     5$
022612  105237  022626                  INCB    8$
022616  100366                          BPL     5$
022620  104011                          ERROR   11              ;*RX FAILED TO FINISH (INTERRUPT)
022622  104400                  6$:     ADVANCE                 ;ADVANCE LOOP
022624  000000                  7$:     0
022626  000000                  8$:     0


                                        ;RECEIVER SERVICE ROUTINE

022630  017704  157212          9$:     MOV     @DZRBUF,R4      ;GET THE CHARACTER
022634  010401                          MOV     R4,R1           ;COPY THE RECEIVED INFORMATION
022636  000301                          SWAB    R1              ;GET THE LINE NUMBER IN THE LOWER BYTE
022640  042701  177770                  BIC     #↑C<7>,R1       ;ISOLATE THE LINE NUMBER
022644  010137  001372                  MOV     R1,SAVLIN       ;FILL LOC. FOR ERROR PRINTOUT
022650  005704                          TST     R4              ;WAS DATA VALID?
022652  100401                          BMI     10$             ;BRANCH IF YES
022654  104023                          ERROR   23              ;ERROR - DATA VALID NOT SET!
022656  006301                  10$:    ASL     R1              ;ALIGN IT ON A WORD BOUNDARY
022660  032704  010000                  BIT     #PARER,R4       ;PARITY ERROR SHOULD BE SET. IS IT?
022664  001013                          BNE     11$             ;IF SO, GO CHECK CHARACTER
022666  013737  002046  001400          MOV     DZRBUF,REGIST   ;SET UP FOR THE ERROR MESSAGE
022674  010405                          MOV     R4,R5
022676  042705  000377                  BIC     #377,R5
022702  156105  001442                  BISB    TRO(R1),R5      ;GET THE CORRECT CHARACTER
022706  052705  110000                  BIS     #DVALID!PARER,R5 ;BUILD WHAT WAS EXPECTED
022712  104006                          ERROR   6               ;*ERROR- DID NOT GET CORRECT INFORMATION
022714  126104  001442          11$:    CMPB    TRO(R1),R4      ;CHECK THE CHARACTER. IS IT CORRECT?
022720  001407                          BEQ     12$             ;IF SO, GO SET UP NEXT CHARACTER
022722  116105  001442                  MOVB    TRO(R1),R5      ;LOAD THE CHARACTER FOR ERROR REPORTING
022726  042705  177400                  BIC     #↑C<377>,R5     ;CLEAR SIGN EXTEND
022732  042704  177400                  BIC     #↑C<377>,R4     ;REMOVE THE JUNK FROM R4, THE ACTUAL CHARACTER
022736  104005                          ERROR   5               ;DATA ERROR
022740  005261  001442          12$:    INC     TRO(R1)         ;SET UP THE NEXT CHARACTER
022744  005203                          INC     R3              ;ADD TO THE TOTAL RECEIVED COUNT
022746  032777  040000  157066          BIT     #TIE,@DZCSR     ;ARE TRANSMISSIONS DONE?
022754  001010                          BNE     13$             ;IF NO, GO RECEIVE SOME MORE
022756  020203                          CMP     R2,R3           ;ARE ALL CHARACTERS RECEIVED?
022760  001006                          BNE     13$             ;IF NO, GO RECEIVE SOME MORE
022762  042777  000100  157052          BIC     #RIE,@DZCSR     ;DISABLE RECEIVER INTERRUPTS
022770  012716  022622                  MOV     #6$,(SP)        ;CRUNCH THE STACK
022774  000002                          RTI                     ;RETURN AND FINISH
022776  012716  022564          13$:    MOV     #4$,(SP)        ;CRUNCH THE STACK
023002  000002                          RTI                     ;GO BACK TO RECEIVER WAIT LOOP
```

```

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

023004  117701  157034          XMTSRV: MOVB    @HDZCSR,R1              ;GET THE LINE NUMBER. IS THE TRANSMITTER
023010  100411                          BMI     1$                      ;REALLY READY? IF SO, GO LOAD THE CHARACTER
023012  013700  001372                  MOV     SAVLIN,R0               ;ADJUST LOCATION SAVLIN
023016  042701  177770                  BIC     #↑C<7>,R1               ;ISOLATE THE LINE NUMBER
023022  010137  001372                  MOV     R1,SAVLIN               ;FOR ERROR PRINTOUT
023026  104003                          ERROR   3                       ;*TRANSMITTER NOT READY- FALSE INTERRUPT
023030  010037  001372                  MOV     R0,SAVLIN               ;RESET SAVLIN TO PREVIOUS VALUE
023034  042701  177770          1$:     BIC     #↑C<7>,R1               ;ISOLATE THE LINE NUMBER
023040  006301                          ASL     R1                      ;MAKE SURE IT REFERENCES A WORD BOUNDARY
023042  116177  001422  157016          MOVB    TD0(R1),@DZTDR          ;LOAD THE CURRENT CHARACTER FOR THIS LINE
023050  005261  001422                  INC     TD0(R1)                 ;SET UP NEXT CHARACTER FOR THIS LINE
023054  005202                          INC     R2                      ;UP THE NUMBER OF TRANSMISSIONS
023056  023761  001376  001422          CMP     XMTCNT,TD0(R1)          ;HAVE WE DONE ALL PATTERNS ON THIS LINE?
023064  001015                          BNE     4$                      ;IF NOT, KEEP ON TRANSMITTING
023066  012700  000001                  MOV     #1,R0                   ;SET UP A DESELECTION POINTER
023072  006201                          ASR     R1                      ;GET THE LINE NUMBER AGAIN
023074  005301                  2$:     DEC     R1                      ;REDUCE THE COUNT. WAS THIS THE LINE?
023076  100402                          BMI     3$                      ;IF SO, GO DISABLE THE ENABLE BIT FOR IT
023100  006300                          ASL     R0                      ;MOVE THE POINTER TO THE NEXT LINE
023102  000774                          BR      2$                      ;GO CHECK THE NEXT LINE
023104  140077  156746          3$:     BICB    R0,@DZTCR               ;DISABLE THE LINE POINTED TO BY R0
023110  001003                          BNE     4$                      ;IF MORE LINES ARE ACTIVE , GO CONTINUE TRANSMIT
023112  042777  040000  156722          BIC     #TIE,@DZCSR             ;IF NOT, DISABLE TRANSMITTER INTERRUPTS
023120  000002                  4$:     RTI                             ;RETURN TO THE TIMING LOOP

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

023122  012737  000004  001222  BUILD:  MOV     #4,$TMP2                ;ROTATE 4 BITS BACK INTO $TMP1
023130  006037  001224          1$:     ROR     $TMP3                   ;GET THE BITS FROM $TMP3, THE HIGH BYTE
023134  006037  001220                  ROR     $TMP1                   ;OF THE RELATIVE TIME COUNTER. PUT THEM BACK
023140  005337  001222                  DEC     $TMP2                   ;INTO $TMP1 USING THE CARRY BIT WITH
                                                                        ;ROTATE INSTRUCTIONS
023144  001371                          BNE     1$                      ;REDUCE COUNT. ALL BITS BACK? IF NOT, GET MORE
023146  000207                          RTS     PC                      ;RETURN TO CALLING TEST

```

```
                                                       ;RECEIVER SERVICE ROUTINE

023150  105777  156666            RXISR1: TSTB    @DZCSR                  ;IS THE RECEIVER REALLY READY?
023154  100401                            BMI     1$                      ;IF SO, GO SERVICE IT
023156  104004                            ERROR   4                       ;*ERROR- RECEIVER DONE FLAG ISN'T SET
023160  017704  156662            1$:     MOV     @DZRBUF,R4              ;SAVE THE RECEIVER INFORMATION
023164  100401                            BMI     2$                      ;IF IT WAS VALID, GO PROCESS IT
023166  104023                            ERROR   23                      ;ERROR- DATA VALID WASN'T SET
023170  032704  070000            2$:     BIT     #OVRRUN!FRMERR!PARER,R4 ;ARE ANY ERROR FLAGS SET?
023174  001403                            BEQ     3$                      ;IF NOT, GO CONTINUE PROCESSING
023176  013700  002046                    MOV     DZRBUF,R0               ;SET UP FOR ERROR REPORTING
023202  104002                            ERROR   2                       ;ERROR- RECEIVER ERROR FLAG SET
023204  010401                    3$:     MOV     R4,R1                   ;COPY THE RECEIVER INFORMATION
023206  000301                            SWAB    R1                      ;GET THE LINE NUMBER IN THE LOWER BYTE
023210  042701  177770                    BIC     #↑C<7>,R1               ;ISOLATE THE LINE NUMBER
023214  006301                            ASL     R1                      ;ALIGN IT ON A WORD BOUNDARY
023216  120461  001442                    CMPB    R4,TRO(R1)              ;IS THE CHARACTER WHAT IT SHOULD BE?
023222  001413                            BEQ     4$                      ;IF SO,GO CONTINUE PROCESSING
023224  116105  001442                    MOVB    TRO(R1),R5              ;GET WHAT WAS EXPECTED FOR ERROR REPORTING
023230  042705  177400                    BIC     #↑C<377>,R5             ;ELIMINATE PROPAGATED SIGN
023234  042704  177400                    BIC     #↑C<377>,R4             ;ISOLATE THE ACTUAL CHARACTER
023240  010137  001372                    MOV     R1,SAVLIN               ;GET THE LINE NUMBER OF THE RECEIVER ERROR
023244  006237  001372                    ASR     SAVLIN                  ;ALIGN IT CORRECTLY FOR REPORTING
023250  104005                            ERROR   5                       ;*DATA ERROR
023252  005261  001442            4$:     INC     TRO(R1)                 ;SET UP THE NEXT EXPECTED CHARACTER
023256  005203                            INC     R3                      ;INCREMENT THE COUNT OF RECEIVED CHARACTERS
023260  032761  000020  001442            BIT     #20,TRO(R1)             ;HAVE ALL CHARACTERS BEEN RECEIVED?
023266  001402                            BEQ     5$                      ;IF NOT, GO RECEIVE SOME MORE
023270  020203                            CMP     R2,R3                   ;HAVE WE RECEIVED ALL CHARACTERS?
023272  001401                            BEQ     6$                      ;IF SO,GO DETERMINE THE TIMING
023274  000002                    5$:     RTI                             ;GO CONTINUE TIMING AND ALLOW INTERRUPTS
023276  004737  023122            6$:     JSR     PC,BUILD                ;GET THE RELATIVE TIME (SIGNIFICANT BITS)

023302  013700  023362                    MOV     OFFSET,R0               ;GET POINTER
023306  013760  001220  002102            MOV     $TMP1,TMTBL(R0)         ; SAVE THIS TEST'S TIME
023314  005737  023362                    TST     OFFSET                  ;FIRST TEST?
023320  001414                            BEQ     7$                      ;IF NOT, GO CHECK THE TIME
023322  005740                            TST     -(R0)                   ;POINT TO THE PREVIOUS  TIME TAKEN
023324  026037  002102  001220            CMP     TMTBL(R0),$TMP1         ;IS THIS TIME WHAT IT SHOULD BE?
023332  101007                            BHI     7$                      ;IF SO, GO TO THE NEXT TEST
023334  016005  002102                    MOV     TMTBL(R0),R5            ;PLACE WHAT WAS EXPECTED IN R5
023340  010137  001372                    MOV     R1,SAVLIN               ;GET THE LINE NUMBER OF THE RECEIVER
023344  006237  001372                    ASR     SAVLIN                  ;MAKE SURE IT'S THE LINE NUMBER
023350  104021                            ERROR   21                      ;TIMING ERROR
023352  042777  000140  156462    7$:     BIC     #RIE!MSENAB,@DZCSR      ;DISABLE THE DEVICE
023360  000002                            RTI                             ;RETURN TO THE PROGRAM
023362  000000                    OFFSET: 0
```

```
                                         ;DZ11 ECHO/CABLE TEST
                                         ;COPYRIGHT 1977, 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>
                                                         ;*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

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


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

 023364  012706  001120           XSTART: MOV     #STACK,SP        ;SET UP PROCESSOR STACK
 023370  106427  000340                   MTPS    #PR7             ;LOCK OUT INTERRUPTS
 023374  012737  023364  001126           MOV     #XSTART,$LPADR   ;SET UP IN CASE OF POWER FAIL
 023402  005037  025560                   CLR     STFLG            ;CLEAR TEST START FLAG
 023406  005037  001242                   CLR     $PASS            ;CLEAR PASS COUNT
 023412  005037  001132                   CLR     $ERTTL           ;CLEAR ERROR COUNT
 023416  105037  001123                   CLRB    $ERFLG           ;CLEAR ERROR FLAG
```

```
023422  005037  025564                    CLR     LAST            ;CLEAR LAST ERROR PC
023426  032777  000001  155524  VEC1:     BIT     #SW00,@SWR      ;IF SW00=1, GET NEW VECTOR
023434  001465                            BEQ     OTHER           ;AND CSR
023436  012701  000300          VEC2:     MOV     #300,R1
023442  012702  000302                    MOV     #302,R2
023446  010221                  1$:       MOV     R2,(R1)+        ;RESTORE TRAPCATCHER
023450  005022                            CLR     (R2)+           ;IN FLOATING VECTOR AREA
023452  022122                            CMP     (R1)+,(R2)+     ;UPDATE THE POINTERS
023454  020127  001000                    CMP     R1,#1000
023460  001372                            BNE     1$
023462  104403                            INSTR                   ;INPUT ADDRESS OF DEVICE VECTOR
023464  025612                            MVECTOR                 ;MESSAGE "VECTOR ADDRESS-"
023466  104405                            PARAM                   ;CONVERT STRING TO OCTAL
023470  000300                            300                     ;LOW LIMIT
023472  000770                            770                     ;HIGH  LIMIT
023474  002072                            DZRIV                   ;LOCATIONS TO BE FILLED
023476     003                  .BYTE     3                       ;LSB MASK
023477     004                  .BYTE     4                       ;NUMBER OF LOCATIONS
023500  104403                            INSTR                   ;INPUT ADDRESS OF DEVICE CSR
023502  025634                            MREGAD                  ;MESSAGE "CONTROL REGISTER ADDRESS-"
023504  104405                            PARAM                   ;CONVERT STRING TO OCTAL
023506  160000                            160000                  ;LOW LIMIT
023510  163700                            163700                  ;HIGH LIMIT
023512  002042                            DZCSR                   ;LOCATIONS TO BE FILLED
023514     007                  .BYTE     7                       ;LSB MASK
023515     001                  .BYTE     1                       ;NUMBER OF LOCATIONS
023516  013737  002042  002046            MOV     DZCSR,DZRBUF    ;BEGIN BUILDING DEVICE ADDRESSES
023524  062737  000002  002046            ADD     #2,DZRBUF       ;FORM THE READ BUFFER ADDRESS
023532  013737  002046  002052            MOV     DZRBUF,DZLPR    ;REMEMBER THAT THIS IS ALSO LINE PARAMETER REG.
023540  013737  002046  002056            MOV     DZRBUF,DZTCR    ;BEGIN BUILDING TRANSMITTER CONTROL REGISTER
023546  062737  000002  002056            ADD     #2,DZTCR        ;FORM THE TRANSMITTER CONTROL REGISTER POINTER
023554  013737  002056  002060            MOV     DZTCR,HDZTCR
023562  005237  002060                    INC     HDZTCR
023566  013737  002056  002066            MOV     DZTCR,DZTDR     ;BEGIN FORMING TRANSMITTER DATA REGISTER
023574  062737  000002  002066            ADD     #2,DZTDR        ;FORM THE TRANSMITTER DATA REGISTER
023602  013737  002066  002062            MOV     DZTDR,DZMSR
023610  032777  000002  155342  OTHER:    BIT     #SW01,@SWR      ;RESELECT OF TEST?
023616  001427                            BEQ     XBEGIN          ;IF NOT, SKIP ASKING WHICH ONE
023620  104403                            INSTR                   ;INPUT WHICH TEST YOU ARE RUNNING
023622  026020                            MWHICH                  ;ECHO OR CABLE
023624  104416                            PAWCH                   ;SET FLAG
023626  025556                            WCHFLG                  ;THIS FLAG
023630  104403                  BAUD:     INSTR                   ;INPUT BAUD RATE
023632  025742                            MSPEED                  ;MESSAGE "BAUD RATE-"
023634  104415                            PARMD                   ;CONVERT DECIMAL STRING TO OCTAL
023636  000062                            50.                     ;LOW LIMIT
023640  022600                            9600.                   ;HIGH LIMIT
023642  025574                            LINESP                  ;LOCATION TO BE FILLED
023644     000                  .BYTE     0                       ;LSB MASK
023645     001                  .BYTE     1                       ;NUMBER OF LOCATIONS
023646  104413                  LINEX:    DEVICE.CLR              ;CLEAR DEVICE
023650  005037  025560                    CLR     STFLG           ;CLEAR PROGRAM START FLAG
023654  104403                            INSTR                   ;INPUT LINE NUMBER
023656  025732                            MLINE                   ;MESSAGE "LINE NUMBER-"
023660  104405                            PARAM                   ;CONVERT  STRING TO OCTAL
023662  000000                            0                       ;LOW LIMIT
```

```
023664  000007                           7                            ;HIGH LIMIT
023666  001372                           SAVLIN                       ;LOCATION TO BE FILLED
023670     000                  .BYTE    0                            ;LSB MASK
023671     001                           .BYTE   1                    ;NUMBER OF LOCATIONS
023672  004537  025362                   JSR     R5,SET

023676  106427  000340          XBEGIN:  MTPS    #PR7                 ;LOCK OUT INTERRUPTS
023702  012706  001120                   MOV     #STACK,SP            ;SET UP PROCESSOR STACK
023706  005037  025562                   CLR     LOCKUP               ;CLEAR TIMEOUT
023712  005737  025556                   TST     WCHFLG               ;ECHO OR CABLE TEST ?
023716  001413                           BEQ     2$                   ;ECHO
023720  012737  024434  001126           MOV     #TEST2,$LPADR        ;CABLE TEST
023726  005737  025560                   TST     STFLG                ;ARE YOU LOOPING ?
023732  001017                           BNE     1$                   ;YES
023734  005137  025560                   COM     STFLG                ;NO
023740  104402  026113                   TYPE    ,MCABLE              ;TYPE CABLE TEST
023744  000412                           BR      1$
023746  012737  023776  001126  2$:      MOV     #TEST1,$LPADR        ;SET UP ECHO TEST
023754  005737  025560                   TST     STFLG                ;ARE YOU LOOPING ?
023760  001004                           BNE     1$                   ;YES
023762  005137  025560                   COM     STFLG                ;NO
023766  104402  026066                   TYPE    ,MTERM               ;TYPE ECHO TEST
023772  000177  155130          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)

023776  104413                  TEST1:   DEVICE.CLR                   ;CLEAR DZ11
024000  012737  000001  001122           MOV     #1,$TSTNM
024006  013777  025602  156042           MOV     NUMTCR,@DZTCR        ;SET TCR BIT
024014  013737  025600  001366           MOV     NUMLIN,PAR           ;SET PARAMETERS
024022  053737  025576  001366           BIS     SPEED,PAR            ;SET BAUD RATE
024030  013777  001366  156014           MOV     PAR,@DZLPR           ;LOAD PARAM.
024036  012777  000040  155776           MOV     #MSENAB,@DZCSR       ;SET SCANN ENABLE
024044  005004                           CLR     R4
024046  012705  026130                   MOV     #MQUICK,R5           ;SET MESSAGE BUFFER
024052  005777  155764          3$:      TST     @DZCSR               ;TRDY?
024056  100404                           BMI     2$                   ;BR IF YES
024060  104414                           DELAY                        ;WAIT
024062  005304                           DEC     R4
024064  001372                           BNE     3$
024066  104003                           ERROR   3                    ;NO TRDY SET! WHY?
024070  005004                  2$:      CLR     R4                   ;RESET COUNTER TO 0
024072  112577  155770                   MOVB    (R5)+,@DZTDR         ;LOAD CHAR
024076  001365                           BNE     3$
024100  004737  007360                   JSR     PC,SERV.G            ;(↑G)?
024104  122777  000377  155046           CMPB    #377,@SWR            ;RE-DO QUICK BROWN?
024112  001731                           BEQ     TEST1                ;BR IF REPEAT PATTERN
024114  104413                           DEVICE.CLR
024116  106427  000340                   MTPS    #PR7                 ;LOCK OUT INTERRUPTS
024122  012737  025072  001360           MOV     #XEOP,NEXT
024130  104413                           DEVICE.CLR
024132  013737  025600  001366           MOV     NUMLIN,PAR           ;SELECT LINE # & SET INTERRUPT ENABLE
024140  053737  025576  001366           BIS     SPEED,PAR            ;SET LINE SPEED AND
                                                                      ;CHARACTER LENGTH (TRANS. & REC.)
024146  052737  010000  001366           BIS     #RCVON,PAR           ;MAKE SURE RECEIVER IS TURNED ON
```

```
024154  013777  001366  155670          MOV     PAR,@DZLPR           ;LOAD THE LINE PARAMETER REGISTER
024162  012777  024236  155702          MOV     #INTSVC,@DZRIV       ;SET UP INTERRUPT SERVICE
024170  013777  025604  155676          MOV     PRIO,@DZRIS          ;AND LEVEL
024176  106437  026456                  MTPS    @#LESS1              ;ALLOW INTERRUPTS
024202  012777  000140  155632          MOV     #RIE!MSENAB,@DZCSR ;SET RECEIVER INTERRUPT ENABLE
024210  104402  025760                  TYPE    ,MCHAR               ;TYPE "ANY CHARACTER"
024214  105777  154744          1$:     TSTB    @$TKS                ;IF SOMBODY HITS A KEY- GET NEW LINE #
024220  100375                          BPL     1$                   ;LOOP HERE
024222  005777  154740                  TST     @$TKB                ;CLEAR CHAR
024226  004737  007360                  JSR     PC,SERV.G            ;MAKE SURE IT WASN'T <↑G>
024232  000137  023646                  JMP     LINEX                ;



                                        ;THE FOLLOWING IS THE RECEIVER INTERRUPT SVC ROUTINE
024236  105777  155600          INTSVC: TSTB    @DZCSR               ;TEST REC. FLAG
024242  100401                          BMI     .+4
024244  104004                          ERROR   4                    ;ERROR - INTERRUPT NOT CAUSED BY FLAG
024246  017737  155574  025606          MOV     @DZRBUF,RECDAT
024254  100401                          BMI     .+4
024256  104023                          ERROR   23                   ;NON- VALID CHARACTER
024260  032737  020000  025606          BIT     #BIT13,RECDAT        ;CHECK FOR FRAMING ERROR
024266  001401                          BEQ     .+4                  ;BR IF NO ERROR
024270  104025                          ERROR   25                   ;EITHER SOMBODY HIT THE
                                                                     ;"BREAK KEY" OR YOU HAVE AN ERROR!
024272  113737  025606  025610          MOVB    RECDAT,TBUF          ;MOVE CHARACTER TO OUTPUT AREA
024300  113737  025606  010620          MOVB    RECDAT,INBUF         ;MOVE CHARACTER TO CHECK FOR ↑C
024306  042737  177600  010620          BIC     #↑C<177>,INBUF       ;STRIP JUNK PLUS PARITY
024314  042737  174377  025606          BIC     #174377,RECDAT       ;SAVE ONLY LINE  NUMBER
024322  000337  025606                  SWAB    RECDAT
024326  023737  001372  025606          CMP     SAVLIN,RECDAT        ;DOES THE LINE # COMPARE?
024334  001401                          BEQ     .+4
024336  104015                          ERROR   15                   ;#WRONG LINE NUMBER
024340  012777  000040  155474          MOV     #MSENAB,@DZCSR       ;START THE TRANSMITTERS SCANNER
024346  123727  010620  000003          CMPB    INBUF,#3                     ;IS IT A ↑C ?
024354  001004                          BNE     1$                   ;NO
024356  104413                          DEVICE.CLR
024360  012716  025072                  MOV     #XEOP,(SP)           ;CRUNCH STACK
024364  000002                          RTI
024366  005003                  1$:     CLR     R3                   ;INITIALIZE DELAY
024370  013777  025602  155460          MOV     NUMTCR,@DZTCR        ;ENABLE THE LINE
024376  005777  155440          10$:    TST     @DZCSR               ;TRANSMITTER READY?
024402  100403                          BMI     2$                   ;IF YES BRANCH
024404  005203                          INC     R3                   ;INCREMENT DELAY
024406  001373                          BNE     10$                  ;DELAY DONE?
024410  104003                          ERROR   3                    ;TRANSMIT READY NOT SET!
024412  113777  025610  155446  2$:     MOVB    TBUF,@DZTDR          ;TRANSMIT THE CHARACTER
024420  012777  000140  155414          MOV     #RIE!MSENAB,@DZCSR        ;RESTART THE RECEIVER
024426  005077  155424                  CLR     @DZTCR               ;CLEAR TCR BIT
024432  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
024434  106427  000340          TEST2:  MTPS    #PR7                 ;DISABLE INTERRUPTS
024440  012737  000002  001122          MOV     #2,$TSTNM
```

```
024446  012737  025072  001360          MOV     #XEOP,NEXT
024454  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!
024456  012737  024464  001362          MOV     #1$,LOCK        ;LOOP
024464  113777  025602  155366  1$:     MOVB    NUMTCR,@HDZTCR  ;SET DTR
024472  005005                          CLR     R5              ;
024474  153705  025602                  BISB    NUMTCR,R5       ;BUILD EXPECTED
024500  000305                          SWAB    R5              ;PUT IN HIGH BYTE
024502  153705  025602                  BISB    NUMTCR,R5
024506  104414                          DELAY                   ;WAIT FOR CABLE DELAY
024510  017704  155346                  MOV     @DZMSR,R4       ;READY MODEM BITS
024514  020504                          CMP     R5,R4           ;ARE THEY OK?
024516  001401                          BEQ     2$              ;BR IF YES
024520  104022                          ERROR   22              ;IS THE TEST CONNECTOR ON?
                                                                ;HAS RIGHT LINE BEEN SELECTED?
                                                                ;IF SO- YOU HAVE A PROBLEM!
                                                                ;MODEM BITS NOT RIGHT
024522  104401                  2$:     SCOP1                   ;LOOP
024524  104413                  3$:     DEVICE.CLR              ;INIT DZ11
024526  013737  025576  001366          MOV     SPEED,PAR       ;SET LINE SPEED
024534  053737  025600  001366          BIS     NUMLIN,PAR      ;SELECT LINE # & REC.  INTERRUPT ENABLE
024542  052737  010000  001366          BIS     #RCVON,PAR      ;ENABLE THE RECEIVER FOR THIS LINE
024550  052777  040140  155264          BIS     #TIE!RIE!MSENAB,@DZCSR ;SET TRANSMITTER INTERRUPT ENABLE
024556  012777  024672  155306          MOV     #INTREC,@DZRIV  ;SET UP INTR SERVICE
024564  013777  025604  155302          MOV     PRIO,@DZRIS     ;SET UP LEVEL
024572  012777  025052  155276          MOV     #INTRAN,@DZTIV  ;SET UP INTR SERVICE
024600  013777  025604  155272          MOV     PRIO,@DZTIS     ;SET UP LEVEL
024606  005001                          CLR     R1              ;RX DATA POINTER- SET TO 0
024610  005002                          CLR     R2              ;TX DATA POINTER- SET TO 0
024612  013777  001366  155232          MOV     PAR,@DZLPR      ;SET THE PARAMETERS AND TURN ON RECEIVER
024620  106437  026456                  MTPS    @#LESS1         ;ALLOW INTERRUPTS
024624  013777  025602  155224          MOV     NUMTCR,@DZTCR   ;SET UP TCR BIT

                                        ;YOU RETURN HERE AFTER EVERY RECEIVER INTERRUPT
024632  105777  154326          SPIN:   TSTB    @$TKS           ;IF SOMEBODY HITS A KEY- GET A NEW LINE #
024636  100006                          BPL     1$              ;BR IF NO KEY HIT
024640  005777  154322                  TST     @$TKB           ;CLEAR CHAR
024644  004737  007360                  JSR     PC,SERV.G       ;MAKE SURE IT WASN'T <↑G>
024650  000137  023646                  JMP     LINEX           ;SW02=1
024654  005237  025562          1$:     INC     LOCKUP          ;INC TIMEOUT FLAG
024660  001364                          BNE     SPIN            ;IF NOT 0  RETURN SPINNING
024662  104011                          ERROR   11              ;*RECEIVER FAILED TO INTERRUPT  CHECK CABLE/TERMINATOR
024664  104413                  QUITS:  DEVICE.CLR
024666  000137  025072                  JMP     XEOP            ;CALL FOR END OF PASS
024672  005037  025562          INTREC: CLR     LOCKUP          ;CLEAR TIMEOUT FLAG
024676  105777  155140                  TSTB    @DZCSR          ;TEST REC DONE
024702  100401                          BMI     .+4             ;YES
024704  104004                          ERROR   4               ;*FALSE INTERRUPT
024706  017737  155134  025606          MOV     @DZRBUF,RECDAT  ;SAVE WORD
024714  100401                          BMI     .+4
024716  104023                          ERROR   23              ;*NON VALID CHARACTER
024720  032737  040000  025606          BIT     #BIT14,RECDAT   ;DATA OVERRUN ?
024726  001401                          BEQ     .+4             ;NO
```

```
024730  104024                                  ERROR   24                      ;*YES
024732  032737  020000  025606                  BIT     #BIT13,RECDAT           ;FRAMING ERROR ?
024740  001401                                  BEQ     .+4                     ;NO
024742  104025                                  ERROR   25                      ;*YES
024744  032737  010000  025606                  BIT     #BIT12,RECDAT           ;PARITY ERROR ?
024752  001401                                  BEQ     .+4                     ;NO
024754  104026                                  ERROR   26                      ;*YES
024756  110105                                  MOVB    R1,R5                   ;SET EXPECTED
024760  042705  177400                          BIC     #↑C<377>,R5             ;CLEAR HIGH BYTE
024764  113704  025606                          MOVB    RECDAT,R4               ;GET FOUND
024770  042704  177400                          BIC     #↑C<377>,R4             ;CLEAR HIGH BYTE
024774  020504                                  CMP     R5,R4   ;OK?
024776  001401                                  BEQ     .+4
025000  104005                                  ERROR   5                       ;DATA ERROR
025002  042737  174377  025606                  BIC     #174377,RECDAT          ;SAVE ONLY LINE NUMBER
025010  000337  025606                          SWAB    RECDAT
025014  023737  001372  025606                  CMP     SAVLIN,RECDAT           ;DOES THE LINE # COMPARE ?
025022  001401                                  BEQ     .+4                     ;YES
025024  104015                                  ERROR   15                      ;*WRONG LINE #
025026  120127  000377                          CMPB    R1,#377                 ;LAST CHARACTER ?
025032  001003                                  BNE     1$                      ;NO
025034  012716  024664                          MOV     #QUITS,(SP)             ;CRUNCH STACK
025040  000403                                  BR      2$
025042  105201                          1$:     INCB    R1                      ;UPDATE EXPECTED DATA
025044  012716  024632                          MOV     #SPIN,(SP)              ;CRUNCH STACK
025050  000002                          2$:     RTI

025052  005777  154764                  INTRAN: TST     @DZCSR  ;TEST TRANSMIT FLAG
025056  100401                                  BMI     .+4
025060  104003                                  ERROR   3                       ;*FALSE INTERRUPT
025062  110277  155000                          MOVB    R2,@DZTDR               ;TRANSMIT A CHARACTER
025066  105202                                  INCB    R2                      ;UPDATE TX DATA
025070  000002                                  RTI     ;RETURN
```

```

                                                ;END OF PASS
                                                ;RESTART TEST

025072  104402                          XEOP:   TYPE                            ;TYPE NAME OF TEST
025074  025670                                  MPASS
025076  005037  025564                          CLR     LAST                    ;CLEAR LAST ERROR PC
025102  105037  001123                          CLRB    $ERFLG                  ;CLEAR ERROR FLAG
025106  000137  023676                  RSTRT:  JMP     XBEGIN

                                                ;CONVERT DECIMAL ASCII STRING TO OCTAL
025112  011605                          .PARMD: MOV     (SP),R5
025114  012537  025276                          MOV     (R5)+,6$
025120  012537  025300                          MOV     (R5)+,7$
025124  012537  025302                          MOV     (R5)+,8$
025130  112537  025304                          MOVB    (R5)+,9$
025134  112537  025305                          MOVB    (R5)+,10$
025140  010516                                  MOV     R5,(SP)
025142  005005                          2$:     CLR     R5
025144  012704  010620                          MOV     #INBUF,R4
025150  122714  000015                          CMPB    #15,(R4)
025154  001424                                  BEQ     3$
025156  121427  000060                  1$:     CMPB    (R4),#'0
025162  002421                                  BLT     3$
025164  121427  000071                          CMPB    (R4),#'9
025170  003016                                  BGT     3$
025172  142714  000060                          BICB    #'0,(R4)
025176  005002                                  CLR     R2
025200  152402                                  BISB    (R4)+,R2
025202  060205                                  ADD     R2,R5
025204  122714  000015                          CMPB    #15,(R4)
025210  001410                                  BEQ     4$
025212  006305                                  ASL     R5      ;X2
025214  010502                                  MOV     R5,R2   ;SAVE X2
025216  006305                                  ASL     R5      ;X4
025220  006305                                  ASL     R5      ;X8
025222  060205                                  ADD     R2,R5   ;TIMES 10
025224  000754                                  BR      1$
025226  104404                          3$:     INSTER
025230  000744                                  BR      2$

                                                ;TEST TO SEE IF NUMBER IS WITHIN LIMITS

025232  020537  025300                  4$:     CMP     R5,7$
025236  101373                                  BHI     3$
025240  020537  025276                          CMP     R5,6$
025244  103770                                  BLO     3$
025246  133705  025304                          BITB    9$,R5
025252  001365                                  BNE     3$

                                                ;STORE NUMBER AT SPECIFIED ADDRESS

025254  013704  025302                          MOV     8$,R4
025260  010524                          5$:     MOV     R5,(R4)+
025262  062705  000002                          ADD     #2,R5
025266  105337  025305                          DECB    10$
```

```
025272  001372                                  BNE     5$
025274  000002                                  RTI
025276  000000                          6$:     0
025300  000000                          7$:     0
025302  000000                          8$:     0
025304     000                          9$:     .BYTE 0
025305     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

025306  017605  000000          .PAWCH:MOV      @(SP),R5
025312  142737  000040  010620          BICB    #40,INBUF               ;SET FOR LOWER CASE INPUT
025320  122737  000105  010620          CMPB    #'E,INBUF               ;IS IT "E" ?
025326  001002                          BNE     1$
025330  105015                          CLRB    (R5)                    ;000
025332  000406                          BR      2$
025334  122737  000103  010620  1$:     CMPB    #'C,INBUF               ;IS IT "C" ?
025342  001005                          BNE     3$
025344  112715  177777                  MOVB    #-1,(R5)                ;3177
025350  062716  000002          2$:     ADD     #2,(SP)
025354  000002                          RTI
025356  104404                  3$:     INSTER                          ;RETRY
025360  000752                          BR      .PAWCH



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

025362  013737  001372  025600  SET:    MOV     SAVLIN,NUMLIN           ;SAVE SAVLIN
025370  013700  001372          XTCR0:  MOV     SAVLIN,R0               ;COPY THE LINE NUMBER FOR LOOP CONTROL
025374  005037  025602                  CLR     NUMTCR                  ;SET A DEFAULT OF LINE 0 OR NO LINES
025400  012702  000001                  MOV     #1,R2                   ;SET A BIT POINTER TO THE FIRST LINE
025404  005300                  XTCR1:  DEC     R0                      ;REDUCE THE INDICATOR.IS IT MINUS YET?
025406  100402                          BMI     SET1                    ;IF SO, R2 POINTS TO THE RIGHT LINE
025410  006302                          ASL     R2                      ;IF NOT, MOVE THE POINTER TO THE NEXT LINE
025412  000774                          BR      XTCR1                   ;GO SEE IF THIS LINE IS THE ONE
025414  012701  025456          SET1:   MOV     #TABLE2,R1
025420  010237  025602                  MOV     R2,NUMTCR               ;COPY THE CORRECT BIT POINTER
025424  022137  025574          1$:     CMP     (R1)+,LINESP
025430  001407                          BEQ     2$
025432  005721                          TST     (R1)+                   ;IS IT THE END OF TABLE?
025434  001373                          BNE     1$                      ;NO
025436  104402  025704                  TYPE    ,MINVAL                 ;INVALID BAUD RATE,BEGIN AGAIN
025442  012705  023630                  MOV     #BAUD,R5                ;JUMP TO BAUD THRU R5
025446  000402                          BR      3$
025450  011137  025576          2$:     MOV     (R1),SPEED              ;SET UP BAUD RATE
025454  000205                  3$:     RTS     R5
```

```
                                            ;THE FOLLOWING IS A TABLE OF LEGAL BAUD RATES (8 BITS/CHAR)
025456  000062                     TABLE2:  .WORD   50.                ;50 BAUD
025460  010070                              .WORD   10070
025462  000113                              .WORD   75.                ;75 BAUD
025464  010470                              .WORD   10470              ;
025466  000156                              .WORD   110.               ;110 BAUD
025470  011070                              .WORD   11070              ;TWO STOP BITS
025472  000207                              .WORD   135.               ;134.5 BAUD
025474  011470                              .WORD   11470              ;TWO STOP BITS
025476  000226                              .WORD   150.               ;150 BAUD
025500  012070                              .WORD   12070              ;TWO STOP BITS
025502  000454                              .WORD   300.               ;300 BAUD
025504  012430                              .WORD   12430              ;ONE STOP BIT
025506  001130                              .WORD   600.               ;600 BAUD
025510  013030                              .WORD   13030              ;ONE STOP BIT
025512  002260                              .WORD   1200.              ;1200 BAUD
025514  013430                              .WORD   13430              ;ONE STOP BIT
025516  003410                              .WORD   1800.              ;1800 BAUD
025520  014030                              .WORD   14030              ;ONE STOP BIT
025522  003720                              .WORD   2000.              ;2000 BAUD
025524  014430                              .WORD   14430              ;ONE STOP BIT
025526  004540                              .WORD   2400.              ;2400 BAUD
025530  015030                              .WORD   15030              ;ONE STOP BIT
025532  007020                              .WORD   3600.              ;3600 BAUD
025534  015430                              .WORD   15430              ;ONE STOP BIT
025536  011300                              .WORD   4800.              ;4800 BAUD
025540  016030                              .WORD   16030              ;ONE STOP BIT
025542  016040                              .WORD   7200.              ;7200 BAUD
025544  016430                              .WORD   16430              ;ONE STOP BIT
025546  022600                              .WORD   9600.              ;9600 BAUD
025550  017070                              .WORD   17070              ;
025552  177777  000000                      .WORD   -1,0               ;TABLE TERMINATOR


025556  000000                     WCHFLG:0                            ;ECHO OR CABLE FLAG
025560  000000                     STFLG: 0                            ;PROGRAM START FLAG
025562  000000                     LOCKUP: 0                           ;TIMEOUT FLAG
025564  000000                     LAST:   0                           ;LAST ERROR PC
025566  000000                     TDATA:  0
025570  000000                     RDATA:  0
025572  000000                     BYTCNT: 0
025574  000156                     LINESP: 110.                        ;DEFAULT BAUD RATE
025576  006307                     SPEED:  6307                        ;DEFAULT 110 BAUD, 8 BITS/CHAR,
                                                                       ;FDX, 2 STOP BITS
025600  000100                     NUMLIN: 100                         ;DEFAULT VALUE, REC. INTERRUPT ENABLED

025602  000001                     NUMTCR: 1                           ;DEFAULT VALUE,TCR BIT 0
025604  000240                     PRIO:   240                         ;DEFAULT DEVICE PRIORITY 5
025606  000000                     RECDAT: 0
025610  000000                     TBUF:   0
025612  053200  041505  047524     MVECTO: .ASCIZ  <200>/VECTOR ADDRESS- /
025634  041600  047117  051124     MREGAD: .ASCIZ  <200>/CONTROL REGISTER ADDRESS- /
025670  050200  051501  020123     MPASS:  .ASCIZ  <200>/PASS DONE./
025704  044600  053116  046101     MINVAL: .ASCIZ  <200>/INVALID BAUD RATE - /
025732  046200  047111  035105     MLINE:  .ASCIZ  <200>/LINE: /
025742  041200  052501  020104     MSPEED: .ASCIZ  <200>/BAUD RATE - /
```

```
 025760  052200  050131  020105  MCHAR:  .ASCIZ  <200>/TYPE A CHAR. ON DZ11 TERMINAL /
 026020  053600  044510  044103  MWHICH: .ASCIZ  <200>/WHICH TEST ? ECHO OR CABLE (E OR C) /
 026066  052200  051105  044515  MTERM:  .ASCIZ  <200>/TERMINAL ECHO TEST /
 026113     200  040503  046102  MCABLE: .ASCIZ  <200>/CABLE TEST /
 026130  006777  177777  177412  MQUICK: .ASCII  <377><15><377><377><12><377><377>
 026137     124  042510  050440          .ASCII  /THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789/
 026234  006777  177777  177412          .ASCII  <377><15><377><377><12><377><377><377><0>
         026246                  .EVEN
                                 ;;*******************************************************
                                         ;UTILITIES
                                 ;;*******************************************************

                                 ;THIS UTILITY CALCULATES PRIORITY LEVEL,SETS UP CSR'S,SETS UP VECTORS.
 026246  006337  026454          DZLEV:  ASL     DZPRT           ;BUILD PRIORITY IN THIS LOCATION
 026252  006337  026454                  ASL     DZPRT           ;USING ARITHMETIC SHIFTS, ROTATE
 026256  006337  026454                  ASL     DZPRT           ;       THE PRIORITY LEVEL PAST
 026262  006337  026454                  ASL     DZPRT           ;       THE BIT POSITIONS CORRE-
 026266  006337  026454                  ASL     DZPRT           ;       SPONDING TO THE CONDITION CODES
 026272  013737  026454  026456          MOV     DZPRT,LESS1     ;MOVE THIS TO LESS1
 026300  162737  000001  026456          SUB     #1,LESS1        ;CREATE THE NEXT LOWEST PRIORITY
 026306  042737  000037  026456          BIC     #37,LESS1       ;INSURE THAT THE  TNZVC BITS ARE CLEAR
 026314  013700  002072                  MOV     DZRIV,R0        ;PLACE THE BASE VECTOR ADDRESS IN R0
 026320  062700  000002                  ADD     #2,R0           ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
 026324  010037  002074                  MOV     R0,DZRIS        ;STORE IT HERE
 026330  062700  000002                  ADD     #2,R0           ;CALCULATE THE TRANSMITTER INTERRUPT VECTOR
 026334  010037  002076                  MOV     R0,DZTIV        ;STORE IT HERE
 026340  062700  000002                  ADD     #2,R0           ;CALCULATE THE TRANSMITTER VECTOR STATUS ADDRESS
 026344  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
 026350  013700  001310                  MOV     $BASE,R0        ;COPY THE ADDRESS BEING LOADED
 026354  010037  002042                  MOV     R0,DZCSR        ;XXX0
 026360  005200                          INC     R0
 026362  010037  002044                  MOV     R0,HDZCSR       ;XXX1
 026366  005200                          INC     R0
 026370  010037  002046                  MOV     R0,DZRBUF       ;XXX2
 026374  010037  002052                  MOV     R0,DZLPR        ;XXX2
 026400  005200                          INC     R0
 026402  010037  002050                  MOV     R0,HDZRBUF      ;XXX3
 026406  010037  002054                  MOV     R0,HDZLPR       ;XXX3
 026412  005200                          INC     R0
 026414  010037  002056                  MOV     R0,DZTCR        ;XXX4
 026420  005200                          INC     R0
 026422  010037  002060                  MOV     R0,HDZTCR       ;XXX5
 026426  005200                          INC     R0
 026430  010037  002062                  MOV     R0,DZMSR        ;XXX6
 026434  010037  002066                  MOV     R0,DZTDR        ;XXX6
 026440  005200                          INC     R0
 026442  010037  002064                  MOV     R0,HDZMSR       ;XXX7
 026446  010037  002070                  MOV     R0,HDZTDR       ;XXX7
 026452  000207                          RTS     PC
 026454  000240                  DZPRT:  PR5
 026456  000200                  LESS1:  PR4     ;LEVEL TO ALLOW INTERRUPTS
```

```
                                          ;ERROR ERROR TABLE
026460  000000                   .ERRTAB:       0          ;ERROR 0
026462  000000                                  0
026464  000000                                  0

026466  026700                                  EM1        ;ERROR
026470  030150                                  DH1
026472  030346                                  DT1

026474  026753                                  EM2        ;ERROR 2
026476  030173                                  DH2
026500  030360                                  DT2

026502  027001                                  EM3        ;ERROR 3
026504  030226                                  DH3
026506  030376                                  DT3

026510  027040                                  EM4        ;ERROR 4
026512  030226                                  DH3
026514  030376                                  DT3

026516  027067                                  EM5        ;ERROR 5
026520  030240                                  DH4
026522  030404                                  DT4

026524  027116                                  EM6        ;ERROR 6
026526  030240                                  DH4
026530  030404                                  DT4

026532  027154                                  EM7        ;ERROR 7
026534  030226                                  DH3
026536  030376                                  DT3

026540  027215                                  EM8        ;ERROR 10
026542  030226                                  DH3
026544  030376                                  DT3

026546  027257                                  EM9        ;ERROR 11
026550  030226                                  DH3
026552  030376                                  DT3

026554  027315                                  EM10       ;ERROR 12
026556  030226                                  DH3
026560  030376                                  DT3

026562  027354                                  EM13       ;ERROR 13
026564  030226                                  DH3
026566  030376                                  DT3

026570  027405                                  EM14       ;ERROR 14
026572  030226                                  DH3
026574  030376                                  DT3

026576  027437                                  EM15       ;ERROR 15
026600  000000                                  0
026602  000000                                  0
```

```
4449
4450  026604  027501                                  EM16
4451  026606  030226                                  DH3
4452  026610  030376                                  DT3
4453
4454  026612  027552                                  EM17      ;ERROR 17
4455  026614  030226                                  DH3
4456  026616  030376                                  DT3
4457
4458  026620  027610                                  EM20
4459  026622  030226                                  DH3
4460  026624  030376                                  DT3
4461
4462  026626  027651                                  EM21      ;ERROR 21
4463  026630  030267                                  DH5
4464  026632  030422                                  DT5
4465
4466  026634  027701                                  EM22      ;ERROR 22
4467  026636  030240                                  DH4
4468  026640  030404                                  DT4
4469
4470  026642  027743                                  EM23      ;ERROR 23
4471  026644  030226                                  DH3
4472  026646  030376                                  DT3
4473
4474  026650  027773                                  EM24
4475  026652  030226                                  DH3
4476  026654  030376                                  DT3
4477
4478  026656  030021                                  EM25
4479  026660  030226                                  DH3
4480  026662  030376                                  DT3
4481
4482  026664  030051                                  EM26
4483  026666  030226                                  DH3
4484  026670  030376                                  DT3
4485
4486  026672  030100                                  EM27
4487  026674  030226                                  DH3
4488  026676  030376                                  DT3
```

```
                                                  ;ERROR MESSAGES
      026700  047200  020117  046123  EM1:        .ASCIZ  <200>/NO SLAVE SYNC RESPONSE FROM DZ11 REGISTER/
      026753     200  042522  044507  EM2:        .ASCIZ  <200>?REGISTER R/W FAILURE?
      027001     200  051124  047101  EM3:        .ASCIZ  <200>/TRANSMIT READY (TRDY) NOT SET/
      027040  051200  041505  044505  EM4:        .ASCIZ  <200>/RECEIVER DONE NOT SET/
      027067     200  040504  040524  EM5:        .ASCIZ  <200>/DATA COMPARISON ERROR/
      027116  042200  030532  020061  EM6:        .ASCIZ  <200>/DZ11 *RECEIVER BUFFER* ERROR/
      027154  052200  040522  051516  EM7:        .ASCIZ  <200>/TRANSMITTER FAILED TO INTERRUPT/
      027215     200  047125  054105  EM8:        .ASCIZ  <200>/UNEXPECTED TRANSMITTER INTERRUPT/
      027257     200  042522  042503  EM9:        .ASCIZ  <200>/RECEIVER FAILED TO INTERRUPT/
      027315     200  047125  054105  EM10:       .ASCIZ  <200>/UNEXPECTED RECEIVER INTERRUPT/
      027354  051600  046111  020117  EM13:       .ASCIZ  <200>/SILO ALARM SET TOO SOON/
      027405     200  044523  047514  EM14:       .ASCIZ  <200>/SILO ALARM FAILED TO SET/
      027437     200  041501  044524  EM15:       .ASCIZ  <200>/ACTION DETECTED ON INVALID LINE./
      027501     200  042522  042101  EM16:       .ASCIZ  <200>/READING DZRBUF DID NOT CLEAR SILO ALARM/
      027552  042200  052101  020101  EM17:       .ASCIZ  <200>/DATA VALID SHOULD NOT BE SET/
      027610  051200  041505  044505  EM20:       .ASCIZ  <200>/RECEIVER DONE SHOULD NOT BE SET/
      027651     200  042522  040514  EM21:       .ASCIZ  <200>/RELATIVE TIMING ERROR./
      027701     200  047515  042504  EM22:       .ASCIZ  <200>/MODEM SIGNAL ERROR ON CABLE TEST/
      027743     200  040504  040524  EM23:       .ASCIZ  <200>/DATA VALID IS NOT SET!/
      027773     200  040504  040524  EM24:       .ASCIZ  <200>/DATA OVERRUN IS SET!/
      030021     200  051106  046501  EM25:       .ASCIZ  <200>/FRAMING ERROR OCCURRED/
      030051     200  040520  044522  EM26:       .ASCIZ  <200>/PARITY ERROR OCCURRED/
      030100  043200  046125  020114  EM27:       .ASCIZ  <200>/FULL BINARY COUNT PATTERN NOT RECEIVED/

      030150  052200  040522  020120  DH1:        .ASCIZ  <200>/TRAP PC  DZ11 REG/
      030173     200  054105  042520  DH2:        .ASCIZ  <200>/EXPECTED  FOUND  REGISTER/
      030226  046200  047111  020105  DH3:        .ASCIZ  <200>/LINE NO./
      030240  042600  050130  041505  DH4:        .ASCIZ  <200>/EXPECTED  FOUND  LINE/
      030267     200  054124  046040  DH5:        .ASCIZ  <200>/TX LINE PREVIOUS TIME  ACTUAL TIME  PARAMETER/

                                          .EVEN
                                                  ;DATA TABLES FOR ERROR MESSAGES
      030346  000002                      DT1:    2
      030350     006     003                      .BYTE   6,3
      030352  001204                              $REG1
      030354     006     001                      .BYTE   6,1
      030356  001202                              $REG0

      030360  000003                      DT2:    3
      030362     006     004                      .BYTE   6,4
      030364  001214                              $REG5
      030366     006     001                      .BYTE   6,1
      030370  001212                              $REG4
      030372     006     001                      .BYTE   6,1
      030374  001202                              $REG0

      030376  000001                      DT3:    1
      030400     003     001                      .BYTE   3,1
      030402  001372                              SAVLIN

      030404  000003                      DT4:    3
      030406     006     004                      .BYTE   6,4
      030410  001214                              $REG5
      030412     006     001                      .BYTE   6,1
      030414  001212                              $REG4
      030416     003      001                  .BYTE   3,1
      030420  001372                           SAVLIN

      030422  000004                  DT5:     4
      030424     003      005                  .BYTE   3,5
      030426  001372                           SAVLIN
      030430     006      011                  .BYTE   6,9.
      030432  001214                           $REG5
      030434     006      007                  .BYTE   6,7
      030436  001220                           $TMP1
      030440     006      001                  .BYTE   6,1
      030442  001400                           REGIST
                                                       ;TABLE OF DELAY TIMES FOR INDIVIDUAL BAUD RATES
                                                       ;----------------------------------------------

      030444  002450                  DLYTBL: 2450                     ;TIME FOR   50 BAUD
      030446  001560                          1560                     ;TIME FOR   75 BAUD
      030450  001120                          1120                     ;TIME FOR  110 BAUD
      030452  000750                           750                     ;TIME FOR  134 BAUD
      030454  000660                           660                     ;TIME FOR  150 BAUD
      030456  000330                           330                     ;TIME FOR  300 BAUD
      030460  000150                           150                     ;TIME FOR  600 BAUD
      030462  000060                            60                     ;TIME FOR 1200 BAUD
      030464  000040                            40                     ;TIME FOR 1800 BAUD
      030466  000030                            30                     ;TIME FOR 2000 BAUD
      030470  000020                            20                     ;TIME FOR 2400 BAUD
      030472  000010                            10                     ;TIME FOR 3600 BAUD
      030474  000001                             1                     ;TIME FOR 4800 BUAD
      030476  000001                             1                     ;TIME FOR 7200 BAUD
      030500  000001                             1                     ;TIME FOR 9600 BAUD
      030502  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.

      030504                          CORMAX:
              000001                  .END
```

N09

```
MD-11-DZDZA-D   MACY11 27(1006)   02-MAR-77  08:23  PAGE 94                                                    PAGE:  0117
DZDZAD.P11    02-MAR-77 08:20           CROSS REFERENCE TABLE -- USER SYMBOLS

ABASE = 160010                 1#      407      448
ACDW1 = 000000               407       450
ACDW2 = 000000               407       451
ACPUOP= 000000               407       422
ACTIVE  001412               510#      784*    1832*    1833     1835*    1840     1841*    1842     1845*
ADDW0 = 000000               407       452
ADDW1 = 000000               407       453
ADDW10= 000000               407       462
ADDW11= 000000               407       463
ADDW12= 000000               407       464
ADDW13= 000000               407       465
ADDW14= 000000               407       466
ADDW15= 000000               407       467
ADDW2 = 000000               407       454
ADDW3 = 000000               407       455
ADDW4 = 000000               407       456
ADDW5 = 000000               407       457
ADDW6 = 000000               407       458
ADDW7 = 000000               407       459
ADDW8 = 000000               407       460
ADDW9 = 000000               407       461
ADEVCT= 000000               407       413
ADEVM = 000000               407       449
ADRCNT  006353              1401*     1438*    1448#
ADVANG= 104400               687#     2089     2343     2517     2531     2575     2584     2714     3239     3484     3529     3645     3743
AENV  = 000000               407       418
AENVM = 000000               407       419
AFATAL= 000000               407       410
AMADR1= 000000               407       435
AMADR2= 000000               407       439
AMADR3= 000000               407       442
AMADR4= 000000               407       445
AMAMS1= 000000               407       429
AMAMS2= 000000               407       437
AMAMS3= 000000               407       440
AMAMS4= 000000               407       443
AMSGAD= 000000               407       415
AMSGLG= 000000               407       416
AMSGTY= 000000               407       409
AMTYP1= 000000               407       430
AMTYP2= 000000               407       438
AMTYP3= 000000               407       441
AMTYP4= 000000               407       444
APASS = 000000               407       412
APRIOR= 000000               407
APTCSU= 000040              1252      1357#
APTENV= 000001              1245      1313     1355#    1630
APTSIZ= 000200              1354#
APTSPO= 000100              1247      1315     1356#
ASWREG= 000000               407       420
ATESTN= 000000               407       411
AUNIT = 000000               407       414
AUSWR = 000000               407       421
AUTO.S  011612              1018      1938#
AVECT = 000300                 1#
AVECT1= 000000               407       446
```

B10

```
MD-11-DZDZA-D   MACY11 27(1006)  02-MAR-77  08:23  PAGE 95                                                    PAGE:  0118
DZDZAD.P11    02-MAR-77 08:20           CROSS REFERENCE TABLE -- USER SYMBOLS

AVECT2= 000000         407      447
BAUD    023630        2969#    4289
BINWRD  006626        1524#
BIT0  = 000001         115#     192      194      196      198      235      254     2394     2533     3612     3710
BIT00 = 000001         105#     115
BIT01 = 000002         104#     114
BIT02 = 000004         103#     113
BIT03 = 000010         102#     112
BIT04 = 000020         101#     111
BIT05 = 000040         100#     110
BIT06 = 000100          99#     109
BIT07 = 000200          98#     108
BIT08 = 000400          97#     107
BIT09 = 001000          96#     106
BIT1  = 000002         114#     193      194      197      198      236      255     2394
BIT10 = 002000          95#     162      163      164      165      183      184      185      186      220      221      222      223
                       228      229      230      231      245      264      276     2394
BIT11 = 004000          94#     224      225      226      227      228      229      230      231      246      265      277     1185
                      2394     2460
BIT12 = 010000          93#     151      171      214      247      266      278     4155
BIT13 = 020000          92#     152      172      248      267      279     4059     4152
BIT14 = 040000          91#     153      173      249      268      280     1167     2031     4149
BIT15 = 100000          90#     154      174      250      269      281      885      888     1850     1981
BIT2  = 000004         113#     195      196      197      198      237      256     1071     2394
BIT3  = 000010         112#     146      201      203      205      207      238      257     3580     3582
BIT4  = 000020         111#     147      202      203      206      207      239      258     3580
BIT5  = 000040         110#     148      204      205      206      207      212      240      259     1950     1960     2031
BIT6  = 000100         109#     149      209      241      260
BIT7  = 000200         108#     150      210      242      261     1016     1660     1673     1694     1951     1958     2033
BIT8  = 000400         107#     159      161      163      165      180      182      184      186      217      219      221      223
                       225      227      229      231      243      262      274     2394
BIT9  = 001000         106#     160      161      164      165      181      182      185      186      218      219      222      223
                       226      227      230      231      244      263      275     2394
BPTVEC= 000014         122#
BRK0  = 000400         274#
BRK1  = 001000         275#
BRK2  = 002000         276#
BRK3  = 004000         277#
BRK4  = 010000         278#
BRK5  = 020000         279#
BRK6  = 040000         280#
BRK7  = 100000         281#
BRW     005232        1076     1204#
BUILD   023122        3816#    3855
BYTCNT  025572        4336#
CHRCNT  006624        1490*    1503*    1521#
CNVRT = 104412         707#    1098     1100     1103     1106     1609     1611     1613     1668
CONVRT= 104411         705#    1028     1627
CORMAX  030504        4490#
C00   = 000400         262#
C01   = 001000         263#
C02   = 002000         264#
C03   = 004000         265#
C04   = 010000         266#
C05   = 020000         267#
C06   = 040000         268#
```

```
C07    = 100000          269#
CR     = 000015           30#    1291    1301
CRLF   = 000200           31#    1262    1301
CSRMAP   011620         1941#
CYCLE    011070         1077     1128    1821#
DATABP   007220         1598*    1601    1625    1628#
DATAHD   007206         1597*    1621    1624#
DCLASM=  104417          717#    2655    2735    2821    2882    2951    3027    3141    3271    3354    3531    3605    3703
DCLR   = 000020          147#    1544    1545    2105    2431    2437    2440
DDISP  = 177570           37#     378
DELAY  = 104414          711#    1968    2412    2543    2549    2590    2596    2626    2676    2699    2762    2785    2857
                        2906     2929    2976    3001    3062    3070    3171    3187    3197    3303    3323    4021    4107
DEVADR   006350         1399*    1435    1446#
DEVICE=  104413          709#    1551    2151    2183    2215    2247    2279    2314    2360    2429    2455    2481    2519
                        2614     2635    2940    3012    3465    3977    4010    4031    4034    4074    4096    4116    4140
DH1      030150         4399     4490#
DH2      030173         4403     4490#
DH3      030226         4407     4411    4423    4427    4431    4435    4439    4443    4451    4455    4459    4471    4475
                        4479     4483    4487    4490#
DH4      030240         4415     4419    4467    4490#
DH5      030267         4463     4490#
DISPLA   001162          378#     800*   1197*   1640*
DISPRE   000174          340#     800
DLYCNT   006722          913*    1003*   1053    1557    1562#   3380
DLYTBL   030444         1003     4490#
DONFLG   001420          519#
DSWR   = 177570           36#     377
DTR0   = 000400          243#    2339
DTR1   = 001000          244#
DTR2   = 002000          245#
DTR3   = 004000          246#
DTR4   = 010000          247#
DTR5   = 020000          248#
DTR6   = 040000          249#
DTR7   = 100000          250#
DT1      030346         4400     4490#
DT2      030360         4404     4490#
DT3      030376         4408     4412    4424    4428    4432    4436    4440    4444    4452    4456    4460    4472    4476
                        4480     4484    4488    4490#
DT4      030404         4416     4420    4468    4490#
DT5      030422         4464     4490#
DVALID=  100000          174#    2460    2695    2781    2853    3214    3667    3765
DZACTV   001404          505#     926*   1043*   1044    1821    1827    1907*   1998*   2002*   2007    2026
DZCR0    001500          567#     839     842     927    2004
DZCR1    001514          574#     928
DZCR10   001640          623#
DZCR11   001654          630#
DZCR12   001670          637#
DZCR13   001704          644#
DZCR14   001720          651#
DZCR15   001734          658#
DZCR16   001750          665#
DZCR17   001764          672#
DZCR2    001530          581#
DZCR3    001544          588#
DZCR4    001560          595#
```

D10

```
MD-11-DZDZA-D   MACY11 27(1006)  02-MAR-77  08:23  PAGE 97                                                   PAGE:  0120
DZDZAD.P11    02-MAR-77 08:20             CROSS REFERENCE TABLE -- USER SYMBOLS

DZCR5     001574         602#
DZCR6     001610         609#
DZCR7     001624         616#
DZCSR     002042         725#   1132    1544*   1545    1552*   2064    2104    2136    2168    2200    2232    2264    2392
                        2430    2615    2666*   2670    2674    2697    2718    2752*   2756    2760    2783    2833*   2855
                        2899*   2901    2916*   2921*   2924    2968*   2970    2982*   2987*   2992*   2995    3010*   3041*
                        3055    3060    3068    3074    3079*   3080*   3119*   3157*   3164    3169    3179    3185    3219
                        3245*   3249*   3254*   3289*   3301    3311    3317    3321    3372*   3373*   3374*   3395    3420
                        3428    3564*   3566    3633*   3638    3677    3681*   3731*   3736    3775    3779*   3810*   3826
                        3868*   3950    3953    4016*   4019    4043*   4053    4071*   4079    4085*   4120*   4143    4178
                        4370*
DZLEV     026246        1854    4351#
DZLPR     002052         729#   2404*   2458*   2466*   2660*   2740*   2827*   2887*   2956*   3032*   3146*   3276*   3360*
                        3533*   3544*   3615*   3713*   3955*   4015*   4039*   4127*   4375*
DZLV0     001504         569#    864     867     885*    888*
DZLV1     001520         576#
DZLV10    001644         625#
DZLV11    001660         632#
DZLV12    001674         639#
DZLV13    001710         646#
DZLV14    001724         653#
DZLV15    001740         660#
DZLV16    001754         667#
DZLV17    001770         674#
DZLV2     001534         583#
DZLV3     001550         590#
DZLV4     001564         597#
DZLV5     001600         604#
DZLV6     001614         611#
DZLV7     001630         618#
DZMSR     002062         733#   2082    2482    2518    2578    3962*   4108    4384*
DZNUM     001410         507#    781     918*   1109    1946*   1987*   1988    1994    1996
DZPRT     026454        1848*   1849    1850*   2892*   2897    2898    2919    2920    2966    2967    2990    2991    3038
                        3040    3561    3562    3631    3632    3729    3730    4351*   4352*   4353*   4354*   4355*   4356
                        4390#
DZRBUF    002046         727#   2070    2456    2703    2750    2789    2834    2861    3052    3090    3215    3229    3294
                        3423    3652    3663    3750    3761    3829    3834    3953*   3954*   3955    3956    4056    4146
                        4374*
DZRIS     002074         739#   2897*   2919*   2966*   2990*   3038*   3120    3121*   3288*   3369*   3480    3481*   3561*
                        3631*   3729*   4041*   4122*   4361*
DZRIV     002072         738#   1135    1847*   2896*   2918*   2965*   2989*   3037*   3120*   3287*   3316*   3368*   3480*
                        3560*   3630*   3728*   3942    4040*   4121*   4359
DZTCR     002056         731#   2076    2295    2338    2405*   2621*   2669*   2708*   2755*   2796*   2837*   2868*   2893*
                        2962*   3051*   3112*   3163*   3233*   3248*   3299*   3337*   3375*   3415*   3563*   3634*   3732*
                        3808*   3956*   3957*   3958    3960    4012*   4078*   4086*   4129*   4380*
DZTDR     002066         735#   2484*   2491*   2680*   2720*   2766*   2839*   2922*   2993*   3066*   3175*   3257*   3307*
                        3319*   3412*   3797*   3960*   3961*   3962    4026*   4084*   4181*   4385*
DZTIS     002100         741#   2898*   2920*   2967*   2991*   3040*   3122    3123*   3371*   3482    3483*   3562*   3632*
                        3730*   4124*   4365*
DZTIV     002076         740#   2895*   2917*   2964*   2988*   3039*   3122*   3246*   3370*   3482*   3559*   3629*   3727*
                        4123*   4363*
DZV     = ****** U      1795
DZVC0     001502         568#    852     855    2047
DZVC1     001516         575#
DZVC10    001642         624#
DZVC11    001656         631#
```

```
DZVC12   001672             638#
DZVC13   001706             645#
DZVC14   001722             652#
DZVC15   001736             659#
DZVC16   001752             666#
DZVC17   001766             673#
DZVC2    001532             582#
DZVC3    001546             589#
DZVC4    001562             596#
DZVC5    001576             603#
DZVC6    001612             610#
DZVC7    001626             617#
DZ.END   002000             679#     824     1833     1842     1944
DZ.MAP   001500             510      565#     784      821     1023     1835     1845     1893     1941     1947     2018
EIAFLG   001414             515#    1849*    2341     2515     2576
EIGHT  = 000030             203#    3515     3518
EIGHTS=  000070             207#
EMTVEC=  000030             125#
EM1      026700            4398     4490#
EM10     027315            4434     4490#
EM13     027354            4438     4490#
EM14     027405            4442     4490#
EM15     027437            4446     4490#
EM16     027501            4450     4490#
EM17     027552            4454     4490#
EM2      026753            4402     4490#
EM20     027610            4458     4490#
EM21     027651            4462     4490#
EM22     027701            4466     4490#
EM23     027743            4470     4490#
EM24     027773            4474     4490#
EM25     030021            4478     4490#
EM26     030051            4482     4490#
EM27     030100            4486     4490#
EM3      027001            4406     4490#
EM4      027040            4410     4490#
EM5      027067            4414     4490#
EM6      027116            4418     4490#
EM7      027154            4422     4490#
EM8      027215            4426     4490#
EM9      027257            4430     4490#
ERRMSG   007174            1596*    1616     1619#
ERRVEC=  000004             118#    1172     1173*    1175*    1178*
ERTABO   007344            1611     1651#
EVEPAR=  000000             213#
EXITER   007300            1639     1642#
FINI     021164            3391     3479#
FIVE   = 000000             200#
FIVES  = 000040             204#
FRMERR=  020000             172#    2853     3434     3832
HALTS    007224            1581     1630#
HDRFLG   001416             517#     812*     822*     966*    1019     1021*
HDZCSR   002044             726#    3398     3788     4372*
HDZLPR   002054             730#    4378*
HDZMSR   002064             734#    4387*
HDZRBU   002050             728#    4377*
```

F10

```
MD-11-DZDZA-D   MACY11 27(1006)  02-MAR-77  08:23  PAGE 99                                                              PAGE:  0122
DZDZAD.P11    02-MAR-77 08:20            CROSS REFERENCE TABLE -- USER SYMBOLS

HDZTCR  002060          732#    2542*    2548*    2589*    2595*    3958*    3959*    4102*    4382*
HDZTDR  002070          736#    2838*    2865*    4388*
HILIM   006346          1398*   1426     1445#
HT    = 000011            28#   1260     1301
INBUF   010620          1368    1404     1763*    1764     1769     1774     1788#    4064*    4065*    4072     4203     4255*    4256
                        4260
INIFLG  001415           516#    803      808*     816      826*     952*    1014     1940*
INIT    020444          3358#   3363
INIT1   020470          3365#   3367
INSTER= 104404           695#   1420     1780     4222     4265
INSTR = 104403           693#    834      847      859      894      901      958      991     1048     1860     3937     3945     3965
                        3969    3979
INSTR2  006146          1375    1387#
INTRAN  025052          4123    4178#
INTREC  024672          4121    4142#
INTSVC  024236          4040    4053#
IOTVEC= 000020           123#   2048*
LAST    025564          3927*   4190*    4333#
LESS1   026456          2961*   2983*    3376*    4042*    4128*    4356*    4357*    4358*    4391#
LF    = 000012            29#   1295     1301
LIMITS  006274          1414    1426#
LINE    001364           497#   1851*    2526     2580     2619     2658     2667     2738     2753     2825     2835     2885     2893
                        2954    2962     3030     3043     3144     3158     3274     3291     3355     3358     3375     3408     3442
                        3469    3516     3610     3634     3708     3732
LINESP  025574          3974    4284     4337#
LINEX   023646          3977#   4049     4136
LINE0   001506           570#    910*     963      973      999
LINE1   001522           577#
LINE10  001646           626#
LINE11  001662           633#
LINE12  001676           640#
LINE13  001712           647#
LINE14  001726           654#
LINE15  001742           661#
LINE16  001756           668#
LINE17  001772           675#
LINE2   001536           584#
LINE3   001552           591#
LINE4   001566           598#
LINE5   001602           605#
LINE6   001616           612#
LINE7   001632           619#
LOBITS  006352          1400*   1430     1447#
LOCK    001362           492#   1212     1214     1568*    1605     2063*    2069*    2075*    2081*    2297*    2326*    2340*    2379*
                        2514*   2572*    2654*    2734*    2802*    2818*    2871*    3140*    3270*    3343*    3510*    4101*
LOCKUP  025562          3991*   4137*    4142*    4332#
LOLIM   006344          1397*   1428     1444#
LP0   = 000000           191#
LP1   = 000001           192#
LP2   = 000002           193#
LP3   = 000003           194#
LP4   = 000004           195#
LP5   = 000005           196#
LP6   = 000006           197#
LP7   = 000007           198#
MAINT = 000010           146#   1202     1772     2137     2432
```

```
MD-11-DZDZA-D   MACY11 27(1006)  02-MAR-77  08:23  PAGE 100                                     PAGE:  0123
DZDZAD.P11    02-MAR-77 08:20           CROSS REFERENCE TABLE -- USER SYMBOLS

MANT0    001512             572#     897     971    2005
MANT1    001526             579#
MANT10   001652             628#
MANT11   001666             635#
MANT12   001702             642#
MANT13   001716             649#
MANT14   001732             656#
MANT15   001746             663#
MANT16   001762             670#
MANT17   001776             677#
MANT2    001542             586#
MANT3    001556             593#
MANT4    001572             600#
MANT5    001606             607#
MANT6    001622             614#
MANT7    001636             621#
MASTEK   010315            1607    1751#
MBADLN   010424             980    1751#
MCABLE   026113            3998    4346#
MCHAR    025760            4044    4346#
MCSRX    010245            1097    1612    1751#
MDATA    010724            1501    1511    1792#
MEPASS   010063            1096    1751#
MERRPC   010372            1610    1751#
MERRX    010272            1105    1751#
MERR2    010123            1751#   1823    2009
MERR3    010172            1040    1751#
MINVAL   025704            4288    4346#
MLINE    025732            3980    4346#
MLOCK    010216            1073    1751#
MNEW     010320            1035    1751#
MNTFLG   001417             518#   1199*   1202*   1552    1767*   1772*   1777*
MODE     001370             499#   1200    1853*   2522    2573    2682    2768    2819    2841    3094    3202    3534    3603
                           3701
MPASS    025670            4189    4346#
MPASSX   010261            1102    1751#
MPFAIL   010020            1744    1751#
MQUICK   026130            4018    4346#
MR       010107            1078    1751#
MREGAD   025634            3946    4346#
MSENAB=  000040             148#   2169    2406    2407    2432    2616    2622    2666    2752    2833    2899    2921    2968
                           2992    3041    3157    3245    3254    3289    3374    3564    3633    3731    3868    4016    4043
                           4071    4085    4120
MSPEED   025742            3970    4346#
MTERM    026066            4004    4346#
MTITLE   001000             349#    807
MTSTN    010303            1608    1751#   1861
MVECTO   025612            3938    4346#
MVECX    010253            1099    1751#
MWHICH   026020            3966    4346#
NEXT     001360             491#   1567    1647    2060*   2103*   2135*   2167*   2199*   2231*   2263*   2294*   2337*   2391*
                           2428*   2454*   2480*   2513*   2524    2571*   2613*   2653*   2733*   2817*   2881*   2950*   3026*
                           3139*   3269*   3353*   3509*   3602*   3700*   4033*   4095*
NOLIST=  ****** U             1
NUMLIN   025600            4013    4035    4118    4274*   4340#
NUMTCR   025602            4012    4078    4102    4104    4106    4129    4276*   4283*   4342#
```

```
ODDPAR= 000200        210#    2823    3607    3614    3616    3705    3712    3714
OFFSET  023362       3511*    3526*   3575*   3857    3859    3870#
ONESTO= 000000        211#
OTHER   023610       3929     3963#
OUT     021112       3386     3464#
OVRRUN= 040000        173#    3226    3434    3832
PAR     001366        498#    1852*   2656    2736    2822    2883    2952    3028    3142    3272    3356    3606    3704
                     4013*    4014*   4015    4035*   4036*   4038*   4039    4117*   4118*   4119*   4127
PARAM = 104405        697#     836     849     861     903     960     993    1050    1862    3939    3947    3981
PARAM1  006214       1403#    1421
PARER = 010000        171#    2853    3434    3661    3667    3759    3765    3832
PARERR  006270       1406     1408    1410    1420#   1427    1429    1431
PARITY= 000100        209#    2823    3608    3706
PARMD = 104415        713#    3971
PAR0    001510        571#     911*    996    1000
PAR1    001524        578#
PAR10   001650        627#
PAR11   001664        634#
PAR12   001700        641#
PAR13   001714        648#
PAR14   001730        655#
PAR15   001744        662#
PAR16   001760        669#
PAR17   001774        676#    1010
PAR2    001540        585#
PAR3    001554        592#
PAR4    001570        599#
PAR5    001604        606#
PAR6    001620        613#
PAR7    001634        620#
PAWCH = 104416        715#    3967
PIRQ  = 177772         35#
PIRQVE= 000240        129#
POPR0 = 012600        138#
POP1SP= 005726        136#
POP2SP= 022626        140#     798    2044    2981    3009    3113    3118
PRIO    025604       4041     4122    4124    4343#
PR0   = 000000         52#
PR1   = 000040         53#
PR2   = 000100         54#
PR3   = 000140         55#
PR4   = 000200         56#    4391
PR5   = 000240         57#    4390
PR6   = 000300         58#
PR7   = 000340         59#     317     321     779*   1068*   2062    3048*   3288    3369    3371    3464*   3921*   3989*
                     4032*    4093*
PS    = 177776         32#      33
PSW   = 177776         33#
PUSHR0= 010046        137#
PUSH1S= 005746        135#
PUSH2S= 024646        139#
PWRVEC= 000024        124#    1713*   1714*   1723*   1729*   1741*   1742*
QUITS   024664       4140#    4172
RCVON = 010000        214#    3515    3518    3532    3543    4038    4119
RDATA   025570       4335#
RDONE = 000200        150#    2394    2670    2697    2756    2783    2855    2924    2995    3068
```

I10

```
MD-11-DZDZA-D   MACY11 27(1006)  02-MAR-77  08:23  PAGE 102                                                          PAGE:  0125
DZDZAD.P11     02-MAR-77 08:20           CROSS REFERENCE TABLE -- USER SYMBOLS

RECDAT  025606           4056*   4059    4063    4064    4066*   4067*   4068    4146*   4149    4152    4155    4160    4165*
                         4166*   4167    4344#
REGIST  001400            503#   2750*   2834*   3549*   3663*   3761*   4490
RESREG  007222           1626    1629#
RESTAR  011434           1746    1886#
RESVEC= 000010            119#
RES05 = 104410            703#   1629
RIE   = 000100            149#   2233    2432    2921    2992    3080    3119    3289    3372    3564    3566    3633    3638
                         3681    3731    3736    3779    3868    4043    4085    4120
RING0 = 000001            254#
RING1 = 000002            255#
RING2 = 000004            256#
RING3 = 000010            257#
RING4 = 000020            258#
RING5 = 000040            259#
RING6 = 000100            260#
RING7 = 000200            261#
RL0   = 000000            179#
RL1   = 000400            180#
RL2   = 001000            181#
RL3   = 001400            182#
RL4   = 002000            183#
RL5   = 002400            184#
RL6   = 003000            185#
RL7   = 003400            186#
RSTART  020434           3356#
RSTRT   025106           4192#
RUN     001406            506#    785*   1827    1830*   1831*   1838*   1839*
RXISR1  023150           3560    3826#
RXSVC   020734           3368    3420#
RXTCR   021212           3355*   3384    3458*   3485#
SAVLIN  001372            500#   2617*   2636*   2664*   2681    2709*   2710    2744*   2767    2797*   2798    2831*   2840
                         2867*   2891*   2960*   3036*   3045*   3105    3150*   3161    3201    3235*   3280*   3297    3339*
                         3402*   3427*   3445    3467*   3475*   3512*   3545*   3546*   3620*   3656*   3718*   3754*   3790
                         3792*   3794*   3845*   3846*   3865*   3866*   3984    4068    4167    4274    4275    4490
SAVNUM  001411            508#    781*    919*   1107*   1109*   1997*
SAVPC   001402            504#   1453*   1653
SAV05 = 104407            701#   1586
SCOP1 = 104401            689#   2093    2323    2366    2554    2601    2707    2721    2793    2866    3234    3255    3338
                         3574    4115
SERV.G  007360           1159    1574    1657#   1664    3573    4028    4048    4135
SET     025362           3987    4274#
SETAPT  011440            811    1893#
SETFLG= 104406            699#    896
SET.PS  010766            316    1795#
SET1    025414           4279    4282#
SEVEN = 000020            202#
SEVENS= 000060            206#
SILOAL= 020000            152#   2394    3179    3185    3219    3311    3321    3428
SILOEN= 010000            151#   2201    2432    3157    3245    3254    3289
SIX   = 000010            201#
SIXS  = 000050            205#
SNAP    020564           3379#   3459
SPACNT  006625           1522#
SPEED   025576           4014    4036    4117    4291*   4338#
SPIN    024632           4132#   4138    4175
```

J10

```
MD-11-DZDZA-D   MACY11 27(1006)  02-MAR-77  08:23  PAGE 103                                                 PAGE:  0126
DZDZAD.P11    02-MAR-77 08:20             CROSS REFERENCE TABLE -- USER SYMBOLS

STACK  = 001120              23#      778     1067     1648     3920     3990
STFLG    025560            3923*    3978*     3995    3997*     4001    4003*    4331#
STKLMT=  177774              34#
STOP     001462             538#     2747     3153     3283     3366     3521     3552     3626     3724
SVO5     006362            1457#
SWR      001160             377#      795     799*      814      818      914      948     1016     1033     1038     1043     1046     1071
                           1167      1185     1210     1217     1575     1580     1638     1643     1645     1657     1665    1666*    1676*
                           1681*    1682*    1683*    1690*     1721    1734*     1857    1926*     3928     3963     4029
SWREG    000176             341#      799
SW0    = 000001              87#
SW00   = 000001              77#       87      814     3928
SW01   = 000002              76#       86     1857     3963
SW02   = 000004              75#       85
SW03   = 000010              74#       84      914      948
SW04   = 000020              73#       83     1046
SW05   = 000040              72#       82
SW06   = 000100              71#       81     1033
SW07   = 000200              70#       80
SW08   = 000400              69#       79     1643
SW09   = 001000              68#       78     1210
SW1    = 000002              86#
SW10   = 002000              67#     1645
SW11   = 004000              66#
SW12   = 010000              65#     1217     1575
SW13   = 020000              64#     1580
SW14   = 040000              63#
SW15   = 100000              62#
SW2    = 000004              85#
SW3    = 000010              84#
SW4    = 000020              83#
SW5    = 000040              82#
SW6    = 000100              81#
SW7    = 000200              80#
SW8    = 000400              79#
SW9    = 001000              78#
S110   = 001000             218#
S1200  = 003400             223#
S134   = 001400             219#
S150   = 002000             220#
S1800  = 004000             224#
S19200=  007400             231#
S2000  = 004400             225#
S2400  = 005000             226#
S300   = 002400             221#
S3600  = 005400             227#
S4800  = 006000             228#
S50    = 000000             216#     3515     3518
S600   = 003000             222#
S7200  = 006400             229#
S75    = 000400             217#
S9600  = 007000             230#
TABLE2   025456            4282     4297#
TBITVE=  000014             120#
TBUF     025610            4063*     4084    4345#
TCR0   = 000001             235#     2296     2405
TCR1   = 000002             236#
```

K10

```
MD-11-DZDZA-D   MACY11 27(1006)  02-MAR-77  08:23  PAGE 104                                                        PAGE:  0127
DZDZAD.P11    02-MAR-77 08:20           CROSS REFERENCE TABLE -- USER SYMBOLS

TCR2   = 000004             237#
TCR3   = 000010             238#
TCR4   = 000020             239#
TCR5   = 000040             240#
TCR6   = 000100             241#
TCR7   = 000200             242#
TDATA    025566            4334#
TD0      001422             522#    2745     2766     2780     2794*    2922     2993     3151     3175     3176*    3281     3307     3308*
                           3309     3319     3364     3412     3413*    3519     3550     3624     3722     3797     3798*    3800
TD1      001424             523#
TD2      001426             524#
TD3      001430             525#
TD4      001432             526#
TD5      001434             527#
TD6      001436             528#
TD7      001440             529#
TEIGHT   002140             764#
TEMP     010662            1790#
TEST1    023776            4000     4010#    4030
TEST2    024434            3994     4093#
TFIVE    002146             767#
TIE    = 040000             153#    2265     2432     2899     2916     2968     2982     2987     3079     3119     3249     3254     3373
                           3564     3633     3677     3731     3775     3810     4120
TKVEC  = 000060             127#
TLAST  = 022354            1880     4490#
TL0    = 000000             158#
TL1    = 000400             159#
TL2    = 001000             160#
TL3    = 001400             161#
TL4    = 002000             162#
TL5    = 002400             163#
TL6    = 003000             164#
TL7    = 003400             165#
TMTBL    002102             748#    3858*    3862     3864
TPVEC  = 000064             128#
TRAPVE=  000034             126#
TRDY   = 100000             154#    2399     2403     2616     2624     2674     2718     2760     2901     2970     3169     3301     3317
TRTVEC=  000014             121#
TR0      001442             530#    3447     3449     3455*    3456     3471     3666     3669     3671     3675*    3764     3767     3769
                           3773*    3840     3842     3848*    3850
TR1      001444             531#
TR2      001446             532#
TR3      001450             533#
TR4      001452             534#
TR5      001454             535#
TR6      001456             536#
TR7      001460             537#
TSEVEN   002142             765#
TSIX     002144             766#
TST1     012374            1161     1868     1884     2058#
TST10    013312            2263     2292#
TST11    013450            2294     2335#
TST12    013632            2337     2389#
TST13    013750            2391     2426#
TST14    014034            2428     2452#
TST15    014124            2454     2478#
```

L10

```
MD-11-DZDZA-D   MACY11 27(1006)  02-MAR-77  08:23  PAGE 105                                                      PAGE:  0128
DZDZAD.P11    02-MAR-77 08:20          CROSS REFERENCE TABLE -- USER SYMBOLS

TST16    014210          2480    2511#
TST17    014404          2513    2569#
TST2     012564          2060    2101#
TST20    014542          2571    2611#
TST21    014666          2613    2651#
TST22    015202          2653    2731#
TST23    015530          2733    2815#
TST24    016006          2817    2879#
TST25    016316          2881    2948#
TST26    016644          2950    3024#
TST27    017276          3026    3137#
TST3     012650          2103    2133#
TST30    020024          3139    3267#
TST31    020406          3269    3351#
TST32    021214          3353    3506#
TST33    021724          3509    3600#
TST34    022354          3602    3698#   4490
TST35 =  ****** U        3700
TST4     012742          2135    2165#
TST5     013034          2167    2197#
TST6     013126          2199    2229#
TST7     013220          2231    2261#
TTST     005010          1074*   1076*   1163#
TWOSTO=  000040           212#   3515    3518
TXSVC    020636          3370    3395#
TYPDAT   007210          1602    1622    1625#
TYPE  =  104402           691#    807     876     979     980    1022    1035    1040    1073    1078    1096    1097    1099
                         1102    1105    1265    1366    1385    1478    1511    1603    1604    1607    1608    1610    1612
                         1614    1618    1623    1667    1669    1697    1743    1823    1859    1877    1882    2009    3998
                         4004    4044    4188    4288
TYPMSG   007100          1600    1603#
T110     002106           751#
T1200    002120           756#
T134     002110           752#
T150     002112           753#
T1800    002122           757#
T2000    002124           758#
T2400    002126           759#
T300     002114           754#
T3600    002130           760#
T4800    002132           761#
T50      002102           749#
T600     002116           755#
T7200    002134           762#
T75      002104           750#
T9600    002136           763#
VECMAP   012202          2008    2016#
VEC1     023426          3928#
VEC2     023436          3930#
WCHFLG   025556          3968    3992    4330#
WRDCNT   006622          1486*   1512*   1520#
WTBS.F   007176          1617    1620#
XBEGIN   023676          3964    3989#   4192
XBX      006766          1576    1578    1580#
XCSR     004742          1098    1130#   1613
XEOP     025072          4033    4075    4095    4141    4188#
```

```
XERR      004764          1106    1139#
XHEAD     010377          1022    1751#
XMTCNT    001376           502#   3558*   3623*   3721*   3800
XMTLIN    001374           501#   3513*   3523*   3525    3548
XMTSRV    023004          3559    3629    3727    3788#
XPASS     004756          1103    1136#
XSTART    023364           345    3920#   3922
XSTATQ    010466          1029    1751#
XTCR0     025370          4275#
XTCR1     025404          4278#   4281
XTSTN     007352          1609    1654#
XVEC      004750          1100    1133#
XX      = 160210           566#    573#    580#    587#    594#    601#    608#    615#    622#    629#    636#    643#    650#
                           657#    664#    671#    678#
YY      = 000500           566#    573#    580#    587#    594#    601#    608#    615#    622#    629#    636#    643#    650#
                           657#    664#    671#    678#
ZZ      = 000020           566#    573#    580#    587#    594#    601#    608#    615#    622#    629#    636#    643#    650#
                           657#    664#    671#    678#
$APTHD    001462           548     554#
$ASTAT= ****** U          1335    1350
$ATYC     005606          1306    1308#
$ATY1     005562          1304#
$ATY3     005570          1250    1305#
$ATY4     005600          1307#   1633
$AUTOB    001154           374#
$BASE     001310           448#    842*   1846*   1894    2004*   4369
$BDADR    001142           369#
$BDDAT    001146           371#
$CDW1     001314           450#   1918    2005*
$CDW2     001316           451#
$CHARC    005556          1267*   1277*   1284    1293*   1298#
$CMTAG    001120           357#
$CM1    = 000006           389#    390#    391#    392#    393#    394#    395#
$CM2    = 000014           389#    390#    391#    392#    393#    394#    395#
$CM3    = 000006           387#    389
$CM4    = 000004           395#    396#    397#    398#    399#
$CPUOP    001262           422#
$CRAP   = 177777             1#   2052#   2056#   2095#   2099#   2125#   2131#   2157#   2163#   2189#   2195#   2221#   2227#
                          2253#   2259#   2285#   2290#   2327#   2333#   2380#   2387#   2417#   2424#   2446#   2450#   2472#
                          2476#   2498#   2508#   2557#   2567#   2604#   2609#   2640#   2649#   2724#   2729#   2805#   2813#
                          2872#   2877#   2941#   2946#   3014#   3022#   3124#   3135#   3259#   3265#   3344#   3349#   3488#
                          3504#   3587#   3598#   3686#   3696#
$CRLF     001231           401#   1266    1301    1478    1603    1604    1614    1697    1859    1877
$DDW0     001320           452#    929    1905    1942
$DDW1     001322           453#
$DDW10    001344           462#
$DDW11    001346           463#
$DDW12    001350           464#
$DDW13    001352           465#
$DDW14    001354           466#
$DDW15    001356           467#
$DDW2     001324           454#
$DDW3     001326           455#
$DDW4     001330           456#
$DDW5     001332           457#
$DDW6     001334           458#
```

```
MD-11-DZDZA-D   MACY11 27(1006)  02-MAR-77  08:23  PAGE 107                                                     PAGE:  0130
DZDZAD.P11    02-MAR-77 08:20           CROSS REFERENCE TABLE -- USER SYMBOLS

$DDW7    001336          459#
$DDW8    001340          460#
$DDW9    001342          461#
$DEVCT   001244          413#
$DEVM    001312          449#     917*     923*     925      926      1038     1906     2007*
$DOAGN   004736         1108      1115     1120     1126#
$E     = 000036            1#     2060     2061#    2103     2104#    2135     2136#    2167     2168#    2199     2200#    2231     2232#
                        2263      2264#    2294     2295#    2337     2338#    2391     2392#    2428     2429#    2454     2455#    2480
                        2481#     2513     2515#    2571     2573#    2613     2614#    2653     2655#    2733     2735#    2817     2818#
                        2881      2882#    2950     2951#    3026     3027#    3139     3140#    3269     3270#    3353     3354#    3509
                        3510#     3602     3603#    3700     3701#
$ENDAD   004726          334      805      1122#    1636
$ENDCT   004712         1117#
$ENV     001254          418#     1245     1313     1337     1630
$ENVM    001255          419#     809      1247     1252     1315
$EOP     004562         1092#     3700
$EOPCT   004704         1114#     1118
$ERFLG   001123          360#     783*     1095*    1151     1181     1183*    1206     1585*    1599     1615*    3926*    4191*
$ERMAX   001135          366#     1206
$ERROR   006736          324      1574#
$ERRPC   001136          367#     787*     1094*    1160*    1582     1584*
$ERRTB   001360          486#
$ERTTL   001132          364#     786*     1141     1642*    3925*
$ETABL   001254          417#
$ETEND   001360          470#     560
$FATAL   001236          410#     1341*
$FFLG    006026         1304*     1307*    1335     1344*    1352#
$FILLC   001176          385#     1270     1301
$FILLS   001175          384#     1301
$GDADR   001140          368#
$GDDAT   001144          370#
$GET42   004716         1119#
$HD    = 000001           10       11
$HIBTS   001462          555#
$ICNT    001124          361#     1189*    1190     1192*    1205
$ILLUP   010012         1713      1729     1748#
$INTAG   001155          375#
$ITEMB   001134          365#     1590*    1632
$LF      001232          402#     1301
$LFLG    006025         1345*     1351#
$LPADR   001126          362#     789*     1077*    1079     1196*    1198     1205     1647*    1649     1875*    1876*    1884*    1886
                        2524*     2525     3922*    3994*    4000*    4005
$LPERR   001130          363#
$MADR1   001266          435#
$MADR2   001272          439#
$MADR3   001276          442#
$MADR4   001302          445#
$MAIL    001234          408#     556      560      1195     1245
$MAMS1   001264          429#
$MAMS2   001270          437#
$MAMS3   001274          440#
$MAMS4   001300          443#
$MBADR   001464          556#
$MFLG    006024         1305*     1311     1346*    1350#
$MSGAD   001250          415#     1321*    1324
$MSGLG   001252          416#     1326*
```

B11

```
MD-11-DZDZA-D   MACY11 27(1006)  02-MAR-77  08:23  PAGE 108                                                          PAGE:  0131
DZDZAD.P11    02-MAR-77 08:20             CROSS REFERENCE TABLE -- USER SYMBOLS

$MSGTY   001234          409#    1319    1327*   1339    1343*
$MTYP1   001265          430#
$MTYP2   001271          438#
$MTYP3   001275          441#
$MTYP4   001301          444#
$MXCNT   005234         1193     1205#   1826
$N     = 000034            1#    2052    2056    2061#   2095    2099    2104#   2125    2131    2136#   2157    2163    2168#
                        2189     2195    2200#   2221    2227    2232#   2253    2259    2264#   2285    2290    2295#   2327
                        2333     2338#   2380    2387    2392#   2417    2424    2429#   2446    2450    2455#   2472    2476
                        2481#    2498    2509    2515#   2557    2567    2573#   2604    2609    2614#   2640    2649    2655#
                        2724     2729    2735#   2805    2813    2818#   2872    2877    2882#   2941    2946    2951#   3014
                        3022     3027#   3124    3135    3140#   3259    3265    3270#   3344    3349    3354#   3488    3504
                        3510#    3587    3598    3603#   3686    3696    3701#   4490#
$NULL    001174          383#    1272    1301
$NWTST=  000000         2057#    2100#   2132#   2164#   2196#   2228#   2260#   2291#   2334#   2388#   2425#   2451#   2477#
                        2510#    2568#   2610#   2650#   2730#   2814#   2878#   2947#   3023#   3136#   3266#   3350#   3505#
                        3599#    3697#
$OVER    005176         1165     1168    1179    1191    1197#
$PASS    001242          412#     782*   1101*   1104*   1111*   1112*   1129    1138    1187    1206    3924*
$PASTM   001470          558#
$PWRAD   010006         1746#
$PWRDN   007646          322      780    1713#   1741
$PWRMG   010002         1744#
$PWRUP   007720         1723     1729#
$QUES    001230          400#     979    1301    1385    1882
$REGAD   001200          387#
$REG0    001202          389#    1462*   1467    4490
$REG1    001204          390#    1461*   1468    4490
$REG2    001206          391#    1460*   1469
$REG3    001210          392#    1459*   1470
$REG4    001212          393#    1458*   1471    4490
$REG5    001214          394#    1457*   1472    4490
$RTNAD   004740         1128#
$SAVR6   010016         1722*    1730    1731*   1732*   1750#
$SCOPE   004772         1158#
$SETUP=  000000         1110     1159
$SVLAD   005160         1176     1194#
$SVPC  = 000040          332#     337
$SWR   = 164000            1#      10     399     400    1089    1110    1121    1127    1129    1152    1153    1154    1155
                        1167     1179    1181    1182    1183    1184    1185    1197    1205    1747    2059    2102    2134
                        2166     2198    2230    2262    2293    2336    2390    2427    2453    2479    2512    2570    2612
                        2652     2732    2816    2880    2949    3025    3138    3268    3352    3507    3601    3699
$SWREG   001256          420#    1926
$SWRMK=  000000         1155
$TESTN   001240          411#    1195*
$TIMES   001226          399#    1110*   1184*   1190    1193*   1205    1826*   3508*
$TKB     001166          380#     880    1166    1372    1378    1659    1672    1693    4047    4134
$TKS     001164          379#     878    1164    1370    1663    1670    1691    4045    4132
$TMP0    001216          395#    3247*   3252*   3514*   3516    3527*   3563
$TMP1    001220          396#     877*    906     918     919     920*   1024*   1753    3556*   3568*   3818*   3858    3862
                        4490
$TMP2    001222          397#     925*    931*    932*   1025*   1026    1755    3816*   3819*
$TMP3    001224          398#    3557*   3570*   3817*
$TN    = 000035           10#    2057    2059#   2100    2102#   2132    2134#   2164    2166#   2196    2198#   2228    2230#
                        2260     2262#   2291    2293#   2334    2336#   2388    2390#   2425    2427#   2451    2453#   2477
                        2479#    2510    2512#   2568    2570#   2610    2612#   2650    2652#   2730    2732#   2814    2816#
```

```
MD-11-DZDZA-D   MACY11 27(1006)  02-MAR-77  08:23  PAGE 109                                              PAGE:  0132
DZDZAD.P11    02-MAR-77 08:20          CROSS REFERENCE TABLE -- USER SYMBOLS

                              2878    2880#   2947    2949#   3023    3025#   3136    3138#   3266    3268#   3350    3352#   3505
                              3507#   3599    3601#   3697    3699#
$TPB      001172              382#    884*    1290*   1301    1378*   1579*   1679*
$TPFLG    001177              386#    1239    1301
$TPS      001170              381#    1288    1301    1376    1577    1677
$TSTM     001466              557#
$TSTNM    001122              359#    788*    1151    1194*   1195    1197    1206    1656    1865    1873    2059*   2102*   2134*
                              2166*   2198*   2230*   2262*   2293*   2336*   2390*   2427*   2453*   2479*   2512*   2570*   2612*
                              2652*   2732*   2816*   2880*   2949*   3025*   3138*   3268*   3352*   3507*   3601*   3699*   4011*
                              4094*
$TYPE     005300              1239#   1332
$TYPEC    005512              1269    1276    1283    1288#   1289
$TYPEX    005560              1294    1296    1299#
$UNIT     001246              414#
$UNITM    001472              559#
$USWR     001260              421#
$VECT1    001304              446#    855*    867*    868*    869*    870*    871*    872*    1895    1898    2047*
$VECT2    001306              447#
$XTSTR    005036              1162    1170#
$Y      = 000020              680#    687     689#    691#    693#    695#    697#    699#    701#    703#    705#    707#    709#
                              711#    713#    715#    717#    719#
$$GET4= 000000                1121#
$4OCAT= ****** U              1167
.       = 030504              311#    312     315#    319#    332     333#    335#    337#    339#    342#    344#    348#    356#
                              403     505#    507#    520#    544     545#    547#    549#    564#    567#    568#    569#    570#
                              571#    572#    574#    575#    576#    577#    578#    579#    581#    582#    583#    584#    585#
                              586#    588#    589#    590#    591#    592#    593#    595#    596#    597#    598#    599#    600#
                              602#    603#    604#    605#    606#    607#    609#    610#    611#    612#    613#    614#    616#
                              617#    618#    619#    620#    621#    623#    624#    625#    626#    627#    628#    630#    631#
                              632#    633#    634#    635#    637#    638#    639#    640#    641#    642#    644#    645#    646#
                              647#    648#    649#    651#    652#    653#    654#    655#    656#    658#    659#    660#    661#
                              662#    663#    665#    666#    667#    668#    669#    670#    672#    673#    674#    675#    676#
                              677#    817     1013#   1129    1205    1206    1301    1353#   1671    1678    1692    1725    1749
                              1789#   1791#   1793#   1803    1825    1967    1970    2012    2036    2377    2748    3053    3056
                              3075    3106    3110    3154    3159    3165    3180    3199    3237    3284    3292    3295    3312
                              3318    3341    3383    3396    3409    3421    3429    3432    3435    3443    3457    3472    3522
                              3553    3627    3725    4054    4057    4060    4069    4144    4147    4150    4153    4156    4163
                              4168    4179    4346#
.ADVAN    006724              688     1567#
.BEGIN    004474              1067#
.CNVRT    006452              708     1479#
.CONVR    006446              706     1478#
.DCLAS    006672              718     1551#
.DELAY    006704              712     1555#
.DEVIC    006652              710     1543#
.ERRTA    026460              1595    4394#
.INSTE    006134              696     1383#
.INSTR    006030              694     1362#
.INST1    006050              1366#   1386
.MSG      006052              1364*   1367#
.PARAM    006154              698     1394#
.PARMD    025112              714     4195#
.PAWCH    025306              716     4254#   4266
.RESOS    006414              704     1467#
.SAVOS    006354              702     1453#
.SCOPE    004772              320     1159#   2048
```

```
.SCOP1  005236             690     1210#
.SETFL  010500             700     1762#    1781
.START  002150             343      776#     789
.TRPSR  006630             326     1532#
.TRPTA  002002             686#    1537
.TYPE   005262             692     1217#
.$ASTA= ****** U          1305     1308
.$X    = 001462            544#     549
```

```
MD-11-DZDZA-D   MACY11 27(1006)  02-MAR-77  08:23  PAGE 112                                          PAGE:  0134
DZDZAD.P11     02-MAR-77 08:20          CROSS REFERENCE TABLE -- MACRO NAMES

COMMEN      1#    130#
ENDCOM      1#    130#
ERROR      24#   2092    2110    2123    2142    2148    2155    2174    2180    2187    2206    2212    2219    2238    2244
         2251    2270    2276    2283    2303    2311    2321    2349    2357    2365    2378    2398    2402    2415    2436
         2444    2463    2470    2488    2495    2547    2553    2594    2600    2629    2633    2672    2679    2702    2706
         2758    2765    2788    2792    2860    2864    2909    2911    2914    2932    2934    2937    2974    2979    2985
         2999    3004    3006    3054    3057    3065    3073    3076    3084    3085    3107    3111    3116    3166    3174
         3181    3190    3218    3221    3232    3306    3313    3326    3331    3334    3389    3390    3397    3410    3422
         3430    3433    3436    3444    3454    3473    3572    3644    3659    3668    3674    3742    3757    3766    3772
         3793    3828    3831    3835    3847    3867    4024    4055    4058    4061    4070    4083    4111    4139    4145
         4148    4151    4154    4157    4164    4169    4180
ESCAPE      1#    130#
GETPRI      1#    130#
GETSWR      1#    130#
MULT        1#    130#
NEWTST      1#    130#   2057    2100    2132    2164    2196    2228    2260    2291    2334    2388    2425    2451    2477
         2510    2568    2610    2650    2730    2814    2878    2947    3023    3136    3266    3350    3505    3599    3697
PASEND      1#   1093
POP         1#    130#   1347    1348    1734    1735
PRGEND      1#   1080
PRGFRT      1#
PUSH        1#    130#   1308    1310    1331    1715    1721
REPORT      1#    130#
SC          1#   1159
SCOPE      25#   1093    2058    2101    2133    2165    2197    2229    2261    2292    2335    2389    2426    2452    2478
         2511    2569    2611    2651    2731    2815    2879    2948    3024    3137    3267    3351    3506    3600    3698
SC1         1#   1139
SETPRI      1#    130#
SETUP       1#    130#
SKIP        1#    130#
SLASH       1#    130#
SPACE     130#
STARS       1#    130#    330     352     403     406     541     543     550    1087    1148    1224    1303    1711    1727
         2057    2100    2132    2164    2196    2228    2260    2291    2334    2388    2425    2451    2477    2510    2568
         2610    2650    2730    2814    2878    2947    3023    3136    3266    3350    3505    3599    3697
SWRSU       1#    130#
TYPBIN      1#    130#
TYPDEC      1#    130#
TYPNAM      1#    130#
TYPNUM      1#    130#
TYPOCS      1#    130#
TYPOCT      1#    130#
TYPTXT      1#    130#
$APTYP      1#
$BUFFE      1#   1785
$CYCLE      1#   1812
$EOP        1#   1080
$GETFL      1#    894
$GETPA      1#    833     846     859     901     957     990    1048    1860
$HEADE      1#     11
$INTSE      1#   2894    2917    2963    2988    3629    3727
$JUNK       1#    566     573     580     587     594     601     608     615     622     629     636     643     650     657
          664     671
$LINEU      1#   2655    2735    2882    2951    3027    3141    3271
$LVLTS      1#   2872    2941
$MRESE      1#   2655    2735    2821    2882    2939    2951    3011    3027    3141    3271    3354    3530    3605    3703
```

F11

```
MD-11-DZDZA-D   MACY11 27(1006)  02-MAR-77  08:23  PAGE 113                                                           PAGE:  0135
DZDZAD.P11     02-MAR-77 08:20          CROSS REFERENCE TABLE -- MACRO NAMES

$MRR          1#   2380
$MRRW         1#   2125    2157    2189    2221    2253
$MRWD         1#   2446    2472
$MSG          1#   1751
$PARTS        1#   3587    3686
$SCOPE        1#   1142
$SETFL        1#   1757
$STAG         1#   2682    2768    2841    3094    3202
$STAGF        1#
$TCR          1#   2285    2327
$TLINE        1#   2604
$TRPDE        1#    687     689     691     693     695     697     699     701     703     705     707     709     711     713
            715     717
$TSTN         1#   2056    2099    2131    2163    2195    2227    2259    2290    2333    2387    2424    2450    2476    2509
           2567    2609    2649    2729    2813    2877    2946    3022    3135    3265    3349    3504    3598    3696
$UNIBU        1#   2052
$VARIA        1#    347
$XZ           1#   2052    2056    2095    2099    2125    2131    2157    2163    2189    2195    2221    2227    2253    2259
           2285    2290    2327    2333    2380    2387    2417    2424    2446    2450    2472    2476    2498    2508    2557
           2567    2604    2609    2640    2649    2724    2729    2805    2813    2872    2877    2941    2946    3014    3022
           3124    3135    3259    3265    3344    3349    3488    3504    3587    3598    3686    3696
$$CMRE      350#    389     390     391     392     393     394
$$CMTM      350#    395     396     397     398
$$ESCA        1#    130#
$$NEWT        1#    130#   2057    2100    2132    2164    2196    2228    2260    2291    2334    2388    2425    2451    2477
           2510    2568    2610    2650    2730    2814    2878    2947    3023    3136    3266    3350    3505    3599    3697
$$SKIP        1#    130#
.EQUAT        1#     20
.HEADE        1#
.KT11         1#
.SETUP        1#
.SWRHI        1#
.$ACT1        1#    328
.$APTB        1#    404#
.$APTH        1#    539
.$APTY        1#   1301
.$ASTA        1#
.$CATC        1#
.$CMTA        1#    350
.$DB2D        1#
.$DB20        1#
.$DIV         1#
.$EOP         1#   1085
.$ERRO        1#
.$ERRT        1#
.$MULT        1#
.$POWE        1#   1709
.$RAND        1#
.$RDDE        1#
.$RDOC        1#
.$READ        1#
.$R2AZ        1#
.$SAVE        1#
.$SB2D        1#
.$SB20        1#
 $SCOP        1#   1146
```

```
.$SIZE       1#
.$SUPR       1#
.$TRAP       1#
.$TYPB       1#
.$TYPD       1#
.$TYPE       1#    1221
.$TYPO       1#
.$40CA       1#
.1170        1#

. ABS.  030504      000

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DZDZAD,DZDZAD/CRF/SOL/NL:TOC+SYSMAC.SML[400,1066],DZDZAD.P11[400,1523]
 RUN-TIME: 17 23 1 SECONDS
 RUN-TIME RATIO: 421/43=9.7
 CORE USED:  51K  (101 PAGES)
```