DZV-11

(4) LINE ASYNCHRONOUS MUX

MD-11-DVDZA-A

TESTS, PART 1 OF 2

EP-DVDZA-A-DL-A

COPYRIGHT © 1977

FICHE 1 OF 1

OCT 1977

digital

MADE IN USA

```
EOF1DVDVCASEQ           00010000      770920            PDP10 411        HDR1DVDZAASEQ               00010000      770920
     PDP10 PAGE:  0001
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DVDZA-A-D
PRODUCT NAME:	DZV11 4 LINE ASYNC MUX TESTS PART 1 OF 2
DATE RELEASED:	APRIL 1977
MAINTAINER:	DIAGNOSTIC ENGINEERING

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

1. ABSTRACT

The function of the DZV11 diagnostics is to verify the option operates according to specifications. The diagnostics also verify that the DZV11 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 DZV11 device addresses and vectors only. All remaining parameters will 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 are three standalone diagnostics (DVDZA,DVDZB,and DVDZC) one system module for DEC X/11 (DZBA), and an overlay for ITEP (DVDZD).

DVDZA together with DVDZB will test all logical functions of the DZV11 interface module.
DVDZC is designed as a non-chainable standalone diagnostic providing the operator with direct control over the testing of all DZV11 EIA cables.

2. REQUIREMENTS

2.1 EQUIPMENT

An LSI11 CPU with minimum 4K of memory.
ASR 33 (or equivalent for console)

DZV11	INTERFACE MODULE
H329	Staggered turnaround connector.
H325	Cable turnaround connector.

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

2.2 STORAGE

Program will use all 4K of memory except where ABL and BOOTSTRAP LOADER reside. Location 1500 thru 1740 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 Starting the processor at the Absolute Loader starting address will load the diagnostic into memory.

4. STARTING PROCEEDURE

A. Set SWR to zero for 'AUTO SIZING' or set SW00=1 for user parameter input from console terminal. NOTE: loc. 000176 is used as a software Switch Register in all of the DZV11 diagnostics. (see Sec. 4.1) On the first startup of the diagnostic if SW07=1 and SW00=0 the program will assume that the status table has been already built from a previous DZV11 diagnostic run. NOTE: any DZV11 diagnostic will overlay the status table when loaded to preserve its contents and thus will not alter a previously built table.

B. Start the diagnostic at Loc. 200(8). The program will type Maindec and program names (if this was the first start up of the program) and also the following: (on the first program run or if parameters were changed)

```
'MAP OF DZV11 STATUS'
1500    160100
1502    000300
1504    000017
1506    017470
1510    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: This program utilizes a Software Switch Register which may be modified by changing Loc. 176 or by typing Control "G" (↑G) on the console terminal while the program is running.

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 and if SW00=0 then the program will assume that the status map has been built from a previous DZV11 diagnostic run.
SW 06	Set:	Reselect DZV11'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

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, mode of operation (EXTERNAL, INTERNAL, OR STAGGERED), and the number of DZV11's that are running. Using this switch alone will default the following parameters: all 4 lines are set to be tested on each DZV11, 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 LSI11 with MOS memory. When running this program on a processor with a faster memory speed this delay count should be adjusted proportionately 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 120C 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 DZV11'S DESIRED ACTIVE. A message is typed out on the console terminal asking the operator to type a bit map of the DZV's desired active. Using this switch allows location DZVACTV to be altered (see Sec. 8.3 for a description of this location).
EXAMPLE:
If the devices corresponding to the DZV11's numbered zero, two, and four in the DZV11 Status Map (Loc. 1500 through 1740) are to be tested, type in: 25
This will set bits zero, two, and four in location DZVACTV. All remaining devices in the status map will then not be tested.

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 DZV11's; the DZV11 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 DZV11 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.
(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 Go to 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 DZV11 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 - The starting address for any DZV11 diagnostic is Loc. 200

NOTE: If address 000042 is non-zero the program assumes it is under ACT11 or XXDP control and will act accordingly. After *ALL* available DZV11s are tested the program will return to 'XXDP' or 'ACT-11'.

5. OPERATING PROCEEDURE

When the program is initially started, messages as described in section four will be printed and the diagnostic will begin running.

5.1 NORMAL START OF DIAGNOSTIC

On the first start of the diagnostic at address 200, if SW00=1 then the following questions are asked and must be answered:

"1ST CSR ADDRESS (160000:163770): "

You must type in the first DZV11 CSR in the system you wish testing to begin at. RANGE: 160000:163770

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

You must type in the vector of the first DZV11 in the system under test. RANGE 300:770

```
"Maintenance Mode
 [EXTERNAL  <H325>      (E)]
 [INTERNAL  <DZCSR03=1>(I)]
 [STAGGERED <H329>      (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 H325 test connector.

"# OF DZV11'S <IN OCTAL> (1:20): "

Type total number of DZV11's to be tested in the system. RANGE is 1 thru 20 in octal.

********* IF SW03=1 THEN THE FOLLOWING WILL BE PRINTED *********

"LINES ACTIVE BY BIT <IN OCTAL> (001:017):"

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).

"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 DZV11'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 DZV11's in the system.

If the mode of operation is different for each DZV11 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 DZV11 under test and INDICATE ONE DZV11 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 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 troubleshooting 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.

6.1 ERROR RECOVERY

If for some reason the DZV11 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 'STSTNM' (address 1246) 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 DZV11 was doing at the time of the error.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

See section 4.1.2
The status table should be verified regardless of how the program was started. Also it is important to use this listing along with the information printed on the TTY to completely 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 DZV11 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.

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 DZV11'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 DVDZA-A CSR:  160100 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.

8.3 KEY LOCATIONS

```
$LPADR (1252)   Contains the address  where  program  will  return  when
                iteration  count  is  reached  or  if  loop  on  test is
                asserted.
NEXT   (1362)   Contains the address of the next test to be peformed.
$TSTNM (1246)   Contains the number of the test now being peformed.
RUN    (1412)   The bit  in  'RUN'  always  points  one  past the  DZV11
                currently     being     tested.     EXAMPLE:        (RUN)
                1412/0000000001000000 Means that DZV11 no.5 is the DZV11
                now running.
STATUS MAP
(1500)-(1740)
                These locations contain the information needed  to  test
                up to 16 (decimal) DZV11s sequentialy.  they contain the
                CSR,VECTOR and STATUS concerning  the  configuration  of
                each DZV11.
DZVACTV(1406)   Each  bit  set  in  this  location  indicates  that  the
                associated  DZV11 will  be  tested  in  turn.  EXAMPLE:
                (DZVACTV) 1406/0000000000011111  means  that  DZV11  no.
                00,01,02,03,04   will  be  tested.  EXAMPLE:  (DZVACTV)
                1406/0000000000010001 Means that DZ11 no.  00,04 will be
                tested.
$BASE  (1174)   Contains the receiver CSR  of  the  current DZV11  under
                test.
```

8.4 MORE ON THAT 'STATUS TABLE' (1500-1740)

```
'MAP OF DZV11 STATUS'
1500    160100
1502    000300
1504    000017
1506    017470
1510    000000
```

The above information will be repeated for each of up to 16 DZV11's in the system(these will follow under this table). EXPLANATION:

1500 160100 This is the system control register for the 1st DZV11 in the system.

1502 000300 This is vector 'A' for the first DZV11 in the system.

1504 000017 This is the binary representation of what lines are to be tested.

1506 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 DZV11 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.

1510 000000 This location will contain either all zeros indicating that internal loop was selected as mode of operation or it will contain 100000 indicating that "staggered mode" was selected or it will contain 000200 indicating that "external" was the mode selected.

The above is repeated for each DZV11 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 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 163770 is reached. If a 'BUS REPLY' response was issued by the DZV11 (or any other device) (no nxm trap), "MASTER SCAN ENABLE" is attempted to be set and the TCR bits for all four lines are set. "TRDY" is then tested to be set and "MASTER SCAN ENABLE" is tested to be still set. The diagnostic will then check that at least one TCR bit is still set. If all of the above worked, this device is assumed to be a DZV11. 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 DZV11, something is wrong and AUTO SIZING should not be done.

8.5.2 FINDING THE VECTOR

The vector area (address 300-776) is filled with the instruction IOT and '.+2' (next address). Bit14 and Bit5 (TX INTERUPT ENABLE AND MSTSCAN ENABLE) are set into the DZVCSR. All TCR bits are set, a delay occurs, and if no interupt occures (because of a bad DZV11) the program assumes vector address 300 and the problem should be fixed in the diagnostic. Once the problem is fixed, the program should be setup again to set the correct vector. If an interupt occurred, the address to which the DZV11 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. In this way 95% of the parameter setup was done by the program and 5% by you.
THEREFORE:
1) ALL FOUR LINES ARE ASSUMED TO BE TESTED.
2) DEFAULT BAUD RATE IS SET TO 17 (19.2 KBAUD).
3) MODE OF OPERATION IS "INTERNAL MODE".

For all parameter adjustments please refer to section 8.4 for greater detail.

9.0 RUNNING THE DZV11 DIAGNOSTIC UNDER APT

9.1.1 THE APT INTERFACE

The DZV diagnostics have been designed to be compatible with the APT (Automated Product Test) system. The DZV logic test diagnostics (DVDZA, and DVDZB) can be run as standalone diagnostics or in either of the APT modes. DVDZC, however is designed as a standalone diagnostic only and requires direct operator participation.

9.1.2 SETTING UP THE DIAGNOSTIC USING APT

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

$SWREG -(1142) used as the software switch register while running under APT.

$VECT1 -(1170) used to specify the first vector address

$BASE -(1174) used to indicate bottom address of DZV11 under test

$DEVM -(1176) a bit map representing which DZV11's will be tested

$CDW1 -(1200) used to indicate which lines to run on all DZV11's

$CDW2 -(1202) used to indicate the default test mode. Set to 0 for internal testing, 200 for external loop back (H325 installed), or set to 100000 for staggered loop back testing (H329 installed).

$DDW0 -(1204) each of the $DDW words describes the parameters (LPR) for a particular DZV11, going up to 16 DZV11's

9.1.3 RUNNING UNDER APT

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 DZV11 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
**************
 DVDZAA SEQ
**************
```

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

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

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

51        MISCELLANEOUS DEFINITIONS

63        GENERAL PURPOSE REGISTER DEFINITIONS

75        PRIORITY LEVEL DEFINITIONS

85        "SWITCH REGISTER" SWITCH DEFINITIONS

113       DA,A BIT DEFINI'IONS (BIT00 TO BIT15)

141       BASIC "CPU" TRAP VE.TOR ADDRESSES

358                              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

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

371                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABO

410       THIS TABLE CON'AINS VARIC'S COMMON STORAGE LOCATIONS
          USED IN THE PROGRAM.

462       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:

468               EM                 ;;POINTS TO THE EI .JR MESSAGE
                  DH                 ;;POINTS TO THE DATA HEADER
                  DT                 ;;POINTS TO THE DATA
                  DF                 ;;POINTS TO THE DATA FORMAT
```

```
1010    INCREMENT THE PASS NUMBER ($PASS)
        IF THERES A MONITOR GO TO IT
        IF THERE ISN'T JUMP TO CYCLE

1072    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

1147    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


1931    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.

1964    ROUTINE USED TO "AUTO SIZE" THE DZV11
        CSR AND VECTOR.
        NOTE:   THE CSR MAY BE ANY WHERE IN THE FLOATING
                ADDRESS RANGE (160000:163770)
                AND THE VECTOR MAY BE ANY WHERE IN THE
                FLOATING VECTOR RANGE (300:770)


2072    *********************** TEST 1 ******************************
        THIS TEST PROVES THE BUS REPLY RESPONSE
        DURING A READ OR WRITE TO THE FOLLOWING ADDRESS:
                DZVCSR, DZVRBUF, DZVTCR, DZVMSR

2115    *********************** TEST 2 ******************************
        THIS TEST PROVES THAT BIT "DCLR"
        CAN BE SET AND THAT IT WILL CLEAR
        BY ITSELF
```

```
2134    ************************ TEST 3 ********************************
        TEST TO VERIFY THAT THE R/W BITS OF THE
        DZVCSR REGISTER CAN BE SET. THEN VERIFY THAT
        THESE BITS CAN BE CLEARED. AND FINALLY, VERIFY
        THAT AFTER BEING SET AGAIN THEY CAN BE
        CLEARED BY A "DEVICE CLEAR".
        THE BITS TESTED ARE:  MAINT, MSENAB, SILOEN,
        RIE, AND TIE.

2185    ************************ TEST 4 ********************************
        THIS TESTS THAT ALL OF THE TCR BITS
        CAN BE: SET, CLEARED, AND CLEARED BY A DEVICE CLEAR.
        THIS TEST ALSO DETERMINES IF THE DTR BITS CAN
        BE SET, CLEARED, AND CLEARED BY A RESET.

2243    ************************ TEST 5 ********************************
        THIS TEST VERIFIES THAT
        BITS "RDONE,TRDY, BIT9, BIT8,
        AND SILOAL" ARE READ ONLY AND THAT TRDY IS
        ZERO UNTIL A LINE IS SELECTED AND MSENAB IS SET.


2275    ************************ TEST 6 ********************************
        THIS TEST VERIFIES THAT:
        TIE,SILOEN,RIE,MSENAB,AND MAINT ARE THE
        ONLY R/W BITS IN THE DZVCSR AND THAT
        SETTING "DCLR" IN THE CSR WILL CLEAR THESE BITS.

2315    ************************ TEST 7 ********************************
        THIS TEST PERFORMS RESET TESTING AND
        TESTING OF READ ONLY REGISTER DZVRBUF
        AND TESTING OF WRITE ONLY REGISTER DZVLPR

2339    ************************ TEST 10 ********************************
        THIS TEST PERFORMS RESET TESTING AND
        TESTING OF READ ONLY REGISTER DZVMSR
        AND TESTING OF WRITE ONLY REGISTER DZVTDR

2364    ************************ TEST 11 ********************************
        VERIFY THAT SETTING "DTR" FOR A LINE WILL
        BRING UP "CO" AND "RING" FOR:
        THE SAME LINE IF IN EXTERNAL MODE
        THE STAGGERED LINE IF IN STAGGERED MODE.
        LINES ARE STAGGERED AS FOLLOWS:
        LINE0 WITH LINE1; LINE2 WITH LINE3.
        THIS TEST IS ONLY RUN IF AN H325 OR H329
        IS CONNECTED ON THE DZV UNDER TEST.

2421    ************************ TEST 12 ********************************
         THIS TEST VERIFIES THAT TRDY IS SET WHEN A LINE
         IS READY TO BE LOADED, AND THAT THE LINE SPECI-
         FTED IN BITS 8-9 OF DZVCSR CORRESPOND
         TO THE LINE SELECTED IN DZVTCR
```

```
2458    *********************** TEST 13 ******************************
        TEST TO TRANSMIT ONE CHAR AND
        RECEIVE ONE CHAR ON ONE LINE
        AT A TIME. 1HE CHAR IS "252" AND
        ALL SELECTED LINES WILL BE TURNED ON .

2463    THIS IS THE FIRST TIME ANY
        DATA IS CHECKED IN THE RECEIVEP.
        USING SWITCH NINE WITH THIS TEST CREATES A TIGHT SCOPE LOOP
        WHICH TRANSMITS A STEADY STREAM OF CHARACTERS.

2539    *********************** TEST 14 ******************************
        THIS TEST VERIFIES THAT EACH RECEIVING LINE CAN BE
        DISABLED BY SETTING RCVON (BIT12 IN THE LPR REGISTER)
        TO ZERO FOR EACH LINE.
        THIS TEST ALSO VERIFIES THAT THE SILO CAN BE
        EMPTIED BY ISSUING A DEVICE MASTER CLEAR.

2624    *********************** TEST 15 ******************************
         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

2698    *********************** TEST 16 ******************************
        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.

2751    *********************** TEST 17 ******************************
         THIS TEST VERIFIES THAT THE DEVICE DOES NOT INTERRUPT
        WHILE THE PROCESSOR STPTUS DOES NOT ALLOW INTERRUPTS
        BUT WILL INTERRUPT IF THE PROCESSOR STATUS
        ALLOWS INTERRUPTS.

2836    *********************** TEST 20 ******************************
        THIS TEST VERIFIES THAT THE RECEIVER WILL
        INTERRUPT BEFORE THE TRANSMITTER EVEN
        THOUGH THE TRANSMITTER WAS ENABLED
        FIRST.  SET PS TO HIGH (MASK INTERRUPTS);
        GET RDONE AND TRDY TO SET;
        SET TX IE AND RX IE;
        CLEAR PS AND EXPECT RX TO INTERRUPT FIRST
```

```
                        .TITLE  MD-11-DVDZA-A
                        ;*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-DVDZAA/<200>/FOUR LINE ASYNC MUX TESTS, PART 1 OF 2"
                                ;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

                        .REM    !
                        ;SWITCH REGISTER OPTIONS
                        ;-----------------------
                        ;

                        SW15=100000             ;=1,HALT ON ERROR
                        SW14=40000              ;=1,LOOP ON CURRENT TEST
                        SW13=20000              ;=1,INHIBIT ERROR TYPEOUT
                        SW12=10000              ;=1,DELETE TYPEOUT/BELL ON ERROR.
                        SW11=4000               ;=1,INHIBIT ITERATIONS
                        SW10=2000               ;=1,ESCAPE TO NEXT TEST ON ERROR
                        SW09=1000               ;=1,LOOP WITH CURRENT DATA
                        SW08=400                ;=1,LOOP ON ERROR
                        SW07=200                ;=1, DO "AUTO SIZING" ON INITAL START UP.
                        SW06=100                ;=1, DESELECT SPECIFIC DEVICES
                                                ;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
                        SW05=40
                        SW04=20                 ;=1, SELECT DELAY PARAMETER
                        SW03=10                 ;=1, SELECT SPECIFIC PARAMETERS
                        SW02=4                  ;=1, LOCK ON TEST SELECT
                        SW01=2                  ;=1, RESTART PROGRAM AT SELECTED TEST
                        SW00=1                  ;=1, SELECT DEVICE ADDRESS, VECTOR, ETC.
                        !
                        .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
```

```
    57                                      .EQUIV  PS,PSW
    58          177774                      STKLMT= 177774          ;;STACK LIMIT REGISTER
    59          177772                      PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
    60          177570                      DSWR=   177570          ;;HARDWARE SWITCH REGISTER
    61          177570                      DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
    62
    63                                      ;*GENERAL PURPOSE REGISTER DEFINITIONS
    64          000000                      R0=     %0              ;;GENERAL REGISTER
    65          000001                      R1=     %1              ;;GENERAL REGISTER
    66          000002                      R2=     %2              ;;GENERAL REGISTER
    67          000003                      R3=     %3              ;;GENERAL REGISTER
    68          000004                      R4=     %4              ;;GENERAL REGISTER
    69          000005                      R5=     %5              ;;GENERAL REGISTER
    70          000006                      R6=     %6              ;;GENERAL REGISTER
    71          000007                      R7=     %7              ;;GENERAL REGISTER
    72          000006                      SP=     %6              ;;STACK POINTER
    73          000007                      PC=     %7              ;;PROGRAM COUNTER
    74
    75                                      ;*PRIORITY LEVEL DEFINITIONS
    76          000000                      PR0=    0               ;;PRIORITY LEVEL 0
    77          000040                      PR1=    40              ;;PRIORITY LEVEL 1
    78          000100                      PR2=    100             ;;PRIORITY LEVEL 2
    79          000140                      PR3=    140             ;;PRIORITY LEVEL 3
    80          000200                      PR4=    200             ;;PRIORITY LEVEL 4
    81          000240                      PR5=    240             ;;PRIORITY LEVEL 5
    82          000300                      PR6=    300             ;;PRIORITY LEVEL 6
    83          000340                      PR7=    340             ;;PRIORITY LEVEL 7
    84
    85                                      ;*"SWITCH REGISTER" SWITCH DEFINITIONS
    86          100000                      SW15=   100000
    87          040000                      SW14=   40000
    88          020000                      SW13=   20000
    89          010000                      SW12=   10000
    90          004000                      SW11=   4000
    91          002000                      SW10=   2000
    92          001000                      SW09=   1000
    93          000400                      SW08=   400
    94          000200                      SW07=   200
    95          000100                      SW06=   100
    96          000040                      SW05=   40
    97          000020                      SW04=   20
    98          000010                      SW03=   10
    99          000004                      SW02=   4
   100          000002                      SW01=   2
   101          000001                      SW00=   1
   102                                      .EQUIV  SW09,SW9
   103                                      .EQUIV  SW08,SW8
   104                                      .EQUIV  SW07,SW7
   105                                      .EQUIV  SW06,SW6
   106                                      .EQUIV  SW05,SW5
   107                                      .EQUIV  SW04,SW4
   108                                      .EQUIV  SW03,SW3
   109                                      .EQUIV  SW02,SW2
   110                                      .EQUIV  SW01,SW1
   111                                      .EQUIV  SW00,SW0
   112
```

```
    113                                              ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
    114          100000                              BIT15=  100000
    115          040000                              BIT14=  40000
    116          020000                              BIT13=  20000
    117          010000                              BIT12=  10000
    118          004000                              BIT11=  4000
    119          002000                              BIT10=  2000
    120          001000                              BIT09=  1000
    121          000400                              BIT08=  400
    122          000200                              BIT07=  200
    123          000100                              BIT06=  100
    124          000040                              BIT05=  40
    125          000020                              BIT04=  20
    126          000010                              BIT03=  10
    127          000004                              BIT02=  4
    128          000002                              BIT01=  2
    129          000001                              BIT00=  1
    130                                              .EQUIV  BIT09,BIT9
    131                                              .EQUIV  BIT08,BIT8
    132                                              .EQUIV  BIT07,BIT7
    133                                              .EQUIV  BIT06,BIT6
    134                                              .EQUIV  BIT05,BIT5
    135                                              .EQUIV  BIT04,BIT4
    136                                              .EQUIV  BIT03,BIT3
    137                                              .EQUIV  BIT02,BIT2
    138                                              .EQUIV  BIT01,BIT1
    139                                              .EQUIV  BIT00,BIT0
    140
    141                                              ;*BASIC "CPU" TRAP VECTOR ADDRESSES
    142          000004                              ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
    143          000010                              RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
    144          000014                              TBITVEC=14              ;;"T" BIT
    145          000014                              TRTVEC= 14              ;;TRACE TRAP
    146          000014                              BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
    147          000020                              IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
    148          000024                              PWRVEC= 24              ;;POWER FAIL
    149          000030                              EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
    150          000034                              TRAPVEC=34              ;;"TRAP" TRAP
    151          000060                              TKVEC=  60              ;;TTY KEYBOARD VECTOR
    152          000064                              TPVEC=  64              ;;TTY PRINTER VECTOR
    153          000240                              PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
    154
    155
    156                                              ;INSTRUCTION DEFINITIONS
    157                                              ;-----------------------
    158
    159          005746                              PUSH1SP=5746     ;DECREMENT PROCESSOR STACK 1 WORD
    160          005726                              POP1SP=5726      ;INCREMENT PROCESSOR STACK 1 WORD
    161          010046                              PUSHR0=10046     ;SAVE R0 ON STACK
    162          012600                              POPR0=12600      ;RESTORE R0 FROM STACK
    163          024646                              PUSH2SP=24646    ;DECREMENT STACK TWICE
    164          022626                              POP2SP=22626     ;INCREMENT STACK TWICE
    165          000200                              MASK=BIT7        ;SET INTERRUPT MASK (INHIBIT FURTHER INTERRUPTS)
    166          000000                              CLEAR=0          ;ALLOW INTERRUPTS (CLEAR PROCESSOR STATUS)
    167
    168
```

```
                                        ;DZV11 CONTROL AND STATUS REGISTER DEFINITIONS
                                        ;(DZVCSR)       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

                                        ;DZVCSR 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


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

                                        ;DZVRBUF 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

                                        ;DZVLPR 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

        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

                                        ;DZVTCR BIT DEFINITIONS
                                        ;----------------------
000001                          TCR0=BIT0               ;ENABLE TRANSMISSION ON LINE 0
000002                          TCR1=BIT1               ;ENABLE TRANSMISSION ON LINE 1
000004                          TCR2=BIT2               ;ENABLE TRANSMISSION ON LINE 2
000010                          TCR3=BIT3               ;ENABLE TRANSMISSION ON LINE 3
000400                          DTR0=BIT8               ;DATA TERMINAL READY FOR LINE 0
001000                          DTR1=BIT9               ;DATA TERMINAL READY FOR LINE 1
002000                          DTR2=BIT10              ;DATA TERMINAL READY FOR LINE 2
004000                          DTR3=BIT11              ;DATA TERMINAL READY FOR LINE 3

                                        ;DZVMSR 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
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

                                        ;DZVTDR 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
```

```

;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
                                                ;--------------------------

        000020                          .=20
000020  004300                                  .SCOPE                  ;SCOPE LOOP HANDLER
000022  000200                                  MASK                    ;HANDLE AT PRIORITY 7
000024  007236                                  $PWRDN                  ;POWER FAIL HANDLER
000026  000340                                  340                     ;SERVICE AT PRIORITY LEVEL 7
000030  006344                                  $ERROR                  ;ERROR HANDLER
000032  000340                                  340                     ;SERVICE AT PRIORITY LEVEL 7
000034  006136                                  .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  004234                                  $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  002116                          JMP     .START          ;GO TO START OF PROGRAM


        001000                          .=1000
001000  005200  040515  047111          MTITLE: .ASCIZ  <200><12>/MAINDEC-11-DVDZAA/<200>/FOUR LINE ASYNC MUX TESTS, PAR
```

```
                001120                                 .=1120
                                               ;;*******************************************************************
                                               .SBTTL  APT MAILBOX-ETABLE

                                               ;;*******************************************************************
                                               .EVEN
        001120                                 $MAIL:                       ;;APT MAILBOX
        001120  000000                         $MSGTY: .WORD    AMSGTY     ;;MESSAGE TYPE CODE
        001122  000000                         $FATAL: .WORD    AFATAL     ;;FATAL ERROR NUMBER
        001124  000000                         $TESTN: .WORD    ATESTN     ;;TEST NUMBER
        001126  000000                         $PASS:  .WORD    APASS      ;;PASS COUNT
        001130  000000                         $DEVCT: .WORD    ADEVCT     ;;DEVICE COUNT
        001132  000000                         $UNIT:  .WORD    AUNIT      ;;I/O UNIT NUMBER
        001134  000000                         $MSGAD: .WORD    AMSGAD     ;;MESSAGE ADDRESS
        001136  000000                         $MSGLG: .WORD    AMSGLG     ;;MESSAGE LENGTH
        001140                                 $ETABLE:                     ;;APT ENVIRONMENT TABLE
        001140     000                         $ENV:   .BYTE    AENV       ;;ENVIRONMENT BYTE
        001141     000                         $ENVM:  .BYTE    AENVM      ;;ENVIRONMENT MODE BITS
        001142  000000                         $SWREG: .WORD    ASWREG     ;;APT SWITCH REGISTER
        001144  000000                         $USWR:  .WORD    AUSWR      ;;USER SWITCHES
        001146  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
        001150     000                         $MAMS1: .BYTE    AMAMS1     ;;HIGH ADDRESS,M.S. BYTE
        001151     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
        001152  000000                         $MADR1: .WORD    AMADR1     ;;HIGH ADDRESS,BLK#1
                                               ;*                           MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABO
        001154     000                         $MAMS2: .BYTE    AMAMS2     ;;HIGH ADDRESS,M.S. BYTE
        001155     000                         $MTYP2: .BYTE    AMTYP2     ;;MEM.TYPE,BLK#2
        001156  000000                         $MADR2: .WORD    AMADR2     ;;MEM.LAST ADDRESS,BLK#2
        001160     000                         $MAMS3: .BYTE    AMAMS3     ;;HIGH ADDRESS,M.S.BYTE
        001161     000                         $MTYP3: .BYTE    AMTYP3     ;;MEM.TYPE,BLK#3
        001162  000000                         $MADR3: .WORD    AMADR3     ;;MEM.LAST ADDRESS,BLK#3
        001164     000                         $MAMS4: .BYTE    AMAMS4     ;;HIGH ADDRESS,M.S.BYTE
        001165     000                         $MTYP4: .BYTE    AMTYP4     ;;MEM.TYPE,BLK#4
        001166  000000                         $MADR4: .WORD    AMADR4     ;;MEM.LAST ADDRESS,BLK#4
        001170  000300                         $VECT1: .WORD    AVECT1     ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
        001172  000000                         $VECT2: .WORD    AVECT2     ;;INTERRUPT VECTOR#2BUS PRIORITY#2
        001174  160010                         $BASE:  .WORD    ABASE      ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
        001176  000001                         $DEVM:  .WORD    ADEVM      ;;DEVICE MAP
        001200  000017                         $CDW1:  .WORD    ACDW1      ;;CONTROLLER DESCRIPTION WORD#1
        001202  000000                         $CDW2:  .WORD    ACDW2      ;;CONTROLLER DESCRIPTION WORD#2
        001204  017470                         $DDW0:  .WORD    ADDW0      ;;DEVICE DESCRIPTOR WORD#0
        001206  017470                         $DDW1:  .WORD    ADDW1      ;;DEVICE DESCRIPTOR WORD#1
        001210  017470                         $DDW2:  .WORD    ADDW2      ;;DEVICE DESCRIPTOR WORD#2
        001212  017470                         $DDW3:  .WORD    ADDW3      ;;DEVICE DESCRIPTOR WORD#3
        001214  017470                         $DDW4:  .WORD    ADDW4      ;;DEVICE DESCRIPTOR WORD#4
        001216  017470                         $DDW5:  .WORD    ADDW5      ;;DEVICE DESCRIPTOR WORD#5
```

```
001220  017470          $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
001222  017470          $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
001224  017470          $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
001226  017470          $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
001230  017470          $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
001232  017470          $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
001234  017470          $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
001236  017470          $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
001240  017470          $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
001242  017470          $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15


001244                  $ETEND:

```

```
                                        .SBTTL  COMMON TAGS

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

 001244                         $CMTAG:                                 ;;START OF COMMON TAGS
 001244  000000                         .WORD   0
 001246     000                 $TSTNM: .BYTE   0                       ;;CONTAINS THE TEST NUMBER
 001247     000                 $ERFLG: .BYTE   0                       ;;CONTAINS ERROR FLAG
 001250  000000                 $ICNT:  .WORD   0                       ;;CONTAINS SUBTEST ITERATION COUNT
 001252  000000                 $LPADR: .WORD   0                       ;;CONTAINS SCOPE LOOP ADDRESS
 001254  000000                 $LPERR: .WORD   0                       ;;CONTAINS SCOPE RETURN FOR ERRORS
 001256  000000                 $ERTTL: .WORD   0                       ;;CONTAINS TOTAL ERRORS DETECTED
 001260     000                 $ITEMB: .BYTE   0                       ;;CONTAINS ITEM CONTROL BYTE
 001261     001                 $ERMAX: .BYTE   1                       ;;CONTAINS MAX. ERRORS PER TEST
 001262  000000                 $ERRPC: .WORD   0                       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
 001264  000000                 $GDADR: .WORD   0                       ;;CONTAINS ADDRESS OF 'GOOD' DATA
 001266  000000                 $BDADR: .WORD   0                       ;;CONTAINS ADDRESS OF 'BAD' DATA
 001270  000000                 $GDDAT: .WORD   0                       ;;CONTAINS 'GOOD' DATA
 001272  000000                 $BDDAT: .WORD   0                       ;;CONTAINS 'BAD' DATA
 001274  000000                         .WORD   0                       ;;RESERVED--NOT TO BE USED
 001276  000000                         .WORD   0
 001300     000                 $AUTOB: .BYTE   0                       ;;AUTOMATIC MODE INDICATOR
 001301     000                 $INTAG: .BYTE   0                       ;;INTERRUPT MODE INDICATOR
 001302  000000                         .WORD   0
 001304  177570                 SWR:    .WORD   DSWR                    ;;ADDRESS OF SWITCH REGISTER
 001306  177570                 DISPLAY: .WORD  DDISP                   ;;ADDRESS OF DISPLAY REGISTER
 001310  177560                 $TKS:   177560                          ;;TTY KBD STATUS
 001312  177562                 $TKB:   177562                          ;;TTY KBD BUFFER
 001314  177564                 $TPS:   177564                          ;;TTY PRINTER STATUS REG. ADDRESS
 001316  177566                 $TPB:   177566                          ;;TTY PRINTER BUFFER REG. ADDRESS
 001320     000                 $NULL:  .BYTE   0                       ;;CONTAINS NULL CHARACTER FOR FILLS
 001321     002                 $FILLS: .BYTE   2                       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
 001322     012                 $FILLC: .BYTE   12                      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
 001323     000                 $TPFLG: .BYTE   0                       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
 001324  000000                 $REGAD: .WORD   0                       ;;CONTAINS THE ADDRESS FROM
                                                                        ;;WHICH ($REG0) WAS OBTAINED
 001326  000000                 $REG0:  .WORD   0                       ;;CONTAINS (($REGAD)+0)
 001330  000000                 $REG1:  .WORD   0                       ;;CONTAINS (($REGAD)+2)
 001332  000000                 $REG2:  .WORD   0                       ;;CONTAINS (($REGAD)+4)
 001334  000000                 $REG3:  .WORD   0                       ;;CONTAINS (($REGAD)+6)
 001336  000000                 $REG4:  .WORD   0                       ;;CONTAINS (($REGAD)+10)
 001340  000000                 $REG5:  .WORD   0                       ;;CONTAINS (($REGAD)+12)
 001342  000000                 $TMP0:  .WORD   0                       ;;USER DEFINED
 001344  000000                 $TMP1:  .WORD   0                       ;;USER DEFINED
 001346  000000                 $TMP2:  .WORD   0                       ;;USER DEFINED
 001350  000000                 $TMP3:  .WORD   0                       ;;USER DEFINED
 001352  000000                 $TMP4:  .WORD   0                       ;;USER DEFINED
 001354  000000                 $TIMES: 0                               ;;MAX. NUMBER OF ITERATIONS
 001356     077                 $QUES:  .ASCII  /?/                     ;;QUESTION MARK
 001357     015                 $CRLF:  .ASCII  <15>                    ;;CARRIAGE RETURN
 001360  000012                 $LF:    .ASCIZ  <12>                    ;;LINE FEED
```

```
                                        .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


001362                                  $ERRTB:

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

001362  000000                          NEXT:   0                        ;ADDRESS OF NEXT TEST TO BE EXECUTED
001364  000000                          LOCK:   0                        ;ADDRESS FOR LOCK ON CURRENT TEST,TIGHT LOOP

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

001366  000017                          LINE:   17                       ;DEFAULT ALL FOUR LINES RUNNING
001370  017470                          PAR:    17470                    ;PARAMETERS: 8 BITS/CHAR,2 STOP BITS,19200 BAUD,
001372  000000                          MODE:   0                        ;DEFAULT MAINTENANCE MODE
001374  000000                          SAVLIN: 0                        ;LINE NUMBER
001376  000000                          XMTLIN: 0                        ;TRANSMISSION LINE NUMBER
001400  000000                          XMTCNT: 0                        ;COUNT OF WORDS IN A TRANSMISSION PATTERN
001402  000000                          REGIST: 0                        ;DEVICE ADDRESS STORAGE LOCATION
001404  000000                          SAVPC:  0                        ;PROGRAM COUNTER STORAGE
001406  000001                          DZVACTV:.BLKW   1                ;*DZV11'S SELECTED ACTIVE.
001410  000001                          SAVACTV:.BLKW   1                ;*A BIT MAP OF DZV11'S IN THE SYSTEM
001412  000001                          RUN:    1                        ;*POINTER ONE PAST RUNNING DEVICE.
001414  000001                          DZVNUM: .BLKB 1                  ;*OCTAL NUMBER OF DZV11'S IN THE SYSTEM
001415     001                          SAVNUM: .BYTE 1                  ;*WORKABLE NUMBER.
001416  000001                          SAVNO:  .BLKB   1                ;*OCTAL NO. OF DZV11'S BEING TESTED
        001420                          .EVEN
001420  001500                          ACTIVE: DZV.MAP                  ;TABLE POINTER.
```

```

                                              ;PROGRAM CONTROL FLAGS
                                              ;---------------------

001422     000                       INIFLG: .BYTE   0                 ;PROGRAM INITIALIZATION FLAG
001423     000                       HDRFLG: .BYTE   0                 ;PROGRAM INITIALIZATION FLAG FOR HEADER MAP
001424     000                       MNTFLG: .BYTE   0                 ;MAINTENANCE BIT SET FLAG
001425     000                       DONFLG: .BYTE   0                 ;TRANSMISSION COMPLETION FLAG
                                     .EVEN
                                     ;DATA VARIABLES
001426  000000                       TD0:    .WORD   0
001430  000000                       TD1:    .WORD   0
001432  000000                       TD2:    .WORD   0
001434  000000                       TD3:    .WORD   0
001436  000000                       TR0:    .WORD   0
001440  000000                       TR1:    .WORD   0
001442  000000                       TR2:    .WORD   0
001444  000000                       TR3:    .WORD   0
001446                               STOP:
                                     .SBTTL  APT PARAMETER BLOCK

                                     ;;************************************************ *************
                                     ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                     ;;**************************************************************
        001446                               .$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  001446                               $APTHDR   ;;POINT TO APT HEADER BLOCK
        001446                               .=.$X     ;;RESET LOCATION COUNTER
                                     ;;**************************************************************
                                     ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                     ;INTERFACE SPEC.

001446                               $APTHD:
001446  000000                       $HIBTS: .WORD   0        ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001450  001120                       $MBADR: .WORD   $MAIL    ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001452  000120                       $TSTM:  .WORD   80.      ;;RUN TIM OF LONGEST TEST
001454  000024                       $PASTM: .WORD   20.      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001456  000000                       $UNITM: .WORD   0.       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITION
001460  000052                               .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
                                             ;DZV11 STATUS TABLE AND ADDRESS ASSIGNMENTS
                                             ;-----------------------------------------

        001500                       .=1500
001500                               DZV.MAP:

001500  000001                       DZCR0:  .BLKW   1        ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 0
001502  000001                       DZVC0:  .BLKW   1        ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 0
001504  000001                       LINE0:  .BLKW   1        ;ALL LINES SELECTED
001506  000001                       PAR0:   .BLKW   1        ;PARAMETERS
001510  000001                       MANT0:  .BLKW   1        ;MAINTENANCE MODE FOR THIS DEVICE

001512  000001                       DZCR1:  .BLKW   1        ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 1
001514  000001                       DZVC1:  .BLKW   1        ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 1
001516  000001                       LINE1:  .BLKW   1        ;ALL LINES SELECTED
```

```
    557  001520  000001                        PAR1:    .BLKW    1         ;PARAMETERS
    558  001522  000001                        MANT1:   .BLKW    1         ;MAINTENANCE MODE FOR THIS DEVICE
    559
    560  001524  000001                        DZCR2:   .BLKW    1         ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 2
    561  001526  000001                        DZVC2:   .BLKW    1         ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 2
    562  001530  000001                        LINE2:   .BLKW    1         ;ALL LINES SELECTED
    563  001532  000001                        PAR2:    .BLKW    1         ;PARAMETERS
    564  001534  000001                        MANT2:   .BLKW    1         ;MAINTENANCE MODE FOR THIS DEVICE
    565
    566  001536  000001                        DZCR3:   .BLKW    1         ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 3
    567  001540  000001                        DZVC3:   .BLKW    1         ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 3
    568  001542  000001                        LINE3:   .BLKW    1         ;ALL LINES SELECTED
    569  001544  000001                        PAR3:    .BLKW    1         ;PARAMETERS
    570  001546  000001                        MANT3:   .BLKW    1         ;MAINTENANCE MODE FOR THIS DEVICE
    571
    572  001550  000001                        DZCR4:   .BLKW    1         ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 4
    573  001552  000001                        DZVC4:   .BLKW    1         ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 4
    574  001554  000001                        LINE4:   .BLKW    1         ;ALL LINES SELECTED
    575  001556  000001                        PAR4:    .BLKW    1         ;PARAMETERS
    576  001560  000001                        MANT4:   .BLKW    1         ;MAINTENANCE MODE FOR THIS DEVICE
    577
    578  001562  000001                        DZCR5:   .BLKW    1         ;CONTROL STATUS REGISTER FOR DZV11 NUMAER 5
    579  001564  000001                        DZVC5:   .BLKW    1         ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 5
    580  001566  000001                        LINE5:   .BLKW    1         ;ALL LINES SELECTED
    581  001570  000001                        PAR5:    .BLKW    1         ;PARAMETERS
    582  001572  000001                        MANT5:   .BLKW    1         ;MAINTENANCE MODE FOR THIS DEVICE
    583
    584  001574  000001                        DZCR6:   .BLKW    1         ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 6
    585  001576  000001                        DZVC6:   .BLKW    1         ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 6
    586  001600  000001                        LINE6:   .BLKW    1         ;ALL LINES SELECTED
    587  001602  000001                        PAR6:    .BLKW    1         ;PARAMETERS
    588  001604  000001                        MANT6:   .BLKW    1         ;MAINTENANCE MODE FOR THIS DEVICE
    589
    590  001606  000001                        DZCR7:   .BLKW    1         ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 7
    591  001610  000001                        DZVC7:   .BLKW    1         ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 7
    592  001612  000001                        LINE7:   .BLKW    1         ;ALL LINES SELECTED
    593  001614  000001                        PAR7:    .BLKW    1         ;PARAMETERS
    594  001616  000001                        MANT7:   .BLKW    1         ;MAINTENANCE MODE FOR THIS DEVICE
    595
    596  001620  000001                        DZCR10:  .BLKW    1         ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 10
    597  001622  000001                        DZVC10:  .BLKW    1         ;RECEIVER ANC BASE VECTOR  FOR DZV11 NUMBER 10
    598  001624  000001                        LINE10:  .BLKW    1         ;ALL LINES SELECTED
    599  001626  000001                        PAR10:   .BLKW    1         ;PARAMETERS
    600  001630  000001                        MANT10:  .BLKW    1         ;MAINTENANCE MODE FOR THIS DEVICE
    601
    602  001632  000001                        DZCR11:  .BLKW    1         ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 11
    603  001634  000001                        DZVC11:  .BLKW    1         ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 11
    604  001636  000001                        LINE11:  .BLKW    1         ;ALL LINES SELECTED
    605  001640  000001                        PAR11:   .BLKW    1         ;PARAMETERS
    606  001642  000001                        MANT11:  .BLKW    1         ;MAINTENANCE MODE FOR THIS DEVICE
    607
    608  001644  000001                        DZCR12:  .BLKW    1         ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 12
    609  001646  000001                        DZVC12:  .BLKW    1         ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 12
    610  001650  000001                        LINE12:  .BLKW    1         ;ALL LINES SELECTED
    611  001652  000001                        PAR12:   .BLKW    1         ;PARAMETERS
    612  001654  000001                        MANT12:  .BLKW    1         ;MAINTENANCE MODE FOR THIS DEVICE
```

```

001656  000001                       DZCR13: .BLKW   1          ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 13
001660  000001                       DZVC13: .BLKW   1          ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 13
001662  000001                       LINE13: .BLKW   1          ;ALL LINES SELECTED
001664  000001                       PAR13:  .BLKW   1          ;PARAMETERS
001666  000001                       MANT13: .BLKW   1          ;MAINTENANCE MODE FOR THIS DEVICE

001670  000001                       DZCR14: .BLKW   1          ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 14
001672  000001                       DZVC14: .BLKW   1          ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 14
001674  000001                       LINE14: .BLKW   1          ;ALL LINES SELECTED
001676  000001                       PAR14:  .BLKW   1          ;PARAMETERS
001700  000001                       MANT14: .BLKW   1          ;MAINTENANCE MODE FOR THIS DEVICE

001702  000001                       DZCR15: .BLKW   1          ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 15
001704  000001                       DZVC15: .BLKW   1          ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 15
001706  000001                       LINE15: .BLKW   1          ;ALL LINES SELECTED
001710  000001                       PAR15:  .BLKW   1          ;PARAMETERS
001712  000001                       MANT15: .BLKW   1          ;MAINTENANCE MODE FOR THIS DEVICE

001714  000001                       DZCR16: .BLKW   1          ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 16
001716  000001                       DZVC16: .BLKW   1          ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 16
001720  000001                       LINE16: .BLKW   1          ;ALL LINES SELECTED
001722  000001                       PAR16:  .BLKW   1          ;PARAMETERS
001724  000001                       MANT16: .BLKW   1          ;MAINTENANCE MODE FOR THIS DEVICE

001726  000001                       DZCR17: .BLKW   1          ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 17
001730  000001                       DZVC17: .BLKW   1          ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 17
001732  000001                       LINE17: .BLKW   1          ;ALL LINES SELECTED
001734  000001                       PAR17:  .BLKW   1          ;PARAMETERS
001736  000001                       MANT17: .BLKW   1          ;MAINTENANCE MODE FOR THIS DEVICE

001740  177777                       DZV.END:        177777
```

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


                                          ;:*************************************************************
                                          ;------------------------------------------------------------
001742                                    .TRPTAB:
        104400                            ADVANCE=TRAP+0          ;CALL TO ADVANCE TO NEXT TEST( OR SCOPE THIS ONE)
001742  006232                                    .ADVANCE
        104401                            SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
001744  004544                                    .SCOP1
        104402                            TYPE=TRAP+2             ;CALL T^ TELETYPE OUTPUT ROUTINE
001746  004570                                    .TYPE
        104403                            INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
001750  005336                                    .INSTR
        104404                            INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
001752  005442                                    .INSTER
        104405                            PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
001754  005462                                    .PARAM
        104406                            SETFLG=TRAP+6           ;CALL TO  SET FLAG ROUTINE
001756  010074                                    .SETFLG
        104407                            SAV05=TRAP+7            ;CALL TO REGISTER SAVE ROUTINE
001760  005662                                    .SAV05
        104410                            RES05=TRAP+10           ;CALL TO REGISTER RESTORE ROUTINE
001762  005722                                    .RES05
        104411                            CONVRT=TRAP+11          ;CALL TO DATA OUTPUT ROUTINE
001764  005754                                    .CONVRT
        104412                            CNVRT=TRAP+12           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
001766  005760                                    .CNVRT
        104413                            DEVICE.CLR=TRAP+13               ;CALL TO ISSUE A DEVICE CLEAR
001770  006160                                    .DEVICE.CLR
        104414                            DELAY=TRAP+14           ;CALL TO DELAY FOR FAST CPU'S
001772  006212                                    .DELAY
        104415                            PARMD=TRAP+15           ;CONVERT DECIMAL STRING TO OCTAL
001/74  011142                                    .PARMD
        104416                            PAWCH=TRAP+16           ;SET FLAG    ECHO OR  CABLE
001776  010214                                    .PAWCH
        104417                            DCLASM=TRAP+17          ;CLEAR DEVICE, SET MAINT. BIT IF I MODE
002000  006200                                    .DCLASM
        104420                            SHIFT=TRAP+20           ;CALL TO ROTATE LINE POINTER
002002  006244                                    .SHIFT
        104421                            LPRSET=TRAP+21          ;CALL TO SET UP LPR DEVICE REGISTER
002004  006262                                    .LPRSET
        104422                            BUFSET=TRAP+22          ;CALL TO ZERO BUFFER AREA
002006  006322                                    .BUFSET

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

```
                                             ;DZV11 VECTOR AND REGISTER INDIRECT POINTERS
                                             ;WORKING AREA

     002010  160040                  DZVCSR: 160040  ;R/W
     002012  160041                  HDZVCSR:160041  ;R/W
     002014  160042                  DZVRBUF:160042  ;READ ONLY
     002016  160043                  HDZVRBUF:160043 ;READ ONLY
     002020  160042                  DZVLPR: 160042  ;WRITE ONLY
     002022  160043                  HDZVLPR:160043  ;WRITE ONL.
     002024  160044                  DZVTCR: 160044  ;R/W
     002026  160045                  HDZVTCR:160045  ;R/W
     002030  160046                  DZVMSR: 160046  ;READ ONLY
     002032  160047                  HDZVMSR:160047  ;READ ONLY
     002034  160046                  DZVTDR: 160046  ;WRITE ONLY
     002036  160047                  HDZVTDR:160047  ;WRITE ONLY

                                             ;DEFAULT DZV VECTORS

     002040  000300                  DZVRIV: 300     ;REC INTR VECTOR
     002042  000302                  DZVRIS: 302     ;REC INTR STATUS
     002044  000304                  DZVTIV: 304     ;XMIT INTR VECTOR
     002046  000306                  DZVTIS: 306     ;XMIT INTR STATUS
```

```

                                                      ;TIME TABLE FOR RELATIVE TIMING TESTS
                                                      ;-----------------------------------

      002050                                  TMTBL:
      002050    000000                        T50:      0
      002052    000000                        T75:      0
      002054    000000                        T110:     0
      002056    000000                        T134:     0
      002060    000000                        T150:     0
      002062    000000                        T300:     0
      002064    000000                        T600:     0
      002066    000000                        T1200:    0
      002070    000000                        T1800:    0
      002072    000000                        T2000:    0
      002074    000000                        T2400:    0
      002076    000000                        T3600:    0
      002100    000000                        T4800:    0
      002102    000000                        T7200:    0
      002104    000000                        T9600:    0
      002106    000000                        TEIGHT:0
      002110    000000                        TSEVEN:   0
      002112    000000                        TSIX:     0
      002114    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

002116                             .START:
002116  000005                             RESET                           ;CLEAR THE WORLD. START NEW ENVIRONMENT
002120  012706  001120                     MOV     #STACK,SP               ;SET UP STACK
002124  106427  000200                     MTPS    #MASK                   ;LOCK OUT INTERRUPTS
002130  012737  007236  000024             MOV     #$PWRON,@#24            ;SET UP POWER FAIL VECTOR
002136  005037  001126                     CLR     $PASS                   ;CLEAR PASS COUNT
002142  105037  001247                     CLRB    $ERFLG                  ;CLEAR ERROR FLAG
002146  012737  001500  001420             MOV     #DZV.MAP,ACTIVE         ;GET MAP POINTER.
002154  012737  000001  001412             MOV     #1,RUN                  ;POINT POINTER TO FIRST DEVICE.
002162  005037  001256                     CLR     $ERTTL                  ;CLEAR ERROR COUNT
002166  005037  001262                     CLR     $ERRPC                  ;CLEAR LAST ERROR POINTER
002172  005037  001246                     CLR     $TSTNM                  ;SET UP FOR TEST 1
002176  012737  002116  001252             MOV     #.START,$LPADR          ;SET UP FOR POWER FAIL BEFORE
                                                                           ;TESTING STARTS
                                           ;SET UP FOR SMALL 11 SWITCH REGISTER COMPATIBILITY
002204  012737  000176  001304             MOV     #SWREG,SWR              ;POINT TO SOFTWARE SWR
002212  012737  000174  001306             MOV     #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REGISTER
002220  105737  001422                     TSTB    INIFLG                  ;HAVE WE ALREADY BEEN HERE TODAY?
002224  001010                             BNE     10$                     ;IF SO, SKIP PRINTING THE TITLE
002226  023727  000042  004234             CMP     @#42,#$ENDAD            ;IF RUNNING UNDER ACT
002234  001402                             BEQ     1$                      ;DON'T PRINT TITLE
002236  104402  001000                     TYPE    ,MTITLE                 ;PRINT THE DIAGNOSTIC'S TITLE
002242  105337  001422             1$:     DECB    INIFLG                  ;SET THE ONCE ONLY FL'G
002246  105737  001141             10$:    TSTB    $ENVM                   ;DETERMINE WHETHER AP, SIZING SHOULD BE DONE
002252  100004                             BPL     15$                     ;IF NOT, GO CHECK FOR AUTO-SIZING
002254  004737  011336                     JSR     PC,SETAPT               ;OTHERWISE, GO DO APT SIZING FROM ETABLE
002260  000137  003540                     JMP     105$                    ;GO PRINT DZV STATUS TABLE
002264  032777  000001  177012     15$:    BIT     #SW00,@SWR              ;RESELECT ?
002272  001002                             BNE     20$                     ;IF YES, GO SET UP THE INFORMATION
002274  000137  002576                     JMP     55$                     ;IF NO, SKIP THE INTERROGATION
002300  012700  001500             20$:    MOV     #DZV.MAP,R0             ;POINT TO THE BEGINNING OF THE MAP TABLE
002304  105037  001423                     CLRB    HDRFLG                  ;MAKE SURE A MAP GETS PRINTED
002310  005020                     25$:    CLR     (R0)+                   ;CLEAR A TABLE LOCATION
002312  020027  001740                     CMP     R0,#DZV.END             ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
002316  001374                             BNE     25$                     ;IF NOT ,CLEAR THE NEXT LOCATION IN THE TABLE
002320  105337  001422                     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 DZV11'S

002324  104403                             INSTR                           ;CALL THE STRING INPUT ROUTINE
002326  003016                             91$                             ;POINTER TO MESSAGE TO BE PRINTED
002330  104405                             PARAM                           ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
002332  160000                             160000                          ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002334  163770                             163770                          ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002336  001500                             DZCR0                           ;POINTER TO MAP LOCATION TO BE FILLED
```

```
002340     007                          .BYTE   7               ;MASK OF INVALID BITS FOR THIS PARAMETER
002341     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
002342  013737  001500  001174          MOV     DZCR0,$BASE     ;COPY BASE ADDRESS TO ETABLE

                                        ;GET THE BASE VECTOR ADDRESS

002350  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
002352  003062                          92$                     ;POINTER TO MESSAGE TO BE PRINTED
002354  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
002356  000300                          300                     ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002360  000776                          776                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002362  001502                          DZVC0                   ;POINTER TO MAP LOCATION TO BE FILLED
002364     003                          .BYTE   3               ;MASK OF INVALID BITS FOR THIS PARAMETER
002365     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
002366  013737  001502  001170          MOV     DZVC0,$VECT1    ;COPY VECTOR TO ETABLE
                                        ;GET THE MODE OF OPERATION (E,I,S)

002374  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
002376  003311                          96$                     ;POINTER TO THE MESSAGE TO BE PRINTED
002400  104406                          SETFLG                  ;CALL THE MAINTENANCE FLAG SETUP ROUTINE
002402  001510                          MANT0                   ;THIS IS THE FLAG BEING SETUP

                                        ;GET THE NUMBER OF DZV11'S RUNNING

002404  104403                          INSTR                   ;CALL THE STRING INPUT ROUTINE
002406  003246                          95$                     ;POINTER TO MESSAGE TO BE PRINTED
002410  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
002412  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002414  000020                          16.                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002416  001344                          $TMP1                   ;POINTER TO MAP LOCATION TO BE FILLED
002420     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
002421     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE

002422  012737  000017  001504          MOV     #17,LINE0       ;SET UP DEFAULT LINES
002430  012737  017470  001506          MOV     #17470,PAR0     ;SET UP DEFAULT LPR PARAMETER
                                                                ;RECEIVER ON; 19.2 KBAUD; 2STOP BITS; 8 BIT/CHAR
002436  032777  000010  176640          BIT     #SW03,@SWR      ;DO YOU WANT PARAMETERS?
002444  001402                          BEQ     30$             ;IF NO, SKIP THE PARAMETER CALL
002446  004737  002626                  JSR     PC,65$          ;GET PARAMETERS
002452  012737  000001  001410  30$:    MOV     #1,SAVACTV      ;INITIALIZE ACTIVE DEVICE SELECTION PARAMETER
002460  113737  001344  001414          MOVB    $TMP1,DZVNUM    ;COPY THE NUMBER OF DEVICES
002466  005337  001344          35$:    DEC     $TMP1           ;$TMP1 CONTAINS THE COUNT OF UNINITIALIZED
002472  001404                          BEQ     40$             ; SELECTED DEVICES
002474  000261                          SEC                     ;SET A BIT FLAG TO INDICATE AN ACTIVE DEVICE
002476  006137  001410                  ROL     SAVACTV         ;POINT TO THE NEXT DEVICE
002502  000771                          BR      35$             ;GO DO THIS PROCEDURE AGAIN
002504  013737  001410  001346  40$:    MOV     SAVACTV,$TMP2   ;# OF TIMES
002512  012700  001500                  MOV     #DZCR0,R0       ;SET A POINTER TO THE SPECIFIED INFORMATION
002516  012701  001512                  MOV     #DZCR1,R1       ;POINT R1 TO THE REST OF THE MAP TABLE
002522  012702  001204                  MOV     #$DDW0,R2       ;POINT TO ETABLE'S DEVICE DESCRIPTOR WORDS
002526  000241                          CLC                     ;INITIALIZE THE "C" BIT FOR A ROTATION
002530  006037  001346                  ROR     $TMP2           ;SKIP MAPPING SETUP FOR DEVICE 0- IT'S DONE
002534  006237  001346          45$:    ASR     $TMP2           ;ISOLATE A SELECTION FLAG IN THE "C" BIT
002540  103404                          BCS     50$             ;IS THIS DEVICE SELECTED? IF YES, GO LOAD TABLE
002542  012711  177777                  MOV     #-1,(R1)        ;TERMINATE THE LIST
002546  000137  003514                  JMP     100$            ;GO TO THE NEXT BLOCK
```

```
002552  012011                          50$:    MOV     (R0)+,(R1)      ;ADDRESS
002554  062721  000010                          ADD     #10,(R1)+       ;POINT TO THE NEXT DZV11 ADDRESS VALUE
002560  012011                                  MOV     (R0)+,(R1)      ;VECTOR
002562  062721  000010                          ADD     #10,(R1)+       ;POINT TO THE NEXT VECTOR VALUE
002566  012021                                  MOV     (R0)+,(R1)+     ;LINES
002570  012021                                  MOV     (R0)+,(R1)+     ;PARAMETERS
002572  012021                                  MOV     (R0)+,(R1)+     ;MAINTENANCE MODE
002574  000757                                  BR      45$
002576  032777  000010  176500          55$:    BIT     #SW03,@SWR      ;ASK PARAMETERS ?
002604  001002                                  BNE     60$             ;IF NO, GO DO AUTO SIZING
002606  000137  003514                          JMP     100$            ;GO SET UP FOR AUTO SIZING
002612  004737  002626                  60$:    JSR     PC,65$          ;GO ASK PARAMETERS
002616  105337  001422                          DECB    INIFLG          ;INSURE NO AUTO SIZE IF QULSTIONS ANSWERED
002622  000137  003540                          JMP     105$            ;GO TO THE NEXT BLOCK

                                                ;GET THE ACTIVE LINES PARAMETER

002626                                  65$:
002626  104403                                  INSTR                   ;CALL THE STRING INPUT ROUTINE
002630  003123                                  93$                     ;POINTER TO MESSAGE TO BE PRINTED
002632  104405                                  PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
002634  000001                                  1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002636  000017                                  17                      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002640  001504                                  LINE0                   ;POINTER TO MAP LOCATION TO BE FILLED
002642     360                                  .BYTE   360             ;MASK OF INVALID BITS FOR THIS PARAMETER
002643     001                                  .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
002644  105037  001423                          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

002650  005737  001510                          TST     MANT0           ;IS STAGGERED THE MODE OF OPERATION?
002654  100021                                  BPL     85$             ;IF NOT, SKIP THIS SEGMENT
002656  013703  001504                          MOV     LINE0,R3        ;GET A SCRATCH COPY OF THE ACTIVE LINES
002662  006003                          70$:    ROR     R3              ;GET A LINE SELECTION BIT(EVEN NUMBER LINE)
002664  103410                                  BCS     80$             ;IF IT IS SELECTED, CHECK TO SEE IF THE NEXT IS
002666  001414                                  BEQ     85$             ;IF ALL HAVE BEEN CHECKED, CONTINUE PROCESSING
002670  006203                                  ASR     R3              ;IF IT IS 0,CHECK TO SEE IF THE NEXT IS TOO
002672  103373                                  BCC     70$             ;IF THIS ONE'S 0 TOO, GO CHECK THE NEXT PAIR
002674  104402  001356                  75$:    TYPE    ,$QUES          ;THIS IS AN INCORRECT PARAMETER
002700  104402  010020                          TYPE    ,MBADLN         ;LET THE USER KNOW ABOUT IT
002704  000750                                  BR      65$             ;GO GET THE CORRECT PARAMETER
002706  001772                          80$:    BEQ     75$             ;IF ANOTHER FLAG ISN'T SET, THERE'S AN ERROR
002710  006203                                  ASR     R3              ;GET THE NEXT FLAG
002712  103370                                  BCC     75$             ;IF IT ISN'T SET, THERE'S AN ERROR
002714  000241                                  CLC                     ;INITIALIZE THE "C" BIT FOR TESTING OF THE NEXT
002716  000761                                  BR      70$             ;GO TEST THE NEXT PAIR OF FLAGS

                                                ;GET THE LINE PARAMETER REGISTER ARGUMENT

002720                                  85$:
002720  104403                                  INSTR                   ;CALL THE STRING INPUT ROUTINE
002722  003176                                  94$                     ;POINTER TO MESSAGE TO BE PRINTED
002724  104405                                  PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
002726  000000                                  0                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
002730  000017                                  17                      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
```

```
    002732  001506                                  PAR0                            ;POINTER TO MAP LOCATION TO BE FILLED
    002734     000                                  .BYTE   0                       ;MASK OF INVALID BITS FOR THIS PARAMETER
    002735     001                                  .BYTE   1                       ;NUMBER OF PARAMETERS TO STORE
    002736  012702  001504                          MOV     #LINE0,R2               ;POINT TO THE LINE SELECTION PARAMETER
    002742  012703  001506                          MOV     #PAR0,R3                ;POINT TO THE CHOSEN PARAMETERS
    002746  011304                                  MOV     (R3),R4                 ;USE BAUD RATE AS AN INDEX IN DELAY TABLE
    002750  006304                                  ASL     R4                      ;ALIGN INDEX ON WORD BOUNDARY
    002752  016437  017360  006230                  MOV     DLYTBL(R4),DLYCNT ;SET THE DELAY COUNT FOR THIS BAUD RATE
    002760  000313                                  SWAB    (R3)                    ;PLACE IN HIGH BYTE
    002762  052713  010070                          BIS     #10070,(R3)             ;PLACE EXTRA PARAMETERS INTO LOC
    002766  011262  000012                  90$:    MOV     (R2',12(R2)             ;LOAD THE LINES
    002772  011363  000012                          MOV     (R3),12(R3)             ;LOAD THE PARAMETERS
    002776  062702  000012                          ADD     #12,R2                  ;POINT TO THE NEXT SET
    003002  062703  000012                          ADD     #12,R3                  ; .. OF BOTH PARAMETERS
    003006  020327  001734                          CMP     R3,#PAR17               ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
    003012  001365                                  BNE     90$                     ;IF NOT, GO LOAD SOME MORE PARAMETERS
    003014  000207                                  RTS     PC                      ;RETURN TO CALLING BLOCK
    003016  030600  052123  041440          91$:    .ASCIZ  <200>/1ST CSR ADDRESS (160000:163770):  /
(1) 003062  030600  052123  053040          92$:    .ASCIZ  <200>/1ST VECTOR ADDRESS (300:770):  /
(1) 003123     200  044514  042516          93$:    .ASCIZ  <200>/LINES ACTIVE BY BIT (IN OCTAL)(001:17):  /
(1) 003176  042200  043105  052501          94$:    .ASCIZ  <200>/DEFAULT BAUD RATE (IN OCTAL)(00:17):  /
(1) 003246  021600  047440  020106          95$:    .ASCIZ  <200>/# OF DZV11'S (IN OCTAL) (1:20):  /
(1) 003311     200  040515  047111          96$:    .ASCII  <200>/MAINTENANCE MODE/
(1) 003332  020200  042533  052130                  .ASCII  <200>/ [EXTERNAL  (H325)      (E)]/
(1) 003366  020200  044533  052116                  .ASCII  <200>/ [INTERNAL  (DZVCSR03=1'(I)]/
(1) 003423     200  055440  052123                  .ASCIZ  <200>/ [STAGGERED (H329)      (S)]: /
(1) 003462  042600  052116  051105          97$:    .ASCIZ  <200>/ENTER DELAY PARAMETER: /
(1)         003514                          .EVEN
(1) 003514                                  100$:
    003514  122737  000377  001422                  CMPB    #377,INIFLG             ;ONLY DO AUTO SIZE ON 1ST START
    003522  001006                                  BNE     105$                    ;
    003524  032777  000200  175552                  BIT     #BIT7,@SWR              ;BIT7=1??
    003532  001002                                  BNE     105$                    ;BR IF NO AUTO SIZE
    003534  004737  011464                          JSR     PC,AUTO.SIZE            ;GO DO THE AUTO SIZE
    003540  105737  001423                  105$:   TSTB    HDRFLG                  ;HAS THE TABLE BEEN TYPED YET?
    003544  001021                                  BNE     120$                    ;IF SO, DON'T TYPE IT AGAIN
    003546  105337  001423                          DECB    HDRFLG                  ;INDICATE THAT THE TABLE WILL BE TYPED
    003552  104402  007772                          TYPE    ,XHEAD                  ;TYPE MAP HEADER
    003556  012700  001500                          MOV     #DZV.MAP,R0             ;SET POINTER
    003562  010037  001344                  110$:   MOV     R0,$TMP1                ;POINT TO THE MAP LOCATION
    003566  012037  001346                          MOV     (R0)+,$TMP2             ;SET DATA
    003572  022737  177777  001346                  CMP     #-1,$TMP2               ;END OF LIST?
    003600  001403                                  BEQ     120$                    ;BR IF YES
    003602  104411                          115$:   CONVRT                          ;CALL THE OCTAL TO ASCII CONVERSION ROUTINE
    003604  010062                                  XSTATQ                          ;CONVERT THE DATA AT THIS ADDRESS
    003606  000765                                  BR      110$                    ;GO PRINT THE NEXT PARAMETER
    003610  013737  001410  001406          120$:   MOV     SAVACTV,DZVACTV ;COPY BIT MAP OF SYSTEM DEVICES ACTIVE
    003616  113737  001414  001416                  MOVB    DZVNUM,SAVNO            ;COPY NO. OF SYSTEM DEVICES ACTIVE
    003624  032777  000100  175452                  BIT     #SW06,@SWR              ;DESELECT SPECIFIC DEVICES??
    003632  001431                                  BEQ     135$                    ;BR IF NO.
    003634                                  121$:
    003634  104403                                  INSTR                           ;CALL THE STRING INPUT ROUTINE
    003636  007710                                  MNEW                            ;POINTER TO MESSAGE TO BE PRINTED
    003640  104405                                  PARAM                           ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
    003642  000001                                  1                               ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
    003644  177777                                  177777                          ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
```

```
003646  001406                                      DZVACTV                     ;POINTER TO MAP LOCATION TO BE FILLED
003650     000                                      .BYTE   0                   ;MASK OF INVALID BITS FOR THIS PARAMETER
003651     001                                      .BYTE   1                   ;NUMBER OF PARAMETERS TO STORE
003652  023737  001406  001410                      CMF     DZVACTV,SAVACTV     ;IS THE VALUE VALID?
003660  101403                                      BLOS    122$                ;BRANCH IF YES
003662  104402  007562                              TYPE    ,MERR3              ;IF NOT THEN TYPE ERROR
003666  000762                                      BR      121$                ;GO REASK QUESTION
003670  105037  001416                      122$:   CLRB    SAVNO               ;CLEAR NO. OF DEVICES BEING TESTED
003674  013737  001406  001344                      MOV     DZVACTV,$TMP1       ;COPY BIT MAP OF ACTIVE DEVICES BEING TESTED
003702  006237  001344                      126$:   ASR     $TMP1               ;SHIFT OUT AN ACTIVE BIT
003706  103002                                      BCC     127$                ;IF NOT ACTIVE SKIP INCREMENT
003710  105237  001416                              INCB    SAVNO               ;IF ACTIVE RECORD IT
003714  001372                              127$:   BNE     126$                ;IF ALL ACTIVE BITS RECORDED DON'T BRANCH
003716  032777  000020  175360              135$:   BIT     #SW04,@SWR          ;CHECK TO SEE IF DELAY COUNT CHANGES
003724  001407                                      BEQ     140$                ;IF NOT, GO CLEAR VECTOR AREA
003726  104403                                      INSTR                       ;CALL THE STRING INPUT ROUTINE
003730  003462                                      97$                         ;POINTER TO MESSAGE TO BE PRINTED
003732  104405                                      PARAM                       ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
003734  000001                                      1                           ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
003736  177777                                      177777                      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
003740  006230                                      DLYCNT                      ;POINTER TO MAP LOCATION TO BE FILLED
003742     000                                      .BYTE   0                   ;MASK OF INVALID BITS FOR THIS PARAMETER
003743     001                                      .BYTE   1                   ;NUMBER OF PARAMETERS TO STORE
003744  012700  000300                      140$:   MOV     #300,R0             ;PREPARE TO CLEAR THE FLOATING
003750  012701  000302                              MOV     #302,R1             ;VECTOR AREA. 300-776
003754  010120                              145$:   MOV     R1,(R0)+            ;START PUTTING "PC+2 - HALT"
003756  005021                                      CLR     (R1)+               ;IN VECTOR AREA.
003760  022021                                      CMP     (R0)+,(R1)+         ;POP POINTERS
003762  022700  001000                              CMP     #1000,R0            ;ALL DONE??
003766  001372                                      BNE     145$                ;BR IF NO.

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

003770  012706  001120                      .BEGIN: MOV     #STACK,SP           ;SET UP STACK
003774  106427  000200                              MTPS    #MASK               ;LOCK OUT INTERRUPTS
004000  005737  000042                              TST     @#42                ;IS PROGRAM UNDER MONITOR CONTROL
004004  001015                                      BNE     2$                  ;BR IF YES
004006  032777  000004  175270                      BIT     #BIT2,@SWR          ;CHECK FOR LOCK ON TEST
004014  001406                                      BEQ     1$                  ;BR IF NO LOCK DESIRED.
004016  104402  007606                              TYPE    ,MLOCK              ;TYPE LOCK SELECTED.
004022  012737  000240  004312                      MOV     #NOP,TTST           ;ADJUST SCOPE ROUTINE.
004030  000403                                      BR      2$                  ;CONTINUE ALONG.
004032  013737  004540  004312              1$:     MOV     BRW,TTST            ;PREPARE NORMAL SCOPE ROUTINE
004040  012737  010436  001252              2$:     MOV     #CYCLE,$LPADR       ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
004046  113737  001416  001415                      MOVB    SAVNO,SAVNUM        ;COPY ACTIVE DEVICES BEING TESTED
004054  104402  007477                              TYPE    ,MR                 ;TYPE "RUNNING"
004060  000177  175166                              JMP     @$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

004064                                    $EOP:
004064  000004                                    SCOPE
004066  005037  001262                            CLR     $ERRPC          ;CLEAR LAST ERROR PC
004072  105037  001247                            CLRB    $ERFLG          ;CLEAR ERROR FLAG
004076  104402  007453                            TYPE    ,MEPASS         ;TYPE END PASS
004102  104402  007635                            TYPE    ,MCSRX          ;TYPE CSR
004106  104412  004250                            CNVRT   ,XCSR           ;SHOW IT
004112  104402  007643                            TYPE    ,MVECX          ;TYPE VECTOR
004116  104412  004256                            CNVRT   ,XVEC           ;SHOW IT
004122  005237  001126                            INC     $PASS           ;RAISE PASS COUNT
004126  104402  007651                            TYPE    ,MPASSX         ;TYPE PASSES
004132  104412  004264                            CNVRT   ,XPASS          ;SHOW IT
004136  005337  001126                            DEC     $PASS           ;RESTORE PASS COUNT
004142  104402  007662                            TYPE    ,MERRX          ;TYPE ERRORS
004146  104412  004272                            CNVRT   ,XERR           ;SHOW IT
004152  005237  001130                            INC     $DEVCT          ;INC DEVCNT FOR APT
004156  105337  001415                            DECB    SAVNUM          ;ARE ALL DEVICES TESTED?
004162  001030                                    BNE     $DOAGN          ;BR IF NO.
004164  113737  001416  001415                    MOVB    SAVNO,SAVNUM    ;RESTORE THE COUNT
004172  005037  001354                            CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
004176  005237  001126                            INC     $PASS           ;;INCREMENT THE PASS NUMBER
004202  042737  100000  001126                    BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
004210  005327                                    DEC     (PC)+           ;;LOOP?
004212  000001                            $EOPCT: .WORD   1
004214  003013                                    BGT     $DOAGN          ;;YES
004216  012737                                    MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
004220  000001                            $ENDCT: .WORD   1
004222  004212                                    $EOPCT
004224  013700  000042                    $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
004230  001405                                    BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
004232  000005                                    RESET                   ;;CLEAR THE WORLD
004234  004710                            $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
004236  000240                                    NOP                     ;;SAVE ROOM
004240  000240                                    NOP                     ;;FOR
004242  000240                                    NOP                     ;;ACT11
004244                                    $DOAGN:
004244  000137                                    JMP     @(PC)+          ;;RETURN
004246  010436                            $RTNAD: .WORD   CYCLE

004250  000001                            XCSR:   1
004252     006     002                            .BYTE   6,2
004254  002010                                    DZVCSR
004256  000001                            XVEC:   1
004260     003     002                            .BYTE   3,2
```

```
004262  002040                                    DZVRIV
004264  000001                           XPASS:   1
004266     006      002                           .BYTE   6,2
004270  001126                                    $PASS
004272  000J01                           XERR:    1
004274     006      002                           .BYTE   6,2
004276  001256                                    $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

004300                                   $SCOPE:
004300  005037  001262                   .SCOPE:  CLR      $ERRPC             ;CLEAR LAST ERROR PC.
004304  022716  012172                            CMP      #TST1+2,(SP)       ;IS THIS THE SCOPE AT THE BEGINNING OF TST1?
004310  001413                                    BEQ      $XTSTR             ;IF SO, DON'T LOOP ON IT
004312  000406                           TTST:    BR       1$                 ;GOTO 1$    (IF LOCK SW02=1; THIS LOC =240)
004314  105777  174770                            TSTB     @$TKS              ;KEYBOARD DONE?
004320  100067                                    BPL      $OVER              ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
004322  017766  174764  177776                    MOV      @$TKB,-2(SP)       ;CLEAR DONE BIT
004330  032777  040000  174746           1$·      BIT      #BIT14,@SWR        ;;LOOP ON PRESENT TEST?
004336  001060                                    BNE      $OVER              ;;YES IF SW14=1
                                         ;#####START OF CODE FOR THE XOR TESTER#####
004340  000416                           $XTSTR:  BR       6$                 ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                              ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
004342  013746  000004                            MOV      @#ERRVEC,-(SP)     ;;SAVE THE CONTENTS OF THE ERROR VECTOR
004346  012737  004366  000004                    MOV      #5$,@#ERRVEC       ;;SET FOR TIMEOUT
004354  005737  177060                            TST      @#177060           ;;TIME OUT ON XOR?
004360  012637  000004                            MOV      (SP)+,@#ERRVEC     ;;RESTORE THE ERROR VECTOR
004364  CJ0436                                    BR       $SVLAD             ;;GO TO THE NEXT TEST
004366  022626                           5$:      CMP      (SP)+,(SP)+        ;;CLEAR THE STACK AFTER A TIME OUT
004370  012637  000004                            MOV      (SP)+,@#ERRVEC     ;;RESTORE THE ERROR VECTOR
004374  000441                                    BR       $OVER              ;;LUJP ON THE PRESENT TEST
004376                                   6$:;#####END OF CODE FOR THE XOR TESTER#####
004376  105737  001247                   2$:      TSTB     $ERFLG             ;;HAS AN ERROR OCCURRED?
004402  001404                                    BEQ      3$                 ;;BR IF NO
004404  105037  001247                   4$:      CLRB     $ERFLG             ;;ZERO THE ERROR FLAG
004410  005037  001354                            CLR      $TIMES             ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
004414  032777  004000  174662           3$:      BIT      #BIT11,@SWR        ;;INHIBIT ITERATIONS?
004422  001011                                    BNE      1$                 ;;BR IF YES
004424  005737  001126                            TST      $PASS              ;;IF FIRST PASS OF PROGRAM
004430  001406                                    BEQ      1$                 ;;       INHIBIT ITERATIONS
004432  005237  001250                            INC      $ICNT              ;;INCREMENT ITERATION COUNT
004436  023737  001354  001250                    CMP      $TIMES,$ICNT       ;;CHECK THE NUMBER OF ITERATIONS MADE
004444  002015                                    BGE      $OVER              ;;BR IF MORE ITERATION REQUIRED
```

```
004446  012737  000001  001250          1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
004454  013737  004542  001354                  MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
004462  105237  001246                  $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
004466  113737  001246  001124                  MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
004474  011637  001252                          MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
004500  013777  001246  174600          $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
004506  013716  001252                          MOV     $LPADR.(SP)     ;;FUDGE RETURN ADDRESS
004512  004737  006772                          JSR     PC,SERV.G       ;FIND UUT IF ↑G WAS TYPED
004516  105037  001424                          CLRB    MNTFLG          ;CLEAR THE MAINTENANCE BIT SETTER AFTER EACH TES
004522  005737  001372                          TST     MODE            ;HAS THE MODE BEEN CHANGED?
004526  001003                                  BNE     4$              ;IF NOT INTERNAL , (^ DO A TEST
004530  112737  000010  001424                  MOVB    #MAINT,MNTFLG   ;IF INTERNAL MODE NOW, SET THE MAINTENANCE BIT
004536  000002                          4$:     RTI                     ;GO DO THE TEST
004540  000406                          BRW:    406
004542  000005                          $MXCNT: 5                       ;;MAX. NUMBER OF ITERATIONS

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

004544  032777  001000  174532          .SCOP1: BIT     #SW09,@SWR      ;IS SW09=1(SET)?
004552  001405                                  BEQ     1$              ;BR IF NOT SET.
004554  005737  001364                          TST     LOCK            ;IS THERE A TIGHT LOOP SPECIFIED?
004560  001402                                  BEQ     1$              ;IF NO, RETURN
004562  013716  001364                          MOV     LOCK,(SP)       ;IF YES, GOTO THE ADDRESS IN LOCK.
004566  000002                          1$:     RTI                     ;GO BACK.

004570  032777  010000  174506          .TYPE:  BIT     #SW12,@SWR      ;INHIBIT ALL PRINTOUT??
004576  001403                                  BEQ     $TYPE           ;IF NOT, GO TYPE
004600  062716  000002                          ADD     #2,(SP)         ;SKIP OVER MESSAGE POINTER
004604  000002                                  RTI                     ;RETURN TO WHERE PROCEDURE WAS INVOKED
                                        .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
                                        ;*

004606  105737  001323                  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
004612  100002                                  BPL     1$              ;;BR IF YES
004614  000000                                  HALT                    ;;HALT HERE IF NO TERMINAL
004616  000430                                  BR      3$              ;;LEAVE
004620  010046                          1$:     MOV     R0,-(SP)        ;;SAVE R0
004622  017600  000002                          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
004626  122737  000001  001140                  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
004634  001011                                  BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
004636  132737  000100  001141                  BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
```

```
004644  001405                              BEQ     62$                     ;;NO,GO CHECK FOR CONSOLE
004646  010037  004656                      MOV     R0,61$                  ;;SETUP MESSAGE ADDRESS FOR APT
004652  004737  005076                      JSR     PC,$ATY3                ;;SPOOL MESSAGE TO APT
004656  000000                      61$:    .WORD   0                       ;;MESSAGE ADDRESS
004660  132737  000040  001141      62$:    BITB    #APTCSUP,$ENVM          ;;APT CONSOLE SUPPRESSED
004666  001003                              BNE     60$                     ;;YES,SKIP TYPE OUT
004670  112046                      2$:     MOVB    (R0)+,-(SP)             ;;PUSH CHARACTER TO BE TYPED ONTO STACK
004672  001005                              BNE     4$                      ;;BR IF IT ISN'T THE TERMINATOR
004674  005726                              TST     (SP)+                   ;;IF TERMINATOR POP IT OFF THE STACK
004676  012600                      60$:    MOV     (SP)+,R0                ;;RESTORE R0
004700  062716  000002              3$:     ADD     #2,(SP)                 ;;ADJUST RETURN PC
004704  000002                              RTI                             ;;RETURN
004706  122716  000011              4$:     CMPB    #HT,(SP)                ;;BRANCH IF <HT>
004712  001430                              BEQ     8$
004714  122716  000200                      CMPB    #CRLF,(SP)              ;;BRANCH IF NOT <CRLF>
004720  001006                              BNE     5$
004722  005726                              TST     (SP)+                   ;;POP  <CR><LF> EQUIV
004724  104402                              TYPE                            ;;TYPE A CR AND LF
004726  001357                              $CRLF
004730  105037  005064                      CLRB    $CHARCNT                ;;CLEAR CHARACTER COUNT
004734  000755                              BR      2$                      ;;GET NEXT CHARACTER
004736  004737  005020              5$:     JSR     PC,$TYPEC               ;;GO TYPE THIS CHARACTER
004742  123726  001322              6$:     CMPB    $FILLC,(SP)+            ;;IS IT TIME FOR FILLER CHARS.?
004746  001350                              BNE     2$                      ;;IF NO GO GET NEXT CHAR.
004750  013746  001320                      MOV     $NULL,-(SP)             ;;GET # OF FILLER CHARS. NEEDED
                                                                            ;;AND THE NULL CHAR.
004754  105366  000001              7$:     DECB    1(SP)                   ;;DOES A NULL NEED TO BE TYPED?
004760  002770                              BLT     6$                      ;;BR IF NO--GO POP THE NULL OFF OF STACK
004762  004737  005020                      JSR     PC,$TYPEC               ;;GO TYPE A NULL
004766  105337  005064                      DECB    $CHARCNT                ;;DO NOT COUNT AS A COUNT
004772  000770                              BR      7$                      ;;LOOP

                                    ;HORIZONTAL TAB PROCESSOR

004774  112716  000040              8$:     MOVB    #' ,(SP)                ;;REPLACE TAB WITH SPACE
005000  004737  005020              9$:     JSR     PC,$TYPEC               ;;TYPE A SPACE
005004  132737  000007  005064              BITB    #7,$CHARCNT             ;;BRANCH IF NOT AT
005012  001372                              BNE     9$                      ;;TAB STOP
005014  005726                              TST     (SP)+                   ;;POP SPACE OFF STACK
005016  000724                              BR      2$                      ;;GET NEXT CHARACTER
005020  105777  174270              $TYPEC: TSTB    @$TPS                   ;;WAIT UNTIL PRINTER IS READY
005024  100375                              BPL     $TYPEC
005026  116677  000002  174262              MOVB    2(SP),@$TPB             ;;LOAD CHAR TO BE TYPED INTO DATA REG.
005034  122766  000015  000002              CMPB    #CR,2(SP)               ;;IS CHARACTER A CARRIAGE RETURN?
005042  001003                              BNE     1$                      ;;BRANCH IF NO
005044  105037  005064                      CLRB    $CHARCNT                ;;YES--CLEAR CHARACTER COUNT
005050  000406                              BR      $TYPEX                  ;;EXIT
005052  122766  000012  000002      1$:     CMPB    #LF,2(SP)               ;;IS CHARACTER A LINE FEED?
005060  001402                              BEQ     $TYPEX                  ;;BRANCH IF YES
005062  105227                              INCB    (PC)+                   ;;COUNT THE CHARACTER
005064  000000                      $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
005066  000207                      $TYPEX: RTS     PC

                                    .SBTTL  APT COMMUNICATIONS ROUTINE

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

```
005070  112737  000001  005334          $ATY1:  MOVB    #1,$FFLG          ;;TO REPORT FATAL ERROR
005076  112737  000001  005332          $ATY3:  MOVB    #1,$MFLG          ;;TO TYPE A MESSAGE
005104  000403                                  BR      $ATYC
005106  112737  000001  005334          $ATY4:  MOVB    #1,$FFLG          ;;TO ONLY REPORT FATAL ERROR
005114                                  $ATYC:
005114  010046                                  MOV     R0,-(SP)          ;;PUSH R0 ON STACK
005116  010146                                  MOV     R1,-(SP)          ;;PUSH R1 ON STACK
005120  105737  005332                          TSTB    $MFLG             ;;SHOULD TYPE A MESSAGE?
005124  001450                                  BEQ     5$                ;;IF NOT:  BR
005126  122737  000001  001140                  CMPB    #APTENV,$ENV      ;;OPERATING UNDER APT?
005134  001031                                  BNE     3$                ;;IF NOT:  BR
005136  132737  000100  001141                  BITB    #APTSPOOL,$ENVM   ;;SHOULD SPOOL MESSAGES?
005144  001425                                  BEQ     3$                ;;IF NOT:  BR
005146  017600  000004                          MOV     @4(SP),R0         ;;GET MESSAGE ADDR.
005152  062766  000002  000004                  ADD     #2,4(SP)
                                                                              ;;BUMP RETURN ADDR.
005160  005737  001120                  1$:     TST     $MSGTYPE          ;;SEE IF DONE W/ LAST XMISSION?
005164  001375                                  BNE     1$                ;;IF NOT:  WAIT
005166  010037  001134                          MOV     R0,$MSGAD         ;;PUT ADDR IN MAILBOX
005172  105720                          2$:     TSTB    (R0)+             ;;FIND END OF MESSAGE
005174  001376                                  BNE     2$
005176  163700  001134                          SUB     $MSGAD,R0         ;;SUB START OF MESSAGE
005202  006200                                  ASR     R0                ;;GET MESSAGE LNGTH IN WORDS
005204  010037  001136                          MOV     R0,$MSGLGT        ;;PUT LENGTH IN MAILBOX
005210  012737  000004  001120                  MOV     #4,$MSGTYPE       ;;TELL APT TO TAKE MSG.
005216  000413                                  BR      5$
005220  017637  000004  005244          3$:     MOV     @4(SP),4$         ;;PUT MSG ADDR IN JSR LINKAGE
005226  062766  000002  000004                  ADD     #2,4(SP)
                                                                              ;;BUMP RETURN ADDRESS
005234  013746  177776                          MOV     177776,-(SP)      ;;PUSH 177776 ON STACK
005240  004737  004606                          JSR     PC,$TYPE          ;;CALL TYPE MACRO
005244  000000                          4$:     .WORD   0
005246                                  5$:
005246  105737  005334                  10$:    TSTB    $FFLG             ;;SHOULD REPORT FATAL ERROR?
005252  001416                                  BEQ     12$               ;;IF NOT:  BR
005254  005737  001140                          TST     $ENV              ;;RUNNING UNDER APT?
005260  001413                                  BEQ     12$               ;;IF NOT:  BR
005262  005737  001120                  11$:    TST     $MSGTYPE          ;;FINISHED LAST MESSAGE?
005266  001375                                  BNE     11$               ;;IF NOT:  WAIT
005270  017637  000004  001122                  MOV     @4(SP),$FATAL     ;;GET ERROR #
005276  062766  000002  000004                  ADD     #2,4(SP)
                                                                              ;;BUMP RETURN ADDR.
005304  005237  001120                          INC     $MSGTYPE          ;;TELL APT TO TAKE ERROR
005310  105037  005334                  12$:    CLRB    $FFLG             ;;CLEAR FATAL FLAG
005314  105037  005333                          CLRB    $LFLG             ;;CLEAR LOG FLAG
005320  105037  005332                          CLRB    $MFLG             ;;CLEAR MESSAGE FLAG
005324  012601                                  MOV     (SP)+,R1          ;;POP STACK INTO R1
005326  012600                                  MOV     (SP)+,R0          ;;POP STACK INTO R0
005330  000207                                  RTS     PC                ;;RETURN
005332     000                          $MFLG:  .BYTE   0                 ;;MESSG. FLAG
005333     000                          $LFLG:  .BYTE   0                 ;;LOG FLAG
005334     000                          $FFLG:  .BYTE   0                 ;;FATAL FLAG
        005336                                  .EVEN
        000200                          APTSIZE=200
        000001                          APTENV=001
        000100                          APTSPOOL=100
        000040                          APTCSUP=040

                                                ;STRING INPUT ROUTINE
```

```
                                                  ;---------------------------
                                                  
005336  010346                          .INSTR:  MOV     R3,-(SP)          ;SAVE R3 ON STACK
005340  010446                                   MOV     R4,-(SP)          ;SAVE R4 ON STACK
005342  017637  000004  005360                   MOV     @4(SP),.MSG       ;GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
005350  062766  000002  000004                   ADD     #2,4(SP)          ;POINT TO INSTRUCTION AFTER ADDRESS POINTER
005356  104402                          .INST1:  TYPE                      ;PRINT THE MESSAGE
005360  000000                          .MSG:    0                         ;MESSAGE IS POINTED TO FROM HERE
005362  012704  010270                           MOV     #INBUF,R4         ;POINT R4 TO THE INPUT BUFFER
005366  012703  000007                           MOV     #7,R3             ;SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
005372  105777  173712                  1$:      TSTB    @$TKS             ;HAS A CHARACTER BEEN RECEIVED?
005376  100375                                   BPL     1$                ;IF NO, KEEP WAITING FOR IT
005400  117714  173706                           MOVB    @$TKB,(R4)        ;IF YES, SAVE IT IN THE INPUT BUFFER
005404  142714  000200                           BICB    #200,(R4)         ;KEEP ONLY THE 7-BIT ASCII INFORMATION
005410  122427  000015                           CMPB    (R4)+,#15         ;IS THIS CHARACTER A LINE FEED?
005414  001417                                   BEQ     INSTR2            ;IF SO, TERMINATE THE INPUT SEQUENCE
005416  105777  173672                  2$:      TSTB    @$TPS             ;IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
005422  100375                                   BPL     2$                ;IF WE CAN'T, WAIT UNTIL WE CAN
005424  017777  173662  173664                   MOV     @$TKB,@$TPB       ;ECHO THE CHARACTER BACK
005432  005303                                   DEC     R3                ;REDUCE THE NUMBER OF CHARACTERS RECEIVED
005434  001356                                   BNE     1$                ;IF WE DON'T HAVE 7, GO GET SOME MORE
005436  012604                                   MOV     (SP)+,R4          ;IF WE HAVE 7, RESTORE R4
005440  012603                                   MOV     (SP)+,R3          ;RESTORE R3
005442  010346                          .INSTE:  MOV     R3,-(SP)          ;SAVE R3 ON THE STACK
005444  010446                                   MOV     R4,-(SP)          ;SAVE R4 ON THE STACK
005446  104402  001356                           TYPE    ,$QUES            ;PRINT A QUESTION MARK... WHAT'S GOING ON?
005452  000741                                   BR      .INST1            ;GO PRINT THE MESSAGE AGAIN
005454  012604                          INSTR2:  MOV     (SP)+,R4          ;RESTORE R4
005456  012603                                   MOV     (SP)+,R3          ;RESTORE R3
005460  000002                                   RTI                       ;RETURN TO THE MAIN PROCEDURE

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

005462  010546                          .PARAM:  MOV     R5,-(SP)          ;SAVE R5 ON THE STACK
005464  010446                                   MOV     R4,-(SP)          ;SAVE R4 ON THE STACK
005466  016605  000004                           MOV     4(SP),R5          ;GET THE SETUP INFORMATION POINTER
005472  012537  005652                           MOV     (R5)+,LOLIM       ;SET THE LOW LIMIT FOR THE INPUT
005476  012537  005654                           MOV     (R5)+,HILIM       ;SET THE HIGH LIMIT FOR THE INPUT
005502  012537  005656                           MOV     (R5)+,DEVADR      ;SAVE THE ADDRESS WHERE THE RESULT WILL BE STORE
005506  112537  005660                           MOVB    (R5)+,LOBITS      ;GET THE MASK OF THE INCORRECT BITS
005512  112537  005661                           MOVB    (R5)+,ADRCNT      ;GET THE COUNT OF ITEMS TO BE STORED
005516  010566  000004                           MOV     R5,4(SP)          ;POINT TO WHERE MAIN LINE PROGRAM WILL RESUME
005522  005005                          PARAM1:  CLR     R5                ;INITIALIZE THE ASCII TO OCTAL RESULT WORD
005524  012704  010270                           MOV     #INBUF,R4         ;POINT TO THE INPUT BUFFER
005530  122714  000015                           CMPB    #15,(R4)          ;IS THIS CHARACTER A CARRIAGE RETURN?
005534  001420                                   BEQ     PARERR            ;IF SO, PRINT THE MESSAGE AGAIN
005536  121427  000060                  1$:      CMPB    (R4),#60          ;IS THIS CHARACTER BELOW THE NUMERIC RANGE?
005542  002415                                   BLT     PARERR            ;IF SO, GO PRINT THE MESSAGE AGAIN
005544  121427  000067                           CMPB    (R4),#67          ;IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
005550  003012                                   BGT     PARERR            ;IF SO, GO PRINT THE MESSAGE AGAIN
005552  142714  000060                           BICB    #60,(R4)          ;ISOLATE THE NUMBER THE CHARACTER REPRESENTS
005556  152405                                   BISB    (R4)+,R5          ;CONCATENATE THESE BITS TO THE ALREADY EXISTING
005560  122714  000015                           CMPB    #15,(R4)          ;IS THE NEXT CHARACTER A CARRIAGE RETURN?
005564  001406                                   BEQ     LIMITS            ;IF SO, GO SEE IF NUMBER IS WITHIN LIMITS
005566  006305                                   ASL     R5                ;CLEAR BIT POSITION 0, MOVE EXISTING STRING TO L
```

```
005570  006305                                 ASL     R5              ;CLEAR POSITION 1, MOVE STRING TO LEFT AGAIN
005572  006305                                 ASL     R5              ;MOVE THE STRING ONE MORE TIME TO MAKE ROOM FOR
                                                                       ;NEXT THREE BITS
005574  000760                                 BR      1$              ;GO GET THE NEXT CHARACTER
005576  104404                        PARERR:  INSTER                  ;THERE WAS AN ERROR. . GO PRINT MESSAGE AGAIN
005600  000750                                 BR      PARAM1          ;TRY GETTING THE PARAMETERS AGAIN

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

005602  020537  005654                LIMITS:  CMP     R5,HILIM        ;DOES RESULT EXCEED ITS MAXIMUM CORRECT VALUE?
005606  101373                                 BHI     PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
005610  020537  005652                         CMP     R5,LOLIM        ;IS THE RESULT LOWER THAN ALLOWED?
005614  103770                                 BLO     PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
005616  133705  005660                         BITB    LOBITS,R5       ;ARE ANY INCORRECT BITS SET IN THE RESULT?
005622  001365                                 BNE     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN

                                               ;STORE NUMBER AT SPECIFIED ADDRESS

005624  013704  005656                         MOV     DEVADR,R4       ;POINT TO THE LOCATION WHERE THE RESULT WILL BE
005630  010524                        1$:      MOV     R5,(R4)+        ;STORE THE RESULT
005632  062705  000002                         ADD     #2,R5           ;CALCULATE THE NEXT DATUM
005636  105337  005661                         DECB    ADRCNT          ;REDUCE COUNT OF STORED RESULTS. IS IT EXCEEDED?
005642  001372                                 BNE     1$              ;IF NOT, GO STORE THE NEXT DATUM
005644  012604                                 MOV     (SP)+,R4        ;RESTORE R4
005646  012605                                 MOV     (SP)+,R5        ;RESTORE R5
005650  000002                                 RTI                     ;RETURN TO THE MAIN PROGRAM

005652  000000                        LOLIM:   0                       ;LOWEST ACCEPTABLE VALUE
005654  000000                        HILIM:   0                       ;HIGHEST ACCEPTABLE
005656  000000                        DEVADR:  0                       ;LOCATION WHERE RESULT WILL BE STORED
005660     000                        LOBITS:  .BYTE   0               ;INCORRECT BITS MASK
005661     000                        ADRCNT:  .BYTE   0               ;COUNT OF ITEMS TO BE STORED

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

005662  016637  000004  001404        .SAV05:  MOV     4(SP),SAVPC     ;SAVE R7 (PC)

                                               ;SAVE R0-R5

005670  010537  001340                SV05:    MOV     R5,$REG5        ;SAVE R5
005674  010437  001336                         MOV     R4,$REG4        ;SAVE R4
005700  010337  001334                         MOV     R3,$REG3        ;SAVE R3
005704  010237  001332                         MOV     R2,$REG2        ;SAVE R2
005710  010137  001330                         MOV     R1,$REG1        ;SAVE R1
005714  010037  001326                         MOV     R0,$REG0        ;SAVE R0
005720  000002                                 RTI                     ;LEAVE.

                                               ;RESTORE R0-R5

005722  013700  001326                .RES05:  MOV     $REG0,R0        ;RESTORE R0
005726  013701  001330                         MOV     $REG1,R1        ;RESTORE R1
005732  013702  001332                         MOV     $REG2,R2        ;RESTORE R2
005736  013703  001334                         MOV     $REG3,R3        ;RESTORE R3
005742  013704  001336                         MOV     $REG4,R4        ;RESTORE R4
```

```
005746  013705  001340                            MOV     $REGS,R5          ;RESTORE R5
005752  000002                                    RTI                       ;LEAVE

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

005754  104402  001357                   .CONVR:  TYPE    ,$CRLF            ;PRINT A CARRIAGE RETURN
005760  010046                           .CNVRT:  MOV     R0,-(SP)          ;SAVE R0
005762  010146                                    MOV     R1,-(SP)          ;SAVE R1
005764  010346                                    MOV     R3,-(SP)          ;SAVE R3
005766  010446                                    MOV     R4,-(SP)          ;SAVE R4
005770  010546                                    MOV     R5,-(SP)          ;SAVE R5
005772  017601  000012                            MOV     @12(SP),R1        ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
005776  062766  000002  000012                    ADD     #2,12(SP)         ;POINT TO WHERE MAIN PROGRAM WILL RESUME
006004  012137  006130                            MOV     (R1)+,WRDCNT      ;GET NUMBER OF WORDS TO BE PRINTED
006010  112105                           1$:      MOVB    (R1)+,R5          ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
006012  112100                                    MOVB    (R1)+,R0          ;GET THE NUMBER OF SPACES TO PRINT
006014  013104                                    MOV     @(R1)+,R4         ;COPY THE WORD TO BE CONVERTED
006016  110537  006132                            MOVB    R5,CHRCNT         ;COPY THE CHARACTER COUNT
006022  010403                           3$:      MOV     R4,R3             ;COPY THE ARGUMENT WORD AGAIN
006024  042703  177770                            BIC     #↑C<7>,R3         ;ISOLATE THRFE BITS TO BE TREATED AS A CHARACTER
006030  062703  000060                            ADD     #060,R3           ;MAKE AN ASCII CHARACTER OUT OF THEM
006034  110346                                    MOVB    R3,-(SP)          ;SAVE THAT CHARACTER
006036  006004                                    ROR     R4                ;MOVE THE NEXT THREE BITS INTO PLACE
006040  006204                                    ASR     R4                ;MOVE THEM AGAIN
006042  006204                                    ASR     R4                ;AND FINALLY A THIRD TIME
006044  005305                                    DEC     R5                ;REDUCE CHARACTER COUNT.ARE ALL CHARACTERS
                                                                            ;BUILT?
006046  001365                                    BNE     3$                ;IF NO, GO BUILD THE NEXT ONE.
006050  012703  010374                            MOV     #MDATA,R3         ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
006054  112623                           4$:      MOVB    (SP)+,(R3)+       ;STORE THE CHARACTER, STARTING WITH THE MOST
006056  105337  006132                            DECB    CHRCNT            ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
006062  001374                                    BNE     4$                ;IF NO, GO TRANSFER ANOTHER
006064  105700                                    TSTB    R0                ;ARE ANY SPACES TO BE PRINTED?
006066  001404                                    BEQ     6$                ;IF NO, DON'T SET UP ANY
006070  112723  000040                   5$:      MOVB    #040,(R3)+        ;ADD A SPACE TO THE OUTPUT BUFFER
006074  105300                                    DECB    R0                ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
006076  001374                                    BNE     5$                ;IF YES, GO ADD ANOTHER SPACE
006100  105013                           6$:      CLRB    (R3)              ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
006102  104402  010374                            TYPE    ,MDATA            ;PRINT THE STRING WE JUST BUILT
006106  005337  006130                            DEC     WRDCNT            ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
006112  001336                                    BNE     1$                ;IF YES, GO CONVERT THEM
006114  012605                                    MOV     (SP)+,R5          ;RESTORE R5
006116  012604                                    MOV     (SP)+,R4          ;RESTORE R4
006120  012603                                    MOV     (SP)+,R3          ;RESTORE R3
006122  012601                                    MOV     (SP)+,R1          ;RESTORE R1
006124  012600                                    MOV     (SP)+,R0          ;RESTORE R0
006126  000002                                    RTI                       ;RETURN TO THE MAIN PROGRAM
006130  000000                           WRDCNT:  0
006132     000                           CHRCNT:  .BYTE                     ;NUMBER OF CHARACTERS TO PRINT
006133     000                           SPACNT:  .BYTE   0                 ;NUMBER OF SPACES TO PRINT

006134  000000                           BINWRD:  0


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

006136  010046                          .TRPSR:   MOV     R0,-(SP)        ;SAVE R0. USE R0 TO FIND TRAP ROUTINE
006140  016600  000002                            MOV     2(SP),R0        ;GET TRAP ADDRESS
006144  005740                                    TST     -(R0)           ;GET TPAP
006146  111000                                    MOVB    (R0),R0         ;GET RIGHT BYTE OF TRAP(TRAP OFFSET)
006150  006300                                    ASL     R0              ;POSITION OFFSET FOR TABLE INDEXING
006152  016000  001742                            MOV     .TRPTAB(R0),R0  ;PLACE INDEXED ADDRESS OF TABLE IN R0
006156  000200                                    RTS     R0              ;TRANSFER TO THAT ADDRESS AND RESTORE OLD R0

                                                  ;DEVICE CLEAR ROUTINE
                                                  ;ISSUE A DEVICE CLEAR
                                                  ;--------------------
006160                                  .DEVICE.CLR:
006160  052777  000020  173622                    BIS     #DCLR,@DZVCSR   ;SET DCLR
006166  032777  000020  173614          1$:       BIT     #DCLR,@DZVCSR   ;DID IT CLEAR?
006174  001374                                    BNE     1$              ;BR IF NO
006176  000002                                    RTI                     ;EXIT ROUTINE

                                                  ;ROUTINE TO HANDLE MAINTENANCE BIT SETTING WITH DEVICE CLEAR
                                                  ;-----------------------------------------------------------
006200  104413                          .DCLASM:  DEVICE.CLR              ;ISSUE A DEVICE CLEAR
006202  153777  001424  173600                    BISB    MNTFLG,@DZVCSR  ;LOAD THE MAINTENANCE BIT IF IT IS I MODE
006210  000002                                    RTI                     ;RETURN TO CALLING ROUTINE

006212                                  .DELAY:
006212  010046                                    MOV     R0,-(SP)        ;SAVE R0
006214  013700  006230                            MOV     DLYCNT,R0       ;SET COUNT
006220  005300                          1$:       DEC     R0              ;DELAY
006222  001376                                    BNE     1$              ;
006224  012600                                    MOV     (SP)+,R0        ;RESTORE R0
006226  000002                                    RTI                     ;LEAVE ROUTINE
006230  000001                          DLYCNT:   .WORD     1             ;PATCHABLE LOC FOR MORE TIME

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

006232  013716  001362                  .ADVANCE: MOV     NEXT,(SP)       ;CRUNCH STACK WITH ADDRESS OF SCOPE CALL
006236  005037  001364                            CLR     LOCK            ;RESET TIGHT LOOP ADDRESS
006242  000002                                    RTI                     ;CHECK TO SEE IF OLD TEST GETS REPEATED

                                                  ;ROUTINE TO SHIFT LINE POINTER
                                                  ;AND SWITCH TESTS IF NECESSARY
                                                  ;-----------------------------
006244  106302                          .SHIFT:   ASLB    R2              ;POINT TO THE NEXT LINE
006246  032702  000020                            BIT     #BIT4,R2        ;HAVE WE PASSED ALL LINE POINTERS?
006252  001402                                    BEQ     1$              ;IF NOT, RETURN TO THE TEST
006254  022626                                    POP2SP                  ;REMOVE THE TRAP CALL FROM THE STACK
006256  104400                                    ADVANCE                 ;GO TO THE NEXT TEST
006260  000002                          1$:       RTI                     ;RETURN TO THE PRESENT TEST

```

```
                                                  ;LINE PARAMETER REGISTER SETUP ROUTINE

        006262  010146                            .LPRSET:MOV     R1,-(SP)        ;SAVE CONTENTS OF R1
        006264  010246                                    MOV     R2,-(SP)        ;SAVE CONTENTS OF R2
        006266  013701  001370                            MOV     PAR,R1          ;MOVE DEFAULT PARAM. INTO R1
        006272  012702  000001                            MOV     #1,R2   ;INIT. FOR LINE 1
        006276  010177  173516                    1$:     MOV     R1,@DZVLPR      ;LOAD PARAM. REGISTER
        006302  005201                                    INC     R1              ;SET R1 FOR NEXT LINE
        006304  106302                                    ASLB    R2              ;SET R2 FOR NEXT LINE
        006306  032702  000020                            BIT     #BIT4,R2        ;ALL LINES DONE?
        006312  001771                                    BEQ     1$              ;IF NO LOAD NEXT LINE
        006314  012602                                    MOV     (SP)+,R2        ;RELOAD R2
        006316  012601                                    MOV     (SP)+,R1        ;RELOAD R1
        006320  000002                                    RTI                     ;RETURN

                                                  ;ROUTINE TO ZERO DATA BUFFER

        006322  010046                            .BUFSET:MOV     R0,-(SP)        ;SAVE CONTENTS OF R0
        006324  012700  001426                            MOV     #TD0,R0         ;SET R0 TO TOP OF BUFFER
        006330  005020                            1$:     CLR     (R0)+           ;CLEAR BUFFER LOCATION
        006332  022700  001446                            CMP     #STOP,R0        ;IS BUFFER ALL CLEARED
        006336  001374                                    BNE     1$              ;IF NOT CLEAR NEXT LOCATION
        006340  012600                                    MOV     (SP)+,R0        ;RELOAD R0
        006342  000002                                    RTI                     ;RETURN

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

        006344  004737  006772                    $ERROR: JSR     PC,SERV.G       ;FIND OUT IF <↑G> WAS HIT
        006350  032777  010000  172726                    BIT     #SW12,@SWR      ;BELL ON ERROR?
        006356  001406                                    BEQ     XBX             ;BR IF NO BELL
        006360  105777  172730                            TSTB    @$TPS           ;TTY READY.
        006364  100003                                    BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
        006366  112777  000207  172722                    MOVB    #207,@$TPB      ;PUSH A BELL AT THE TTY.
        006374  032777  020C00  172702            XBX:    BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
        006402  001113                                    BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
        006404  021637  001262                            CMP     (SP),$ERRPC     ;WAS THIS ERROR FOUND LAST TIME?
        006410  001404                                    BEQ     1$              ;BR IF YES
        006412  011637  001262                            MOV     (SP),$ERRPC     ;RECORD BEING HERE
        006416  105037  001247                            CLRB    $ERFLG          ;PREPARE HEADER
        006422  104407                            1$:     SAVOS                   ;SAVE ALL PROC REGISTERS
        006424  011605                                    MOV     (SP),R5         ;GET THE PC OF ERROR
        006426  162705  000002                            SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
        006432  011504                                    MOV     (R5),R4         ;GET ERROR INSTRUCTION
        006434  110437  001260                            MOVB    R4,$ITEMB       ;COPY TEST NUMBER FOR APT HANDLING
        006440  006304                                    ASL     R4              ;MULT BY TWO
        006442  061504                                    ADD     (R5),R4         ;DOUBLE IT
        006444  006304                                    ASL     R4              ;MULT AGAIN
        006446  042704  177001                            BIC     #177001,R4      ;CLEAR JUNK
        006452  062704  016200                            ADD     #.ERRTAB,R4     ;GET POINTER
        006456  012437  006602                            MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
        006462  012437  006614                            MOV     (R4)+,DATAHD    ;GET DATA HEADRER
        006466  011437  006626                            MOV     (R4),DATABP     ;GET DATA TABLE
        006472  105737  001247                            TSTB    $ERFLG          ;TYPE HEADER
        006476  001403                                    BEQ     TYPMSG          ;BR IF YES
        006500  005737  006626                            TST     DATABP          ;DOES DATA TABLE EXIST?
```

```
006504  001044                                BNE     TYPDAT          ;BR IF YES.
006506  104402  001357              TYPMSG:   TYPE    ,$CRLF          ;TYPE A CARRIAGE RETURN
006512  104402  001357                        TYPE    ,$CRLF          ;AND TYPE ANOTHER
006516  005737  001364                        TST     LOCK
006522  001402                                BEQ     1$
006524  104402  007705                        TYPE    ,MASTEK
006530  104402  007673              1$:       TYPE    ,MTSTN
006534  104412  006764                        CNVRT   ,XTSTN          ;SHOW IT
006540  104402  007765                        TYPE    ,MERRPC         ;TYPE PC.
006544  104412  006756                        CNVRT   ,ERTABO         ;SHOW IT
006550  104402  007635                        TYPE    ,MCSRX
006554  104412  004250                        CNVRT   ,XCSR
006560  104402  001357                        TYPE    ,$CRLF          ;GIVE A CR/LF
006564  112737  177777  001247                MOVB    #-1,$ERFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
006572  005737  006602                        TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
006576  001402                                BEQ     WTBS.FM         ;BR IF NO.
006600  104402                                TYPE                    ;TYPE
006602  000000                      ERRMSG:   0                       ;    ERROR MESSAGE
006604                              WTBS.FM:
006604  005737  006614                        TST     DATAHD          ;DATA HEADER?
006610  001402                                BEQ     TYPDAT          ;BR IF NO
006612  104402                                TYPE                    ;TYPE
006614  000000                      DATAHD:   0                       ;    DATA HEADER
006616  005737  006626              TYPDAT:   TST     DATABP          ;DATA TABLE?
006622  001402                                BEQ     RESREG          ;BR IF NO.
006624  104411                                CONVRT                  ;SHOW
006626  000000                      DATABP:   0                       ;    DATA TABLE
006630  104410                      RESREG:   RESOS                   ;RESTORE PROC REGISTERS
006632  122737  000001  001140      HALTS:    CMPB    #APTENV,$ENV    ;IS APT RUNNING?
006640  001007                                BNE     15$             ;SKIP APT CALL IF NOT
006642  113737  001260  006654                MOVB    $ITEMB,5$       ;COPY ERROR NUMBER
006650  004737  005106                        JSR     PC,$ATY4        ;CALL APT SERVICE
006654  000000                      5$:       .WORD   0               ;ERROR NUMBER STUCK HERE
006656  000777                      10$:      BR      10$             ;LOCK UP HERE
006660  022737  004234  000042      15$:      CMP     #$ENDAD,@#42    ;CHECK TO SEE IF IN ACT-11 MODE
006666  001403                                BEQ     20$             ;IF SO, HANDLE ACCORDINGLY
006670  005777  172410                        TST     @SWR            ;HALT ON ERROR?
006674  100004                                BPL     EXITER          ;BR IF NO HALT ON ERROR
006676  016677  000002  172402      20$:      MOV     2(SP),@DISPLAY  ;SHOW ERROR PC IN DATA DISPLAY
006704  000000                                HALT                    ;HALT
006706  005237  001256              EXITER:   INC     $ERTTL          ;UPDATE ERROR COUNT
006712  004737  006772                        JSR     PC,SERV.G       ;FIND OUT IF ↑G WAS TYPED
006716  032777  000400  172360                BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
006724  001007                                BNE     1$              ;BR IF YES
006726  032777  002000  172350                BIT     #SW10,@SWR      ;GOTO NEXT TEST?
006734  001407                                BEQ     2$              ;BR IF NO
006736  013737  001362  001252                MOV     NEXT,$LPADR     ;SET FOR NEXT TEST
006744  012706  001120              1$:       MOV     #STACK,SP       ;RESET SP
006750  000177  172276                        JMP     @$LPADR         ;GOTO SPECIFIED TEST
006754  000002                      2$:       RTI                     ;RETURN
006756  000001                      ERTABO:   1
006760     006     002                        .BYTE   6,2
006762  001404                                SAVPC
006764  000001                      XTSTN:    1
006766     002     002                        .BYTE   2,2
006770  001246                                $TSTNM
```

```
006772 017746 172314               SERV.G: MOV     @$TKB,-(SP)       ;OTHERWISE, GET THE LAST CHARACTER TYPED
006776 042716 000200                       BIC     #BIT7,(SP)        ;STRIP PARITY(EIGHTH) BIT
007002 122726 000007                       CMPB    #7,(SP)+          ;IS IT ↑G?
007006 001076                              BNE     6$                ;IF NOT, IGNORE INPUT
007010 032777 004000 172272                BIT     #4000,@$TKS       ;RX BUSY?
007016 001365                              BNE     SERV.G            ;BR IF YES
007020 017737 172260 007226                MOV     @SWR,90$          ;SAVE .SWR).
007026 104402 007206               1$:     TYPE    ,89$              ;TYPE HEADER FOR OLD SWITCH REGISTER
007032 104412 007220                       CNVRT   ,88$              ;TYPE THE NUMBER ITSELF
007036 104402 007230                       TYPE    ,91$              ;AFTER HAVING CONVERTED IT TO ASCII
007042 105037 007234                       CLRB    92$               ;CLEAR SWR CHANGE FLAG
007046 005077 172232                       CLR     @SWR              ;CLEAR THE SOFTWARE SWITCH REGISTER
007052 105777 172232               3$:     TSTB    @$TKS             ;WAIT FOR DONE.
007056 100375                              BPL     3$                ;CONTINUE WAITING FOR IT
007060 017746 172226                       MOV     @$TKB,-(SP)       ;PUT THE CHARACTER ON THE STACK
00,064 042716 000200                       BIC     #BIT7,(SP)        ;STRIP PARITY BIT
007070 122726 000015                       CMPB    #15,(SP)+         ;IS IT THE CARRIAGE RETURN CHAR?
007074 001433                              BEQ     4$                ;IF SO, GO PRINT CRLF
007076 105777 172212               2$:     TSTB    @$TPS             ;IS THE OUTPUT BUFFER AVAILABLE
007102 100375                              BPL     2$                ;IF NOT, WAIT FOR IT TO BE READY
007104 105237 007234                       INCB    92$               ;INDICATE THAT THE SWR :AS CHANGED
007110 C:4677 172202                       MOV     -(SP),@$TPB       ;PLACE THE CHARACTE: THERE(ECHO BACK)
007114 0.J241                              CLC                       ;GET READY TO ROTATE
007116 006177 172162                       ROL     @SWR              ;MOVE THE EXISTING BITS OVER
007:22 006177 172156                       ROL     @SWR              ;TO MAKE ROOM FOR THE INCOMING
007126 006177 172152                       ROL     @SWR              ;THREE BITS FROM THIS CHARACTER
007132 103735                              BCS     1$                ;ERROR
00,134 022627 000060                       CMP     (SP)+,#60         ;IS IT LOWER THAN 0?
007140 002732                              BLT     1$                ;IF SO, GO ASK AGAIN
007142 026627 177776 000067                CMP     -2(SP),#67        ;IS IT HIGHER THAN 7?
007150 003326                              BGT     1$                ;IF SO, GO ASK AGAIN
007152 042746 177770                       BIC     #↑C<7>,-(SP)      ;ISOLATE INFORMATION BITS
007156 052677 172122                       BIS     (SP)+,@SWR        ;ADD THEM TO THE SWITCH REGISTER
007162 000733                              BR      3$                ;GO CHECK FOR THE NEXT CHARACTER
007164 105737 007234               4$:     TSTB    92$               ;HAS THE SWR BEEN CHANGED?
007170 001003                              BNE     5$                ;IF YES GO TYPE CRLF
007172 013777 007226 172104                MOV     90$,@SWR          ;IF NOT RESTORE SWR
007200 104402 001357               5$:     TYPE    ,$CRLF            ;TYPE A CARRIAGE RETURN AND LINE FEED
007204 000207                      6$:     RTS     PC                ;RETURN TO CALLING PROCEDURE

007206 020200 051450 051127        89$:    .ASCIZ  <200>? (SWR)=/?
007214 036451 000057
                                   .EVEN
007220 000001                      88$:    1
007222    006    000                       .BYTE   6,0
007224 007226                              90$
007226 000000                      90$:    .WORD   0
007230 036457 000057               91$:    .ASCIZ  ?/=/?
007234    000                      92$:    .BYTE   0
       007236                      .EVEN
                                   .SBTTL  POWER DOWN AND UP ROUTINES


                                   ;;*****************************************************************
                                   ;POWER DOWN ROUTINE
007236 012737 007402 000024        $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
007244 012737 000340 000026                MOV     #340,@#PWRVEC+2 ;;PRIO:7
```

```
      007252  010046                                   MOV     R0,-(SP)            ;;PUSH R0 ON STACK
      007254  010146                                   MOV     R1,-(SP)            ;;PUSH R1 ON STACK
      007256  010246                                   MOV     R2,-(SP)            ;;PUSH R2 ON STACK
      007260  010346                                   MOV     R3,-(SP)            ;;PUSH R3 ON STACK
      007262  010446                                   MOV     R4,-(SP)            ;;PUSH R4 ON STACK
      007264  010546                                   MOV     R5,-(SP)            ;;PUSH R5 ON STACK
      007266  017746  172012                           MOV     @SWR,-(SP)          ;;PUSH @SWR ON STACK
      007272  010637  007406                           MOV     SP,$SAVR6           ;;SAVE SP
      007276  012737  007310  000024                   MOV     #$PWRUP,@#PWRVEC    ;;SET UP VECTOR
      007304  000000                                   HALT
      007306  000776                                   BR      .-2                 ;;HANG UP

                                             ;;*************************************************************
                                             ;POWER UP ROUTINE
      007310  012737  007402  000024         $PWRUP:   MOV     #$ILLUP,@#PWRVEC    ;;SET FOR FAST DOWN
      007316  013706  007406                           MOV     $SAVR6,SP           ;;GET SP
      007322  005037  007406                           CLR     $SAVR6              ;;WAIT LOOP FOR THE TTY
      007326  005237  007406                 1$:       INC     $SAVR6              ;;WAIT FOR THE INC
      007332  001375                                   BNE     1$                  ;;OF  WORD
      007334  012677  171744                           MOV     (SP)+,@SWR          ;;POP STACK INTO @SWR
      007340  012605                                   MOV     (SP)+,R5            ;;POP STACK INTO R5
      007342  012604                                   MOV     (SP)+,R4            ;;POP STACK INTO R4
      007344  012603                                   MOV     (SP)+,R3            ;;POP STACK INTO R3
      007346  012602                                   MOV     (SP)+,R2            ;;POP STACK INTO R2
      007350  012601                                   MOV     (SP)+,R1            ;;POP STACK INTO R1
      007352  012600                                   MOV     (SP)+,R0            ;;POP STACK INTO R0
      007354  012737  007236  000024                   MOV     #$PWRDN,@#PWRVEC    ;;SET UP THE POWER DOWN VECTOR
      007362  012737  000340  000026                   MOV     #340,@#PWRVEC+2     ;;PRIO:7
      007370  104402                                   TYPE                        ;;REPORT THE POWER FAILURE
      007372  007410                         $PWRMG:   .WORD   MPFAIL              ;;POWER FAIL MESSAGE POINTER
      007374  012716                                   MOV     (PC)+,(SP)          ;;RESTART AT RESTART
      007376  010776                         $PWRAD:   .WORD   RESTART             ;;RESTART ADDRESS
      007400  000002                                   RTI
      007402  000000                         $ILLUP:   HALT                        ;;THE POWER UP SEQUENCE WAS STARTED
      007404  000776                                   BR      .-2                 ;; BEFORE THE POWER DOWN WAS COMPLETE
      007406  000000                         $SAVR6:   0                           ;;PUT THE SP HERE
      007410  050200  051127  043040         MPFAIL:   .ASCIZ  <200>/PWR FAILED. RESTART AT LAST TEST /
   (2) 007453     200  047105  020104         MEPASS:   .ASCIZ  <200>/END PASS DVDZA-A  /
   (2) 007477     200  052522  047116         MR:       .ASCIZ  <200>/RUNNING   /
   (2) 007513     200  051120  043517         MERR2:    .ASCIZ  <200>/PROGRAM INDICATES NO DEVICES PRESENT./
   (2) 007562  044600  051516  043125         MERR3:    .ASCIZ  <200>/INSUFFICIENT DATA!/
   (2) 007606  046200  041517  020113         MLOCK:    .ASCIZ  <200>/LOCK ON SELECTED TEST/
   (2) 007635     103  051123  020072         MCSRX:    .ASCIZ   /CSR: /
   (2) 007643     126  041505  020072         MVECX:    .ASCIZ  /VEC: /
   (2) 007651     120  051501  042523         MPASSX:   .ASCIZ  /PASSES: /
   (2) 007662  051105  047522  051522         MERRX:    .ASCIZ  /ERRORS: /
   (2) 007673     124  051505  020124         MTSTN:    .ASCIZ  /TEST NO: /
   (2) 007705     052  000040                 MASTEK:   .ASCIZ  /* /
   (2) 007710  052200  050131  020105         MNEW:     .ASCIZ  <200>/TYPE A BIT MAP OF DZV11'S DESIRED ACTIVE:  /
   (2) 007765     120  035103  000040         MERRPC:   .ASCIZ  /PC: /
   (2) 007772  046600  050101  047440         XHEAD:    .ASCIZ  <200>/MAP OF DZV11 STATUS/<200>
   (2) 010020  044600  046114  043505         MBADLN:   .ASCIZ  <200>/ILLEGAL ENTRY IN STAGGERED MODE/<200>
   (2)                                        .EVEN
   (2) 010062  000002                         XSTATQ:   2
      010064     006     003                             .BYTE   6,3
      010066  001344                                   $TMP1
```

```
010070     006      002                          .BYTE   6,2
010072  001346                                   $TMP2
                                         .EVEN
                                                 ;THIS ROUTINE ESTABLISHES WHICH MAINTENANCE MODE THE DEVICE IS IN
                                                 ;---------------------------------------------------------------
                                                 ;E=EXTERNAL LOOP BACK
                                                 ;I=INTERNAL LOOP BACK
                                                 ;S=STAGGERED LOOP BACK
010074  017605  000000                   .SETFLG:MOV     @(SP),R5        ;PICK UP ADDRESS OF TAG
010100  042737  000040  010270                   BIC     #40,INBUF       ;STRIP LOWER CASE
010106  122737  000105  010270                   CMPB    #'E,INBUF       ;IS IT EXTERNAL LOOP BACK ?
010114  001005                                   BNE     4$              ;NO
010116  013715  010206                           MOV     1$,(R5)         ;YES STORE INFO
010122  105037  001424                           CLRB    MNTFLG          ;SET MAINT BIT =0
010126  000422                                   BR      7$              ;GET OUT
010130  122737  000111  010270           4$:     CMPB    #'I,INBUF       ;IS IT INTERNAL LOOP BACK ?
010136  001006                                   BNE     5$              ;NO
010140  013715  010210                           MOV     2$,(R5)         ;YES STORE INFO
010144  112737  000010  001424                   MOVB    #MAINT,MNTFLG   ;SET UP THE MAINTENANCE FLAG LOADER
010152  000410                                   BR      7$              ;GET OUT
010154  122737  000123  010270           5$:     CMPB    #'S,INBUF       ;IS IT STAGGERED LOOP BACK ?
010162  001007                                   BNE     6$              ;WHAT ?
010164  013715  010212                           MOV     3$,(R5)         ;YES STORE INFO
010170  105037  001424                           CLRB    MNTFLG          ;ZERO BITS
010174  062716  000002                   7$:     ADD     #2,(SP)         ;POP AROUND
010200  000002                                   RTI
010202  104404                           6$:     INSTER                  ;RETRY
010204  000733                                   BR      .SETFLG         ;DITTO
010206  000200                           1$:     .WORD   200             ;EXTERNAL =  E
010210  000000                           2$:     .WORD   0               ;INTERNAL = I
010212  100000                           3$:     .WORD   100000          ;STAGGERED = S
```

```
                                                ;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

010214  017605  000000                  .PAWCH:MOV      @(SP),R5
010220  142737  000040  010270                  BICB    #40,INBUF               ;SET FOR LOWER CASE INPUT
010226  122737  000105  010270                  CMPB    #'E,INBUF               ;IS IT "E" ?
010234  001002                                  BNE     1$
010236  105015                                  CLRB    (R5)                    ;000
010240  000406                                  BR      2$
010242  122737  000103  010270          1$:     CMPB    #'C,INBUF               ;IS IT "C" ?
010250  001005                                  BNE     3$
010252  112715  177777                          MOVB    #-1,(R5)                ;3177
010256  062716  000002                  2$:     ADD     #2,(SP)
010262  000002                                  RTI
010264  104404                          3$:     INSTER                          ;RETRY
010266  000752                                  BR      .PAWCH

                                                ;BUFFERS FOR INPUT-OUTPUT

010270  000000                          INBUF:  0
        010332                          .=.+40
010332  000000                          TEMP:   0
        010374                          .=.+40
010374  000000                          MDATA:  0
        010436                          .=.+40

```

```
                                        ;
                                        ;ROUTINE USED TO "CYCLE" THROUGH UP TO SIXTEEN DZV11'S
                                        ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
                                        ;AND RUNS THE SPECIFIED DZV11'S.   THIS ROUTINE *MUST*
                                        ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
                                        ;SETUP NECESSARY.
                                        ;

010436  005737  001406          CYCLE:  TST     DZVACTV           ;ARE ANY DZV11'S TO BE TESTED?
010442  001004                          BNE     1$                ;BR IF OK.
010444  104402  007513                  TYPE    ,MERR2            ;NO DZV11'S SELECTED!!
010450  000000                          HALT                      ;STOP THE SHOW.
010452  000776                          BR      .-2               ;DISQUALIFY CONT. SW.
010454  013737  004542  001354  1$:     MOV     $MXCNT,$TIMES     ;RESTORE THE NUMBER OF ITERATIONS TO MAKE
010462  033737  001412  001406          BIT     RUN,DZVACTV       ;IS THIS ONE "ACTIVE"
010470  001017                          BNE     2$                ;BR IF GOOD ONE FOUND.
010472  006137  001412                  ROL     RUN               ;UPDATE POINTER
010476  005537  001412                  ADC     RUN               ;CATCH CARRY FROM RUN
010502  062737  000012  001420          ADD     #12,ACTIVE        ;UPDATE ADDRESS POINTER.
010510  022737  001740  001420          CMP     #DZV.END,ACTIVE   ;HAVE WE PASSED THE END OF THE MAP?
010516  001356                          BNE     1$                ;IF NO, KEEP GOING; NOT ALL TESTED FOR.
010520  012737  001500  001420          MOV     #DZV.MAP,ACTIVE   ;RESET ADDRESS POINTER.
010526  000752                          BR      1$                ;KEEP LOOKING FOR ACTIVE DZV11
010530  006137  001412          2$:     ROL     RUN               ;UPDATE POINTER.
010534  005537  001412                  ADC     RUN               ;CATCH CARRY.
010540  013700  001420                  MOV     ACTIVE,R0         ;GET ADDRESS POINTER.
010544  062737  000012  001420          ADD     #12,ACTIVE        ;UPDATE.
010552  022737  001740  001420          CMP     #DZV.END,ACTIVE
                                                                  ;ALL DONE?
010560  001003                          BNE     3$                ;BR IF NO.
010562  012737  001500  001420          MOV     #DZV.MAP,ACTIVE   ;RESTORE POINTER.
010570  012037  001174          3$:     MOV     (R0)+,$BASE       ;LOAD SYSTEM CTRL. REG
010574  012037  002040                  MOV     (R0)+,DZVRIV      ;LOAD VECTOR
010600  012037  001366                  MOV     (R0)+,LINE        ;SET UP DZV LINES ACTIVE
010604  012037  001370                  MOV     (R0)+,PAR         ;SET UP PARAMETERIZATION
010610  012037  001372                  MOV     (R0)+,MODE        ;SET UP MAINTENANCE MODE
010614  105037  001424                  CLRB    MNTFLG  ;RESET MAINT. FLAG IF
010620  005737  001372                  TST     MODE              ;RUNNING TESTS
010624  001003                          BNE     9$                ;IN
010626  112737  000010  001424          MOVB    #MAINT,MNTFLG     ;INTERNAL MAINT. MODE
010634  004737  011002          9$:     JSR     PC,DZVLEV         ;SET UP
010640  005737  000042                  TST     @#42              ;ARE WE UNDER MONITOR CONTROL?
010644  001051                          BNF     7$                ;IF YES, SKIP THIS SETUP
010646  032777  000002  170430          BIT     #SW01,@SWR        ;IF SW01=1, GET STARTING TEST #
010654  001445                          BEQ     7$                ;BR IF NO TEST IS TO BE INPUTTED
010656  104402  001357          4$:     TYPE    ,$CRLF
010662  104403                          INSTR                     ;CALL THE STRING INPUT ROUTINE
010664  007673                          MTSTN                     ;POINTER TO MESSAGE TO BE PRINTED
010666  104405                          PARAM                     ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
010670  000001                          1                         ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
010672  001000                          1000                      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
010674  001246                          $TSTNM                    ;POINTER TO MAP LOCATION TO BE FILLED
010676     000                          .BYTE   0                 ;MASK OF INVALID BITS FOR THIS PARAMETER
010677     001                          .BYTE   1                 ;NUMBER OF PARAMETERS TO STORE
010700  012700  012170                  MOV     #TST1,R0
```

```
010704  022710  000004                  5$:     CMP     #4,(R0)
010710  001020                                  BNE     6$
010712  022760  012737  000002                  CMP     #12737,2(R0)
010720  001014                                  BNE     6$
010722  023760  001246  000004                  CMP     $TSTNM,4(R0)    ;IS THIS THE TEST ?
010730  001010                                  BNE     6$              ;IF NOT, DON'T PROCESS NUMBER
010732  010037  001252                          MOV     R0,$LPADR       ;SAVE PC
010736  062737  000002  001252                  ADD     #2,$LPADR       ;POP OVER PREVIOUS SCOPE
010744  104402  001357                          TYPE    ,$CRLF
010750  000412                                  BR      8$
010752  005720                          6$:     TST     (R0)+
010754  020027  015646                          CMP     R0,#TLAST+10
010760  001351                                  BNE     5$
010762  104402  001356                          TYPE    ,$QUES
010766  000733                                  BR      4$
010770  012737  012170  001252          7$:     MOV     #TST1,$LPADR    ;PREPARE TEST ADDRESS
010776                                  8$:
010776  000177  170250                  RESTART:JMP     @$LPADR         ;GO START TESTING.***WARNING!****
                                                                        ;THIS JUMP IS USED BY POWER UP ROUTINE!!!!

                                        ;THIS UTILITY SETS UP CSR'S,SETS UP VECTORS.
011002  013700  002040                  DZVLEV: MOV     DZVRIV,R0       ;PLACE THE BASE VECTOR ADDRESS IN R0
011006  062700  000002                          ADD     #2,R0           ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
011012  010037  002042                          MOV     R0,DZVRIS       ;STORE IT HERE
011016  062700  000002                          ADD     #2,R0           ;CALCULATE THE TRANSMITTER INTERRUPT VECTOR
011022  010037  002044                          MOV     R0,DZVTIV       ;STORE IT HERE
011026  062700  000002                          ADD     #2,R0           ;CALCULATE THE TRANSMITTER VECTOR STATUS ADDRESS
011032  010037  002046                          MOV     R0,DZVTIS       ;STORE IT HERE

                                        ;THIS SEGMENT SETS UP POINTERS FOR THE GIVEN DZV11. $BASE IS THE BASE ADDRESS
                                        ;OF THE DEVICE
011036  013700  001174                          MOV     $BASE,R0        ;COPY THE ADDRESS BEING LOADED
011042  010037  002010                          MOV     R0,DZVCSR       ;XXX0
011046  005200                                  INC     R0
011050  010037  002012                          MOV     R0,HDZVCSR      ;XXX1
011054  005200                                  INC     R0
011056  010037  002014                          MOV     R0,DZVRBUF      ;XXX2
011062  010037  002020                          MOV     R0,DZVLPR       ;XXX2
011066  005200                                  INC     R0
011070  010037  002016                          MOV     R0,HDZVRBUF     ;XXX3
011074  010037  002022                          MOV     R0,HDZVLPR      ;XXX3
011100  005200                                  INC     R0
011102  010037  002024                          MOV     R0,DZVTCR       ;XXX4
011106  005200                                  INC     R0
011110  010037  002026                          MOV     R0,HDZVTCR      ;XXX5
011114  005200                                  INC     R0
011116  010037  002030                          MOV     R0,DZVMSR       ;XXX6
011122  010037  002034                          MOV     R0,DZVTDR       ;XXX6
011126  005200                                  INC     R0
011130  010037  002032                          MOV     R0,HDZVMSR      ;XXX7
011134  010037  002036                          MOV     R0,HDZVTDR      ;XXX7
011140  000207                                  RTS     PC
```

```
                                               ;CONVERT DECIMAL ASCII STRING TO OCTAL
011142  011605                         .PARMD: MOV     (SP),R5
011144  012537  011326                         MOV     (R5)+,6$
011150  012537  011330                         MOV     (R5)+,7$
011154  012537  011332                         MOV     (R5)+,8$
011160  112537  011334                         MOVB    (R5)+,9$
011164  112537  011335                         MOVB    (R5)+,10$
011170  010516                                 MOV     R5,(SP)
011172  005005                         2$:     CLR     R5
011174  012704  010270                         MOV     #INBUF,R4
011200  122714  000015                         CMPB    #15,(R4)
011204  001424                                 BEQ     3$
011206  121427  000060                 1$:     CMPB    (R4),#'0
011212  002421                                 BLT     3$
011214  121427  000071                         CMPB    (R4),#'9
011220  003016                                 BGT     3$
011222  142714  000060                         BICB    #'0,(R4)
011226  005002                                 CLR     R2
011230  152402                                 BISB    (R4)+,R2
011232  060205                                 ADD     R2,R5
011234  122714  000015                         CMPB    #15,(R4)
011240  001410                                 BEQ     4$
011242  006305                                 ASL     R5      ;X2
011244  010502                                 MOV     R5,R2   ;SAVE X2
011246  006305                                 ASL     R5      ;X4
011250  006305                                 ASL     R5      ;X8
011252  060205                                 ADD     R2,R5   ;TIMES 10
011254  000754                                 BR      1$
011256  104404                         3$:     INSTER
011260  000744                                 BR      2$

                                               ;TEST TO SEE IF NUMBER IS WITHIN LIMITS

011262  020537  011330                 4$:     CMP     R5,7$
011266  101373                                 BHI     3$
011270  020537  011326                         CMP     R5,6$
011274  103770                                 BLO     3$
011276  133705  011334                         BITB    9$,R5
011302  001365                                 BNE     3$

                                               ;STORE NUMBER AT SPECIFIED ADDRESS

011304  013704  011332                         MOV     8$,R4
011310  010524                         5$:     MOV     R5,(R4)+
011312  062705  000002                         ADD     #2,R5
011316  105337  011335                         DECB    10$
011322  001372                                 BNE     5$
011324  000002                                 RTI
011326  000000                         6$:     0
011330  000000                         7$:     0
011332  000000                         8$:     0
011334     000                         9$:     .BYTE 0
011335     000                         10$:    .BYTE 0
```

```
                                        ;*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.

011336  012700  001500          SETAPT: MOV     #DZV.MAP,R0     ;POINT TO THE DEVICE MAP TABLE
011342  013701  001174                  MOV     $BASE,R1        ;BUILD DEVICE ADDRESSES IN R1
011346  013702  001170                  MOV     $VECT1,R2       ;BUILD DEVICE VECTORS IN R2
011352  042702  177007                  BIC     #↑C<770>,R2     ;STRIP AWAY OTHER INFORMATION

011356  012704  001204                  MOV     #$DDW0,R4       ;POINT TO THE BEGINNING OF DEVICE PARAMETERS
011362  013705  001176                  MOV     $DEVM,R5        ;GET THE MAP OF ACTIVE DEVICES
011366  105037  001414                  CLRB    DZVNUM          ;INITIALIZE NO. OF DEVICES IN SYSTEM
011372  005037  001410                  CLR     SAVACTV         ;CLEAR THE ACTIVE BIT MAP
011376  006005                  1$:     ROR     R5              ;GET A DEVICE SELECTION BIT
011400  103407                          BCS     3$              ;IF IT IS SELECTED, GO SET UP A MAP
011402  001422                          BEQ     5$              ;IF NO MORE ARE SELECTED, GET OUT OF SETUP
011404  005724                          TST     (R4)+           ;POINT TO NEXT DEVICE DESCRIPTOR
011406  062701  000010          2$:     ADD     #10,R1          ;SET UP THE NEXT ADDRESS
011412  062702  000010                  ADD     #10,R2          ;SET UP THE NEXT VECTOR GROUP
011416  000767                          BR      1$              ;GO SEE IF MORE DEVICES REMAIN
011420  006137  001410          3$:     ROL     SAVACTV         ;SET BIT IN ACTIVE DEVICE MAP
011424  105237  001414                  INCB    DZVNUM          ;INCREMENT NO. OF ACTIVE DEVICES IN SYSTEM
011430  010120                          MOV     R1,(R0)+        ;LOAD DEVICE ADDRESS
011432  010220                          MOV     R2,(R0)+        ;LOAD THE VECTOR ADDRESS
011434  013720  001200                  MOV     $CDW1,(R0)+     ;GET THE NUMBER OF LINES IN OPERATION
011440  012420                          MOV     (R4)+,(R0)+     ;LOAD DEVICE PARAMETERS
011442  013720  001202                  MOV     $CDW2,(R0)+     ;LOAD DEFAULT TESTING MODE
011446  000757                          BR      2$              ;GO BUILD THE NEXT ADDRESS
011450  012710  177777          5$:     MOV     #-1,(R0)        ;TERMINATE THE DEVICE MAP
011454  012737  001142  001304          MOV     #$SWREG,SWR     ;SET TO SOFTWARE APT SWITCH REGISTER
011462  000207                          RTS     PC              ;RETURN TO PRINT STATUS TABLE


                                        ;*ROUTINE USED TO "AUTO SIZE" THE DZV11
                                        ;*CSR AND VECTOR.
                                        ;*NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
                                        ;*      ADDRESS RANGE (160000:163770)
                                        ;*      AND THE VECTOR MAY BE ANY WHERE IN THE
                                        ;*      FLOATING VECTOR RANGE (300:770)
                                        ;*

011464                          AUTO.SIZE:
011464  000005                          RESET                   ;INSURE A BUS INIT.
011466  105337  001422                  DECB    INIFLG          ;SHOW THAT I WAS HERE
011472  012702  001500          CSRMAP: MOV     #DZV.MAP,R2     ;LOAD MAP POINTER.
011476  012703  001204                  MOV     #$DDW0,R3       ;POINT TO ETABLE DEVICE DESCRIPTOR WORDS
011502  005022                  1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
011504  022702  001740                  CMP     #DZV.END,R2     ;ALL DONE?
011510  001374                          BNE     1$              ;BR IF NO
011512  105037  001414                  CLRB    DZVNUM          ;SET OCTAL NUMBER OF DZV11'S TO 0
011516  012702  001500                  MOV     #DZV.MAP,R2
011522  012701  160000                  MOV     #160000,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
011526  012737  011772  000004          MOV     #6$,@#4         ;SET FOR NON-EXISTENT DEVICE TIME OUT
011534  052711  000040          2$:     BIS     #BIT5,(R1)      ;TRY TO SET MASTER SCAN ENABLE
011540  052761  000017  000004          BIS     #17,4(R1)       ;TRY TO TRANSMIT ON ANY LINE
011546  005000                          CLR     R0              ;USE R0 AS A COUNTER
```

```
011550  005711                            7$:     TST     (R1)               ;HAS TRANSMITTER READY COME UP?
011552  100403                                    BMI     8$                 ;IF SO, GO GET A FINAL CHECK
011554  005300                                    DEC     R0                 ;REDUCE COUNT. TIME UP?
011556  001374                                    BNE     7$                 ;IF NOT, KEEP WAITING
011560  000437                                    BR      3$                 ;ASSUME IT'S NOT A DZV11
011562  032761  000017  000004            8$:     BIT     #17,4(R1)          ;ARE ANY TCR BITS  STILL SET? THEY SHOULD BE
011570  001433                                    BEQ     3$                 ;IF IT'S NOT, ASSUME IT'S NOT A DZV11
011572  032711  000040                            BIT     #BIT5,(R1)         ;IS MASTER SCAN ENABLE STILL SET?
011576  001430                                    BEQ     3$                 ;IF NOT, ASSUME IT'S NOT A DZV11
011600  052711  000020                            BIS     #20,(R1)           ;SET DEVICE CLEAR
011604  000240                                    NOP
011606  032711  000040                            BIT     #40,(R1)           ;DID SCANNER CLEAR
011612  001022                                    BNE     3$                 ;IF NOT ASSUME IT IS NOT DZV
011614  005061  000004                            CLR     4(R1)              ;GET RID OF TCR BITS
                                          ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DZV11 CSR ADDRESS.
011620  010122                                    MOV     R1,(R2)+           ;STORE CSR IN CORE TABLE.
011622  005722                                    TST     (R2)+              ;POP OVER VECTOR STORE AREA
011624  012722  000017                            MOV     #17,(R2)+          ;SET THE DEFAULT LINE SELECTION PARAMETER
011630  012712  017470                            MOV     #17470,(R2)        ;SET THE DEFAULT PARAMETERS
011634  012223                                    MOV     (R2)+,(R3)+        ;COPY PARAMETERS INTO ETABLE DESCRIPTOR
011636  005022                                    CLR     (R2)+              ;SET THE DEFAULT MODE OF OPERATION
011640  012712  177777                            MOV     #-1,(R2)           ;TERMINATE LIST
011644  105237  001414                            INCB    DZVNUM             ;UPDATE DEVICE COUNTER
011650  122737  000020  001414                    CMPB    #20,DZVNUM         ;ARE MAX. NO. OF DEV FOUND?
011656  001405                                    BEQ     100$               ;YES DON'T LOOK FOR ANY MORE.
011660  062701  000010                    3$:     ADD     #10,R1             ;UPDATE CSR POINTER ADDRESS
011664  022701  164000                            CMP     #164000,R1
011670  001321                                    BNE     2$                 ;BR IF MORE ADDRESS TO CHECK.
011672                                    100$:
011672  105737  001414                            TSTB    DZVNUM             ;WERE ANY DZV11'S FOUND AT ALL?
011676  001430                                    BEQ     5$                 ;ERROR AUTO SIZER FOUND NO DZV11'S IN THIS SYS.
011700  113701  001414                            MOVB    DZVNUM,R1
011704  012737  000001  001410                    MOV     #1,SAVACTV         ;CREATE A BIT MAP OF THE ACTIVE
011712  005301                            4$:     DEC     R1                 ;DEVICES IN THE SYSTEM
011714  001404                                    BEQ     98$
011716  000261                                    SEC
011720  006137  001410                            ROL     SAVACTV
011724  000772                                    BR      4$
011726  013737  001500  001174            98$:    MOV     DZCR0,$BASE        ;POINT TO THE ADDRESS OF FIRST DEVICE
011734  013737  001510  001202                    MOV     MANTO,$CDW2        ;INDICATE TO ETABLE WHAT MODE IS BEING USED
011742  012737  000006  000004            99$:    MOV     #6,@#4             ;RESTORE TRAP VECTOR
011750  013737  001410  001176                    MOV     SAVACTV,$DEVM      ;SAVE ACTIVE REGISTER
011756  000410                                    BR      VECMAP             ;GO FIND THE VECTOR NOW.
011760  104402  007513                    5$:     TYPE    ,MERR2             ;NOTIFY OPR THAT NO DZV11'S FOUND.
011764  005000                                    CLR     R0                 ;MAKE DATA DISPLAY ZERO
011766  000000                                    HALT                       ;STOP THE SHOW
011770  000776                                    BR      .-2                ;DISABLE CONT. SW.
011772  012716  011660                    6$:     MOV     #3$,(SP)           ;ENTERED BY NON-EXISTENT TIME-OUT
011776  000002                                    RTI                        ;RETURN TO MAINSTREAM

012000  012737  000200  000022            VECMAP: MOV     #MASK,@#22         ;SET IOT TRAP PRIORITY
012006  012737  012122  000020                    MOV     #4$,@#20           ;SET IOT TRAP VECTOR
012014  012702  001500                            MOV     #DZV.MAP,R2        ;SET SOFTWARE POINTER
012020  012700  000300                            MOV     #300,R0            ;FLOATING VECTORS START HERE.
012024  012701  000302                            MOV     #302,R1            ;PC OF IOT INSTR.
012030  010120                            1$:     MOV     R1,(R0)+           ;START FILLING VECTOR AREA
```

```
012032  012721  000004                  MOV     #4,(R1)+           ;WITH .+2; IOT
012036  022021                          CMP     (R0)+,(R1)+        ;ADD 2 TO R0 +R1
012040  020127  001000                  CMP     R1,#1000           ;HAS THE VECTOR AREA BEEN EXCEEDED?
012044  101771                          BLOS    1$                 ;BR IF MORE TO FILL
012046  013704  001410                  MOV     SAVACTV,R4         ;STORE TEMPORARILY
012052  006004                  2$:     ROR     R4                 ;BRING OUT A BIT
012054  103036                          BCC     5$                 ;BR IF ALL DONE
012056  106427  000000                  MTPS    #0                 ;ZERO CPU PRIO
012062  012772  040040  000000          MOV     #BIT14+BITS,@(R2)      ;SET TIE AND MAS SCAN
012070  011201                          MOV     (R2),R1            ;GET CSR
012072  112761  000017  000004          MOVB    #17,4(R1)          ;SET THE TCR BITS FOR ALL LINES
                                                                   ;ATTEMPT TO FORCE AN INTERRUPT
012100  005200                          INC     R0                 ;STALL
012102  001376                          BNE     .-2                ;       FOR TIME TO INTERRUPT
012104  012762  000300  000002          MOV     #300,2(R2)         ;NO INTERRUPT ASSUME 300 AND FIX DZV11 LATER
012112  000005                          RESET                      ;INIT
012114  062702  000012          3$:     ADD     #12,R2             ;POP SOFTWARE POINTER
012120  000757                          BR      2$                 ;KEEP GOING
012122  011662  000002          4$:     MOV     (SP),2(R2)         ;GET VECTOR ADDRESS
012126  162762  000010  000002          SUB     #10,2(R2)          ;POINT BACK TO THE CORRECT VECTOR
012134  042762  000007  000002          BIC     #7,2(R2)           ;CLEAR JUNK
012142  022626                          POP2SP                     ;POP IOT JUNK OFF STACK
012144  012716  012114                  MOV     #3$,(SP)           ;SET FOR RETURN
012150  000002                          RTI
012152  013737  001502  001170  5$:     MOV     DZVCO,$VECT1       ;COPY VECTOR OF FIRST DEVICE INTO ETABLE
012160  012737  004300  000020          MOV     #.SCOPE,IOTVEC     ;RESTORE THE SCOPE TRAP
012166  000207                          RTS     PC                 ;ALL DONE WITH "AUTO SIZING"
```

```

                                                 ;************************ TEST 1 ******************************
                                                 ;*THIS TEST PROVES THE BUS REPLY RESPONSE
                                                 ;*DURING A READ OR WRITE TO THE FOLLOWING ADDRESS:
                                                 ;*        DZVCSR, DZVRBUF, DZVTCR, DZVMSR
                                         ;;*  TEST 1
                                         ;;*************************************************************
012170  000004                           TST1:   SCOPE
012172  012737  000001  001246                   MOV     #1,$TSTNM        ;LOAD THE NUMBER OF THIS TEST
012200  012737  012360  001362                   MOV     #TST2,NEXT       ;POINT TO THE START OF THE NEXT TEST
012206  012737  012346  000004                   MOV     #5$,4            ;SET TRAP VECTOR
012214  012737  000200  000006                   MOV     #MASK,6          ;SET PRIORITY TO HIGH(MASK INTERRUPTS)
012222  012737  012230  001364                   MOV     #1$,LOCK         ;SET RETURN IF SW09=11
012230  013700  002010                   1$:     MOV     DZVCSR,R0        ;SET ADDRESS TO TEST
012234  011001                                   MOV     (R0),R1          ;READ THE ADDRESS
012236  000240                                   NOP                      ;WASTE TIME
012240  005010                                   CLR     (R0)             ;WRITE THE ADDRESS
012242  000240                                   NOP                      ;WASTE TIME
012244  012737  012252  001364                   MOV     #2$,LOCK         ;SET RETURN ADDRESS FOR SW09
012252  013700  002014                   2$:     MOV     DZVRBUF,R0       ;SET ADDRESS TO TEST
012256  011001                                   MOV     (R0),R1          ;READ THE ADDRESS
012260  000240                                   NOP                      ;
012262  005010                                   CLR     (R0)             ;WRITE THE ADDRESS
012264  000240                                   NOP                      ;WASTE TIME
012266  012737  012274  001364                   MOV     #3$,LOCK         ;SET RETURN ADDRESS FOR SW09
012274  013700  002024                   3$:     MOV     DZVTCR,R0        ;SET ADDRESS TO TEST
012300  011001                                   MOV     (R0),R1          ;READ THE ADDRESS
012302  000240                                   NOP                      ;
012304  005010                                   CLR     (R0)             ;WRITE THE ADDRESS
012306  000240                                   NOP
012310  012737  012316  001364                   MOV     #4$,LOCK         ;SET RETURN ADDRESS
012316  013700  002030                   4$:     MOV     DZVMSR,R0        ;SET ADDRESS TO TEST
012322  011001                                   MOV     (R0),R1          ;READ FROM ADDRESS
012324  000240                                   NOP                      ;
012326  005010                                   CLR     (R0)             ;WRITE THE ADDRESS
012330  000240                                   NOP                      ;
012332  012737  000006  000004                   MOV     #6,4             ;SET TRAP CATCHER BACK TO NORMAL
012340  005037  000006                           CLR     6                ;
012344  104400                                   ADVANCE                  ;SCOPE THIS TEST
012346  011601                           5$:     MOV     (SP),R1          ;SAVE PC OF TRAP
012350  022626                                   POP2SP                   ;POP TRAP OFF STACK
012352  104001                                   ERROR   1                ;*NO BUS REPLY RESPONSE.
012354  104401                                   SCOP1                    ;SW09=1?
012356  000111                                   JMP     (R1)             ;RTI
                                                 ;*********************** TEST 2 ******************************
                                                 ;*THIS TEST PROVES THAT BIT "DCLR"
                                                 ;*CAN BE SET AND THAT IT WILL CLEAR
                                                 ;*BY ITSELF
                                         ;;*  TEST 2
                                         ;;*************************************************************
012360  000004                           TST2:   SCOPE
012362  012737  000002  001246                   MOV     #2,$TSTNM        ;LOAD THE NUMBER OF THIS TEST
012370  012737  012424  001362                   MOV     #TST3,NEXT       ;POINT TO THE START OF THE NEXT TEST
012376  013700  002010                           MOV     DZVCSR,R0        ;SET POINTER
012402  012710  000020                           MOV     #DCLR,(R0)       ;SET DCLR
012406  005005                                   CLR     R5               ;SET EXPECTED TO 0
```

```
012410  005003                                  CLR     R3              ;DUAL LOOP COUNTER
012412  011004                          2$:     MOV     (R0),R4         ;IS DCLR CLEAR?
012414  001403                                  BEQ     3$              ;IF YES , GO TO THE NEXT TEST
012416  105203                                  INCB    R3              ;IF NO COUNT 1 OF 256 TICKS
012420  001374                                  BNE     2$              ;HAS THE TIME EXPIRED? IF NO, GO TEST BIT AGAIN
012422  104002                                  ERROR   2               ;*DCLR FAILED TO CLEAR
012424                                  3$:
                                                ;********************** TEST 3 ****************************
                                                ;*TEST TO VERIFY THAT THE R/W BITS OF THE
                                                ;*DZVCSR REGISTER CAN BE SET. THEN VERIFY THAT
                                                ;*THESE BITS CAN BE CLEARED. AND FINALLY, VERIFY
                                                ;*THAT AFTER BEING SET AGAIN THEY CAN BE
                                                ;*CLEARED BY A "DEVICE CLEAR".
                                                ;*THE BITS TESTED ARE:  MAINT, MSENAB, SILOEN,
                                                ;*RIE, AND TIE.
                                        ;:*  TEST 3
                                        ;:*********************************************************
012424  000004                          TST3:   SCOPE
012426  012737  000003  001246                  MOV     #3,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
012434  012737  012602  001362                  MOV     #TST4,NEXT      ;POINT TO THE START OF THE NEXT TEST
012442  013700  002010                          MOV     DZVCSR,R0       ;GET BASE ADDRESS
012446  012703  012562                          MOV     #5$,R3          ;SET R3 TO TOP OF TABLE
012452  011305                          1$:     MOV     (R3),R5         ;SET BIT
012454  012737  012462  001364                  MOV     #11$,LOCK       ;SETUP FOR TIGHT SCOPE LOOP
012462  010510                          11$:    MOV     R5,(R0)         ;SET BIT IN DEVICE
012464  011004                                  MOV     (R0),R4         ;READ THE BIT FROM DEVICE
012466  020504                                  CMP     R5,R4           ;WAS BIT SET?
012470  001401                                  BEQ     2$              ;BR IF YES
012472  104002                                  ERROR   2               ;*BIT R/W FAILURE
012474  104401                          2$:     SCOP1                   ;IS SWITCH 9 SET?
012476  012737  012504  001364                  MOV     #12$,LOCK       ;SET FOR NEXT TIGHT SCOPE LOOP
012504  040510                          12$:    BIC     R5,(R0)         ;CLEAR THE BIT.
012506  011004                                  MOV     (R0),R4         ;READ DEVICE
012510  001403                                  BEQ     3$              ;BR IF BITS WERE CLEARED.
012512  005005                                  CLR     R5              ;CLEAR FOR ERROR PRINTOUT
012514  104002                                  ERROR   2               ;*BIT FAILED TO CLEAR
012516  011305                                  MOV     (R3),R5         ;RESTORE THE BIT.
012520  104401                          3$:     SCOP1                   ;SW09 SET?
012522  012737  012530  001364                  MOV     #13$,LOCK       ;SET UP FOR NEXT TIGHT SCOPE
012530  010510                          13$:    MOV     R5,(R0)         ;SET THE BIT AGAIN
012532  104413                                  DEVICE.CLR              ;ISSUE DEVICE CLEAR
012534  011004                                  MOV     (R0),R4         ;READ THE BIT.
012536  001403                                  BEQ     4$              ;BR IF CIT CLEARED BY INIT (DEVICE CLEAR)
012540  005005                                  CLR     R5              ;SET EXPECTED TO ZERO
012542  104002                                  ERROR   2               ;*BIT NOT CLEARED BY DEVICE CLEAR
012544  011305                                  MOV     (R3),R5         ;RESTORE BIT AGAIN
012546  104401                          4$:     SCOP1                   ;SW09 SET?
012550  062703  000002                          ADD     #2,R3           ;POP R3
012554  005713                                  TST     (R3)            ;IS THIS THE END OF TABLE?
012556  001407                                  BEQ     6$              ;IF YES GET OUT
012560  000734                                  BR      1$              ;OTHERWISE TEST NEXT BIT
012562  000010                          5$:     #MAINT                  ;CSR BIT: INTERNAL MAINTENANCE
012564  000040                                  #MSENAB                 ;CSR BIT: MASTER SCAN ENABLE
012566  010000                                  #SILOEN                 ;CSR BIT: SILO ENABLE
012570  000100                                  #RIE                    ;CSR BIT: RECEIVER INTER. ENABLE
012572  040000                                  #TIE                    ;CSR BIT: TRANS. INTER. ENABLE
```

```
012574  000000                                   #0                    ;END OF TABLE
012576  005037  001364                  6$:      CLR     LOCK          ;ZERO LOCK INDICATOR
                                                 ;*********************** TEST 4 ******************************
                                                 ;*THIS TESTS THAT ALL OF THE TCR BITS
                                                 ;*CAN BE: SET, CLEARED, AND CLEARED BY A DEVICE CLEAR.
                                                 ;*THIS TEST ALSO DETERMINES IF THE DTR BITS CAN
                                                 ;*BE SET, CLEARED, AND CLEARED BY A RESET.
                                        ;:*  TEST 4
                                        ;:************************************************************
012602  000004                          TST4:    SCOPE
012604  012737  000004  001246                   MOV     #4,$TSTNM     ;LOAD THE NUMBER OF THIS TEST
012612  012737  013006  001362                   MOV     #TST5,NEXT    ;POINT TO THE START OF THE NEXT TEST
012620  013700  002024                           MOV     DZVTCR,R0     ;SET DEVICE ADDRESS
012624  012703  012712                           MOV     #5$,R3        ;SET R3 POINTER TO TOP OF TABLE
012630  012737  012640  001364          1$:      MOV     #11$,LOCK     ;SET LOCK FOR SW09 SCOPE LOOP
012636  011305                                   MOV     (R3),R5       ;SET EXPECTED RESULTS
012640  010510                          11$:     MOV     R5,(R0)       ;SET THE BIT
012642  011004                                   MOV     (R0),R4       ;READ THE BIT FROM THE DEVICE
012644  020504                                   CMP     R5,R4         ;DID THE BIT SET?
012646  001401                                   BEQ     2$            ;BR IF YES
012650  104002                                   ERROR   2             ;*BIT FAILED TO SET.
012652  104401                          2$:      SCOP1                 ;SW09 SET?
012654  012737  012662  001364                   MOV     #3$,LOCK      ;SET UP FOR NEXT TIGHT SCOPE LOOP
012662  040510                          3$:      BIC     R5,(R0)       ;CLEAR THE BIT
012664  011004                                   MOV     (R0),R4       ;READ THE REGISTER
012666  001403                                   BEQ     4$            ;BR IF YES
012670  005005                                   CLR     R5            ;SET EXPECTED TO 0
012672  104002                                   ERROR   2             ;*REPORT BIT NOT CLEAR
012674  011305                                   MOV     (R3),R5       ;RESTORE R5
012676  104401                          4$:      SCOP1                 ;SW09 SET?
012700  062703  000002                           ADD     #2,R3         ;POP POINTER TO NEXT TABLE ENTRY
012704  005713                                   TST     (R3)          ;END OF TABLE?
012706  001412                                   BEQ     6$            ;IF YES JUMP OVER TABLE
012710  000747                                   BR      1$            ;START TESTING NEXT BIT
012712  000001                          5$:      #TCR0                 ;TCR BIT FOR LINE 0
012714  000002                                   #TCR1                 ;TCR BIT FOR LINE 1
012716  000004                                   #TCR2                 ;TCR BIT FOR LINE 2
012720  000010                                   #TCR3                 ;TCR BIT FOR LINE 3
012722  000400                                   #DTR0                 ;DTR BIT FOR LINE 0
012724  001000                                   #DTR1                 ;DTR BIT FOR LINE 1
012726  002000                                   #DTR2                 ;DTR BIT FOR LINE 2
012730  004000                                   #DTR3                 ;DTR BIT FOR LINE 3
012732  000000                                   #0                    ;END OF TABLE
012734  005037  001364                  6$:      CLR     LOCK          ;CLEAR TIGHT SCOPE LOOP INDIC.
012740  012710  177777                           MOV     #-1,(R0)      ;SET ALL BITS IN TCR REGISTER
012744  012705  007400                           MOV     #007400,R5    ;SET FXPECTED
012750  104413                                   DEVICE.CLR            ;SET DCLR BIT IN CSR
012752  011004                                   MOV     (R0.,R4       ;READ REGISTER
012754  020504                                   CMP     R5,R4         ;TCR BITS CLEARED?
012756  001401                                   BEQ     7$            ;IF YES BRANCH
012760  104002                                   ERROR   2             ;TCR BITS NOT CLEARED!
012762  005005                          7$:      CLR     R5            ;SET EXPECTED TO ZERO
012764  005227  000000                  8$:      INC     #0            ;DELAY FOR ACT
012770  001375                                   BNE     8$            ;
012772  012710  177777                           MOV     #-1,(R0)      ;SET ALL POSSIBLE BITS
012776  000005                                   RESET                 ;DO BUS INIT
```

```
013000  011004                                  MOV     (RO),R4         ;DID REGISTER CLEAR?
013002  001401                                  BEQ     9$              ;IF YES GET OUT
013004  104002                                  ERROR   2               ;REGISTER DID NOT CLEAR!
013006                                  9$:
                                                ;*********************** TEST 5 ******************************
                                                ;*THIS TEST VERIFIES THAT
                                                ;*BITS "ROONE,TRDY, BIT9, BIT8,
                                                ;*AND SILOAL" ARE READ ONLY AND THAT TRDY IS
                                                ;*ZERO UNTIL A LINE IS SELECTED AND MSENAB IS SET.
                                                ;*
                                        ;;*  TEST 5
                                        ;;*************************************************************
013006  000004                          TST5:   SCOPE
013010  012737  000005  001246                  MOV     #5,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
013016  012737  013110  001362                  MOV     #TST6,NEXT      ;POINT TO THE START OF THE NEXT TEST
013024  013700  002010                          MOV     DZVCSR,RO       ;SET ADDPESS TO RO
013030  104413                                  DEVICE.CLR              ;DO A DEVICE CLEAR
013032  005005                                  CLR     R5              ;SET EXPECTED TO 0
013034  012710  121600                          MOV     #RDONE+TRDY+BIT9+BIT8+SILOAL,(RO)
                                                                        ;WRITE THE BITS
013040  011004                                  MOV     (RO),R4         ;READ BACK THE BITS
013042  001401                                  BEQ     2$              ;BR IF NONE ARE SET.
013044  104002                                  ERROR   2               ;*BITS ZERE SET.
013046  012705  100040                  2$:     MOV     #TRDY+MSENAB,R5 ;SET EXPECTED BIT
013052  052777  000017  166744                  BIS     #17,@DZVTCR     ;SET TCR BITS FOR ALL LINES
013060  052710  000040                          BIS     #MSENAB,(RO)    ;SET SCAN ENABLE
013064  005002                                  CLR     R2              ;SET COUNTER TO ZERO
013066  011004                          3$:     MOV     (RO),R4         ;READ THE REGISTER
013070  042704  001400                          BIC     #BIT9!BIT8,R4   ;MASK OUT LINE NO.
013074  020504                                  CMP     R5,R4           ;BIT SET?
013076  001404                                  BEQ     4$              ;BR IF YES
013100  104414                                  DELAY                   ;STALL TIME
013102  005202                                  INC     R2              ;UPDATE COUNTER
013104  001370                                  BNE     3$              ;BR IF COUNTER NOT DONE.
013106  104002                                  ERROR   2               ;*TRDY NOT SET!
013110                                  4$:
                                                ;*********************** TEST 6 ******************************
                                                ;*THIS TEST VERIFIES THAT:
                                                ;*TIE,SILOEN,RIE,MSENAB,AND MAINT ARE THE
                                                ;*ONLY R/W BITS IN THE DZVCSR AND THAT
                                                ;*SETTING "DCLR" IN THE CSR WILL CLEAR THESE BITS.
                                        ;;*  TEST 6
                                        ;;*************************************************************
013110  000004                          TST6:   SCOPE
013112  012737  000006  001246                  MOV     #6,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
013120  012737  013240  001362                  MOV     #TST7,NEXT      ;POINT TO THE START OF THE NEXT TEST
013126  104413                                  DEVICE.CLR              ;SET DCLR IN CSR
013130  013700  002010                          MOV     DZVCSR,RO       ;SET UP FOR ERROR MESSAGE
013134  012710  177757                          MOV     #↑C<DCLR>,(RO)  ;TRY TO SET ALL BITS EXCEPT DCLR
013140  012705  050150                          MOV     #TIE!SILOEN!RIE!MSENAB!MAINT,R5 ;MAKE EXPECTED
013144  011004                                  MOV     (RO),R4         ;ACTUAL
013146  020405                                  CMP     R4,R5           ;CMP EXPECTED VS ACTUAL
013150  001401                                  BEQ     1$              ;YES
013152  104002                                  ERROR   2               ;*NO
013154  105010                          1$:     CLRB    (RO)            ;CLEAR LOW BYTE OF CSR
013156  105005                                  CLRB    R5              ;CLEAR LOW BYTE OF EXPECTED DATA
```

```
013160  011004                                 MOV     (R0),R4              ;READ CSR
013162  020405                                 CMP     R4,R5                ;DOES CSR COMPARE WITH EXPECTED?
013164  001401                                 BEQ     3$                   ;BRANCH IF YES
013166  104002                                 ERROR   2                    ;IF NOT PRINT ERROR
013170  012710  177757                 3$:     MOV     #↑C<DCLR>,(R0)       ;SET ALL CSR BITS POSSIBLE
013174  105077  166612                         CLRB    @#DZVCSR             ;CLEAR HIGH BYTE OF CSR
013200  012705  000150                         MOV     #RIE!MSENAB!MAINT,R5     ;SET EXPECTED IN R5
013204  011004                                 MOV     (R0),R4              ;READ CSR REGISTER ·
013206  020405                                 CMP     R4,R5                ;DOES ACTUAL=EXPECTED
013210  001401                                 BEQ     4$                   ;IF YES CONTINUE
013212  104002                                 ERROR   2                    ;IF NO PRINT ERROR
013214  012710  177757                 4$:     MOV     #↑C<DCLR>,(R0)       ;SET ALL POSSIBLE CSR BITS
013220  005005                                 CLR     R5                   ;SET R5 TO EXPECTED RESULTS
013222  052710  000020                         BIS     #DCLR,(R0)           ;DEVICE MASTER RESET
013226  000240                                 NOP
013230  011004                                 MOV     (R0),R4              ;ACTUAL
013232  020405                                 CMP     R4,R5                ;CMP ACTUAL VS EXPECTED
013234  001401                                 BEQ     2$                   ;YES
013236  104002                                 ERROR   2                    ;*NO
013240                                 2$:
                                               ;************************ TEST 7 *****************************
                                               ;*THIS TEST PERFORMS RESET TESTING AND
                                               ;*TESTING OF READ ONLY REGISTER DZVRBUF
                                               ;*AND TESTING OF WRITE ONLY REGISTER DZVLPR
                                       ;;*  TEST 7
                                       ;;*************************************************************
013240  000004                         TST7:   SCOPE
013242  012737  000007  001246                 MOV     #7,$TSTNM            ;LOAD THE NUMBER OF THIS TEST
013250  012737  013324  001362                 MOV     #TST10,NEXT          ;POINT TO THE START OF THE NEXT TEST
013256  104413                                 DEVICE.CLR                   ;CLEAR DZV11
013260  013700  002014                         MOV     DZVRBUF,R0           ;SET UP FOR ERROR MESSAGE
013264  011005                                 MOV     (R0),R5              ;COPY PRESENT CONTENTS
013266  042705  106000                         BIC     #DVALID!BIT11!BIT10,R5 ;CLEAR ILLEGAL BITS
013272  012777  177777  166520                 MOV     #-1,@DZVLPR          ;TRY TO WRITE ALL 1'S
013300  011004                                 MOV     (R0),R4              ;ACTUAL
013302  020405                                 CMP     R4,R5                ;CMP ACTUAL VS EXPECTED
013304  001401                                 BEQ     1$                   ;IF YES,GO CONTINUE PROCESSING
013306  104002                                 ERROR   2                    ;*ERROR- BIT PATTERN NOT CORRECT
013310  005077  166504                 1$:     CLR     @DZVLPR ;TRY TO WRITE ALL ZEROES
013314  011004                                 MOV     (R0),R4              ;READ REGISTER
013316  020405                                 CMP     R4,R5                ;CMP ACTUAL VS. EXPECTED
013320  001401                                 BEQ     2$                   ;BRANCH IF EQUAL
013322  104002                                 ERROR   2                    ;VALUES DID NOT COMPARE
013324                                 2$:
                                               ;*********************** TEST 10 *****************************
                                               ;*THIS TEST PERFORMS RESET TESTING AND
                                               ;*TESTING OF READ ONLY REGISTER DZVMSR
                                               ;*AND TESTING OF WRITE ONLY REGISTER DZVTDR
                                       ;;*  TEST 10
                                       ;;*************************************************************
013324  000004                         TST10:  SCOPE
013326  012737  000010  001246                 MOV     #10,$TSTNM           ;LOAD THE NUMBER OF THIS TEST
013334  012737  013410  001362                 MOV     #TST11,NEXT          ;POINT TO THE START OF THE NEXT TEST
013342  104413                                 DEVICE.CLR                   ;CLEAR DZV11
013344  013700  002030                         MOV     DZVMSR,R0            ;SET UP FOR ERROR MESSAGE
013350  011005                                 MOV     (R0),R5              ;COPY PRESENT CONTENTS
```

```
013352  042705  170360                   BIC     #170360,R5        ;CLEAR ILLEGAL BITS
013356  112777  177777  166450           MOVB    #-1,@DZVTDR       ;TRY TO WRITE ALL 1'S
013364  011004                           MOV     (R0),R4           ;ACTUAL
013366  020405                           CMP     R4,R5             ;CMP ACTUAL VS EXPECTED
013370  001401                           BEQ     1$                ;IF YES,GO CONTINUE PROCESSING
013372  104002                           ERROR   2                 ;*ERROR- BIT PATTERN NOT CORRECT
013374  005077  166434           1$:     CLR     @DZVTDR ;TRY TO WRITE ALL ZEROES
013400  011004                           MOV     (R0),R4           ;READ REGISTER
013402  020405                           CMP     R4,R5             ;CMP ACTUAL VS. EXPECTED
013404  001401                           BEQ     2$                ;BRANCH IF EQUAL
013406  104002                           ERROR   2                 ;VALUES DID NOT COMPARE
013410                           2$:

                                         ;*********************** TEST 11 ****************************
                                         ;*VERIFY THAT SETTING "DTR" FOR A LINE WILL
                                         ;*BRING UP "CO" AND "RING" FOR:
                                         ;*THE SAME LINE IF IN EXTERNAL MODE
                                         ;*THE STAGGERED LINE IF IN STAGGERED MODE.
                                         ;*LINES ARE STAGGERED AS FOLLOWS:
                                         ;*LINE0 WITH LINE1; LINE2 WITH LINE3.
                                         ;*THIS TEST IS ONLY RUN IF AN H325,OR H329
                                         ;*IS CONNECTED ON THE DZV UNDER TEST.

                                 ;;*  TEST 11
                                 ;;*****************************************************************
013410  000004                   TST11:  SCOPE
013412  012737  000011  001246           MOV     #11,$TSTNM        ;LOAD THE NUMBER OF THIS TEST
013420  012737  013604  001362           MOV     #TST12,NEXT       ;POINT TO THE START OF THE NEXT TEST
013426  005737  001372                   TST     MODE              ;TEST TO SEE IF TESTING WITH
013432  001001                           BNE     8$                ;CONNECTOR
013434  104400                           ADVANCE                   ;IF NO, GO TO NEXT TEST
013436  012737  013526  001364   8$:     MOV     #10$,LOCK         ;SET FOR TIGHT SCOPE LOOP
013444  104413                           DEVICE.CLR                ;SET DCLR IN CSR TO ZERO DEVICE
013446  013700  002030                   MOV     DZVMSR,R0         ;SET REGISTER
013452  005003                           CLR     R3                ;ZERO LINE NUMBER
013454  012702  000001                   MOV     #1,R2             ;SET POINTER
013460  130237  001366           1$:     BITB    R2,LINE           ;TEST THIS LINE?
013464  001003                           BNE     3$                ;YES
013466  005203                   2$:     INC     R3                ;LINE #
013470  104420                           SHIFT                     ;GET NEXT LINE
013472  000772                           BR      1$                ;TEST NEXT LINE
013474  010204                   3$:     MOV     R2,R4             ;SAVE BINARY BIT FOR LINE #
013476  105737  001372                   TSTB    MODE              ;RUNNING IN EXTERNAL MODE?
013502  100406                           BMI     5$                ;IF YES SKIP STAGGERED SETUP
013504  032703  000001                   BIT     #BIT0,R3          ;IF EVEN LINE
013510  001402                           BEQ     4$                ;GO GET ODD PARTNER
013512  006204                           ASR     R4                ;OTHERWISE GET EVEN COMPANION
013514  000401                           BR      5$                ;GO SETUP EXPECTED RESULTS
013516  006304                   4$:     ASL     R4                ;FIND ODD PARTNER
013520  010405                   5$:     MOV     R4,P5             ;LOAD R5 FOR EXPECTED
013522  000305                           SWAB    R5                ;PLACE IN UPPER BYTE
013524  150405                           BISB    R4,R5             ;SET FOR RING BITS
013526  150277  166274           10$:    BISB    R2,@HDZVTCR       ;SET DTR BIT
013532  104414                           DELAY                     ;DELAY FOR CABLE LAG
013534  011004                           MOV     (R0),R4           ;MOVE RESULTS OF MSR REGISTER TO R4
013536  020504                           CMP     R5,R4             ;RESULTS=EXPECTED?
```

```
013540  001401                                    BEQ     6$                  ;IF YES CONTINUE
013542  104002                                    ERROR   2                   ;IF NOT PRINT ERROR RESULTS
013544  104401                           6$:     SCOP1                       ;IS SW09 SET?
013546  012737  013554  001364                    MOV     #11$,LOCK           ;SET UP FOR NEXT TIGHT SCOPE
013554  140277  166246                   11$:     BICB    R2,@HDZVTCR         ;CLEAR  DTR BIT FOR LINE UNDER TEST
013560  104414                                    DELAY                       ;DELAY FOR CABLE LAG
013562  011004                                    MOV     (R0),R4             ;LOAD MSR REGISTER INTO R4
013564  001402                                    BEQ     7$                  ;IF CO AND RING CLEARED CONTINUE
013566  005005                                    CLR     R5                  ;OTHERWISE SET EXPECTED FOR ERROR
013570  104002                                    ERROR   2                   ;PRINTOUT
013572  104401                            7$:     SCOP1                       ;IS SW09 SET?
013574  012737  013526  001364                    MOV     #10$,LOCK           ;RESET TIGHT SCOPE LOOP
013602  000731                                    BR      2$                  ;GET NEXT LINE

                                                  ;****************** ***** TEST 12 ******************************
                                                  ;* THIS TEST VERIFI.S THAT TRDY IS SET WHEN A LINE
                                                  ;* IS READY TO BE LOADED, AND THAT THE LINE SPECI-
                                                  ;* FIED IN BITS 8-9 OF DZVCSR CORRESPOND
                                                  ;* TO THE LINE SELECTED IN DZVTCR
                                          ;;*  TEST 12
                                          ;;**********************************************************
013604  000004                            TST12:  SCOPE
013606  012737  000012  001246                    MOV     #12,$TSTNM          ;LOAD THE NUMBER OF THIS TEST
013614  012737  013736  001362                    MOV     #TST13,NEXT         ;POINT TO THE START OF THE NEXT TEST
013622  104413                                    DEVICE.CLR                  ;ISSUE A "DEVICE CLEAR" (RESET)
013624  012737  013660  001364                    MOV     #2$,LOCK            ;SET UP FOR TIGHT SCOPE LOOP
013632  005037  001374                            CLR     SAVLIN              ;INITIALIZE FOR ERROR PRINTOUT
013636  013700  002010                            MOV     DZVCSR,R0           ;SET POINTER
013642  012705  100040                            MOV     #MSENAB!TRDY,R5     ;START THE EXPECTED LINE NUMBER AT 0
013646  012702  000001                            MOV      #1,R2              ;USING R2 AS A BIT POINTER, POINT TO LINE 0
013652  130237  001366                    1$:     BITB    R2,LINE             ;IS THIS LINE SELECTED?
013656  001421                                    BEQ     6$                  ;IF NO, SKIP THE STARTUP
013660  050277  166140                    2$:     BIS     R2,@DZVTCR          ;SET THE GO BIT FOR THIS LINE
013664  052710  000040                            BIS     #MSENAB,(R0)        ;START THE SCANNER
013670  005004                                    CLR     R4                  ;SET FOR DELAY
013672  005710                            3$:     TST     (R0)                ;TX READY?
013674  100404                                    BMI     4$                  ;BR IF YES
013676  104414                                    DELAY                       ;DELAY
013700  005204                                    INC     R4                  ;COUNTER
013702  001373                                    BNE     3$                  ;BR IF <>0!
013704  104003                                    ERROR   3                   ;*TX NOT READY!
013706  011004                            4$:     MOV     (R0),R4             ;GET THE LINE POINTED TO BY THE SCANNER
013710  020405                                    CMP     R4,R5               ;IS THE LINE NUMBER WHAT IT SHOULD BE?
013712  001 '01                                   BEQ     5$                  ;IF YES,GO WORK ON THE NEXT LINE
013714  104002                                    ERROR   2                   ;*LINE NUMBER DID NOT MATCH TCR BIT
013716  104401                            5$:     SCOP1                       ;IS SW09 SET?
013720  104413                                    DEVICE.CLR                  ;SET DCLR IN CSR;SETUP FOR NEXT LINE
013722  062705  000400                    6$:     ADD     #400,R5             ;POINT TO THE NEXT EXPECTED LINE
013726  104420                                    SHIFT                       ;POINT TO THE NEXT LINE.ARE ALL LINES TESTED?
013730  005237  001374                            INC     SAVLIN              ;ADJUST FOR ERROR PRINTOUT
013734  000746                                    BR      1$                  ;IF NOT, GO DO THE NEXT LINE
                                                  ;************************* TEST 13 *****************************
                                                  ;*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 .
```

```
                                                ;*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 13
                                        ;;*************************************************************
013736  000004                          TST13:  SCOPE
013740  012737  000013  001246                  MOV     #13,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
013746  012737  014226  001362                  MOV     #TST14,NEXT     ;POINT TO THE START OF THE NEXT TEST
013754  012737  014210  001364                  MOV     #16$,LOCK       ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELEC
013762  104417                                  DCLASM                  ;SET DCLR IN CSR AND SET MAINT MODE
013764  104421                                  LPRSET                  ;LOAD LPR REGISTER FOR ALL LINES
013766  005037  001374                          CLR     SAVLIN          ;INIT. FOR ERROR PRINTOUT
013772  105037  001425                          CLRB    DONFLG          ;INIT FOR TCR BIT HANDLER
013776  012702  000001                          MOV     #1,R2           ;LINE POINTER
014002  012701  000252                          MOV     #252,R1         ;SAVE CHARACTER TO BE TRANSMITTED
014006  052777  000040  165774                  BIS     #MSENAB,@DZVCSR ;START SCANNER
014014  030237  001366                  3$:     BIT     R2,LINE         ;VALID LINE ?
014020  001467                                  BEQ     15$             ;NO SET UP NEXT LINE
014022  010277  165776                          MOV     R2,@DZVTCR      ;SET TCR BIT
014026  005005                          5$:     CLR     R5              ;SET R5 FOR A DELAY LOOP
014030  105777  165754                          TSTB    @DZVCSR         ;IS REC DONE = 0 ?
014034  100001                                  BPL     6$              ;IF YES, ALLOW TIME FOR TRDY TO SET
014036  104020                                  ERROR   20              ;*REC DONE SHOULD = 0
014040  005777  165744                  6$:     TST     @DZVCSR         ;TRDY SET?
014044  100404                                  BMI     7$              ;IF YES BRANCH
014046  104414                                  DELAY                   ;IF NO THEN WAIT FOR IT
014050  005205                                  INC     R5              ;DELAY LOOP
014052  001372                                  BNE     6$              ;BRANCH BACK AND TEST AGAIN
014054  104003                                  ERROR   3               ;*TRDY FAILED TO SET!
014056  105737  001425                  7$:     TSTB    DONFLG          ;HAVE WE ALREADY SENT CHARAC.
014062  001041                                  BNE     13$             ;IF YES GO CLEAR TCR BIT
014064  105237  001425                          INCB    DONFLG          ;IF NOT INDICATE HAVING BEEN HERE
014070  110177  165740                          MOVB    R1,@DZVTDR      ;LOAD CHARACTER
014074  013705  001374                          MOV     SAVLIN,R5       ;MAKE EXPECTED LINE #
014100  005737  001372                          TST     MODE            ;IS THIS TEST IN STAGGERED MODE?
014104  100006                                  BPL     10$             ;IF NOT, SKIP STAGGERED SETUP

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

014106  006205                                  ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
014110  103402                                  BCS     8$              ;IF IT IS SET, GO CLEAR IT
014112  000261                                  SEC                     ;IF IT IS CLEAR SET IT HERE
014114  000401                                  BR      9$              ;SKIP THE CLEARING
014116  000241                          8$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
014120  006105                          9$:     ROL     R5              ;GET THE NEW BIT BACK INTO R5
014122  000305                          10$:    SWAB    R5              ;MOVE THE LINE NUMBER TO THE UPPER BYTE
014124  150105                                  BISB    R1,R5           ;ADD CHARACTER
014126  052705  100000                          BIS     #DVALID,R5      ;ADD DATA VALID
014132  005003                                  CLR     R3
014134  105777  165650                  11$:    TSTB    @DZVCSR         ;IS RDONE SET?
014140  100404                                  BMI     12$             ;IF YES GO GET CHAR.
014142  104414                                  DELAY                   ;IF NOT THEN WAIT
014144  005203                                  INC     R3              ;DELAY LOOP
014146  001372                                  BNE     11$             ;DELAY DONE?
014150  104004                                  ERROR   4               ;*RDONE FAILED TO SET!
```

```
014152  017704  165636                   12$:    MOV     @DZVRBUF,R4      ;LOAD THE VALUE ACTUALLY RECEIVED
014156  020405                                   CMP     R4,R5            ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
014160  001722                                   BEQ     5$               ;IF YES, GO DO THE NEXT LINE
014162  104006                                   ERROR   6                ;*NO DATA/CONTENTS DID NOT COMPARE
014164  000720                                   BR      5$               ;GO BACK AND WAIT TO CLEAR TCR BIT
014166  104401                           13$:    SCOP1                    ;CHECK TO SEE IF SWITCH NINE IS SET
014170  105037  001425                           CLRB    DONFLG           ;SET UP FOR NEXT LINE
014174  005077  165624                           CLR     @DZVTCR          ;CLEAR PREVIOUS TCR BIT
014200  005237  001374                   15$:    INC     SAVLIN           ;SET LINE INDICATOR FOR NEXT LINE
014204  104420                                   SHIFT                    ;CALCULATE NEXT LINE
014206  000702                                   BR      3$               ;GET GET STARTED

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

014210  005777  165574                   16$:    TST     @DZVCSR          ;IS TRANSMITTER READY?
014214  100375                                   BPL     16$              ;IF NOT, WAIT FOR IT
014216  110177  165612                           MOVB    R1,@DZVTDR       ;LOAD THE CHARACTER
014222  104401                                   SCOP1                    ;LOOP AGIN IF SW09=1
014224  000760                                   BR      13$              ;OTHERWISE, GO PICK UP THE TEST NORMALLY

                                                 ;*********************** TEST 14 ****************************
                                                 ;*THIS TEST VERIFIES THAT EACH RECEIVING LINE CAN BE
                                                 ;*DISABLED BY SETTING RCVON (BIT12 IN THE LPR REGISTER)
                                                 ;*TO ZERO FOR EACH LINE.
                                                 ;*THIS TEST ALSO VERIFIES THAT THE SILO CAN BE
                                                 ;*EMPTIED BY ISSUING A DEVICE MASTER CLEAR.
                                         ;;*  TEST 14
                                         ;;***********************************************************
014226  000004                           TST14:  SCOPE
014230  012737  000014  001246                   MOV     #14,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
014236  012737  014550  001362                   MOV     #TST15,NEXT      ;POINT TO THE START OF THE NEXT TEST
014244  105037  001425                           CLRB    DONFLG           ;CLEAR TEST CONTROL FLAG
014250  005037  001374                           CLR     SAVLIN           ;CLEAR LINE INDICATOR
014254  104417                                   DCLASM                   ;ISSUE A DEVICE MASTER CLEAR
                                                                          ;AND SET MAINT BIT IF NECESSARY
014256  013701  001370                           MOV     PAR,R1           ;SAVE DEFAULT PARAMETERS
014262  042737  010000  001370                   BIC     #RCVON,PAR       ;DISABLE RECEIVER IN DEFAULT PAR.
014270  10442'                           100$:   LPRSET                   ;LOAD PARAMETERS IN LPR REGISTER
014272  0101,   001370                           MOV     R1,PAR           ;RESTORE DEFAULT PARAMETERS
014276  012701  000252                           MOV     #252,R1          ;LOAD A CHARAC. INTO R1
014302  013702  001366                           MOV     LINE,R2          ;COPY AN IMAGE OF THE ACTIVE LINES
014306  010277  165512                           MOV     R2,@DZVTCR       ;SET TCR BITS FOR ALL ACTIVE LINES
014312  052777  000040  165470                   BIS     #MSENAB,@DZ/CSR  ;SET MASTER SCAN ENABLE
014320  005005                           1$:     CLR     R5               ;INIT DELAY COUNTER
014322  005777  165462                   2$:     TST     @DZVCSR          ;IS TRANS READY SET?
014326  100404                                   BMI     3$               ;BRANCH IF YES
014330  104414                                   DELAY                    ;WAIT FOR TRDY TO SET
014332  005205                                   INC     R5               ;INCREMENT DELAY COUNTER
014334  001372                                   BNE     2$               ;RETURN TO CHECK TRDY
014336  104003                                   ERROR   3                ;TRDY FAILED TO SET!
014340  117705  165446                   3$:     MOVB    @HDZVCSR,R5      ;MOVE LINE NO. TO R5
014344  012703  000001                           MOV     #1,R3            ;INIT TCR POINTER
014350  042705  177774                           BIC     #↑C<3>,R5        ;ISOLATE LINE NO.
014354  001403                                   BEQ     31$              ;IF LINE 0 BRANCH
014356  106303                           30$:    ASLB    R3               ;SHIFT R3 POINTER TO NEXT LINE
014360  005305                                   DEC     R5               ;DECREMENT LINE NO.
```

```
014362  001375                   BNE     30$               ;WHEN R5=0, R3 POINTS TO LINE TCR
014364  030302           31$:    BIT     R3,R2             ;HAS CHARACTER BEEN SENT?
014366  001007                   BNE     4$                ;BRANCH IF NO
014370  140377  165430           BICB    R3,@DZVTCR        ;IF YES THEN CLEAR TCR BIT
014374  001351                   BNE     1$                ;IF ALL CHARAC. SENT DROP THROUGH
014376  105737  001425           TSTB    DONFLG            ;IF NO MORE ACTIVE IS THIS SECOND
                                                           ;TIME HERE?
014402  001037                   BNE     10$               ;IF YES SKIP TO SECOND PART OF TEST
014404  000404                   BR      5$                ;IF FIRST TIME HERE GO ZERO TCR BITS
014406  110177  165422   4$:     MOVB    R1,@DZVTDR        ;LOAD CHAR. INTO BUFFER
014412  040302                   BIC     R3,R2             ;INDICATE CHARAC. SENT ON THIS LINE
014414  000741                   BR      1$                ;GO BACK AND WAIT FOR TRDY TO SET
014416  005077  165402   5$:     CLR     @DZVTCR           ;CLEAR OUT TCR BITS
014422  005005                   CLR     R5                ;INIT DELAY COUNTER
014424  105777  165360   6$:     TSTB    @DZVCSR           ;IS RECEIV. DONE SET?
014430  100002                   BPL     7$                ;IF NOT THEN WAIT TO SEE IF IT WILL
014432  104020                   ERROR   20                ;REC DONE SHOULD NOT SET!
014434  000403                   BR      8$                ;GO FIND WHICH LINE RECEIVED
014436  104414           7$:     DELAY                     ;STALL FOR RECEIVER
014440  005205                   INC     R5                ;INCREMENT DELAY COUNTER
014442  001370                   BNE     6$                ;IF NOT DONE GO RETEST REC DONE
014444  017704  165344   8$:     MOV     @DZVRBUF,R4       ;READ REC. BUFFER
014450  100007                   BPL     9$                ;IS DVALID SET?
014452  000304                   SWAB    R4                ;IF YES GET LINE NO.
014454  042704  177774           BIC     #↑C<3>,R4         ;ISOLATE LINE NO.
014460  010437  001374           MOV     R4,SAVLIN         ;SET UP LINE NO. FOR ERROR REPORT
014464  104017                   ERROR   17                ;DVALID SHOULD NOT BE SET
014466  000766                   BR      8$                ;GO CHECK FOR ANY OTHER CHAR. IN SILO
014470  105237  001425   9$:     INCB    DONFLG            ;INDICATE THAT FIRST PART OF TEST IS DONE
014474  013701  001370           MOV     PAR,R1            ;SAVE DEFAULT LINE PARAM.
014500  000673                   BR      100$              ;NOW GO RELOAD LPR REGISTER TO
                                                           ;TURN RECEIVERS ON
014502  005005           10$:    CLR     R5                ;ZERO DELAY COUNTER
014504  104414           11$:    DELAY                     ;WAIT FOR ALL CHARAC. TO BE RECEIVED
014506  005205                   INC     R5                ;INCREASE DELAY COUNT
014510  001375                   BNE     11$               ;CONT. DELAY IF NOT FINISHED
014512  104413                   DEVICE.CLR                ;ISSUE A MASTER CLEAR
014514  000240                   NOP
014516  000240                   NOP
014520  105777  165264           TSTB    @DZVCSR           ;NOW IS RECEIV. DONE SET?
014524  100003                   BPL     12$               ;BRANCH IF NO
014526  005037  001374           CLR     SAVLIN            ;CLEAR LINE NO FOR ERROR REPORT
014532  104020                   ERROR   20                ;REC. DONE SHOULD NOT BE SET!
014534  017704  165254   12$:    MOV     @DZVRBUF,R4       ;READ REC. BUFFER
014540  100003                   BPL     13$               ;IS DVALID SET? IT SHOULDN'T BE
014542  005037  001374           CLR     SAVLIN            ;DEVICE. CLR DID NOT ZERO SILO
014546  104017                   ERROR   17                ;PRINT OUT THE ERROR.(LINE NO. IS IRRELEVANT)
014550                   13$:

                                 ;************************ TEST 15 ******************************
                                 ;* 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 15
                         ;;****************************************************************
```

```
014550  000004                              TST15:  SCOPE
014552  012737  000015  001246                      MOV     #15,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
014560  012737  015040  001362                      MOV     #TST16,NEXT      ;POINT TO THE START OF THE NEXT TEST
014566  012737  014654  001364                      MOV     #5$,LOCK         ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELEC
014574  104417                                      DCLASM                   ;SET DCLR AND SET MNTFLG
014576  104421                                      LPRSET                   ;LOAD LPR REGISTER FOR ALL LINES
014600  005037  001374                              CLR     SAVLIN           ;INIT FOR FIRST LINE
014604  104422                                      BUFSET                   ;ZERO BUFFER AREA
014606  105037  001425                              CLRB    DONFLG           ;ZERO TCR BIT HANDLER FLAG
014612  012702  000001                              MOV     #1,R2            ;LINE POINTER
014616  052777  000040  165164                      BIS     #MSENAB,@DZVCSR  ;START SCANNER
014624  030237  001366                      3$:     BIT     R2,LINE          ;VALID LINE ?
014630  001477                                      BEQ     15$              ;NO SET UP NEXT LINE
014632  010277  165166                              MOV     R2,@DZVTCR       ;SET TCR BIT
014636  013700  001374                              MOV     SAVLIN,R0        ;ADJUST BUFFER POINTER
014642  006300                                      ASL     R0               ;OFFSET
014644  105777  165140                      4$:     TSTB    @DZVCSR          ;IS REC DONE = 0 ?
014650  100001                                      BPL     5$               ;IF YES, ALLOW TIME FOR TRDY TO SET
014652  104020                                      ERROR   20               ;*REC DONE SHOULD = 0
014654  005005                              5$:     CLR     R5               ;USE R5 AS TIMER WAITING FOR TRDY TO SET
014656  005777  165126                      6$:     TST     @DZVCSR          ;IS THE TRANSMITTER READY?
014662  100404                                      BMI     7$               ;IF SO, GO TRANSMIT A CHARACTER
014664  104414                                      DELAY                    ;WAIT A LITTLE BIT
014666  005205                                      INC     R5               ;UP THE LOCAL COUNTER.TIME EXCEEDED?
014670  001372                                      BNE     6$               ;IF NOT, GO TRY AGAIN
014672  104003                                      ERROR   3                ;*TRDY FAILED TO SET!
014674  105737  001425                      7$:     TSTB    DONFLG           ;ALL CHARAC. TRANS.?
014700  001047                                      BNE     14$              ;IF YES GO ZERO TCR BIT
014702  116077  001426  165124                      MOVB    TDO(R0),@DZVTDR  ;LOAD CHARACTER
014710  013705  001374                              MOV     SAVLIN,R5        ;MAKE EXPECTED LINE #
014714  005737  001372                              TST     MODE             ;IS THIS TEST IN STAGGERED MODE?
014720  100006                                      BPL     10$              ;IF NOT, SKIP STAGGERED SETUP

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

014722  006205                                      ASR     R5               ;GET THE LAST BIT INTO THE CARRY BIT
014724  103402                                      BCS     8$               ;IF IT IS SET, GO CLEAR IT
014726  000261                                      SEC                      ;IF IT IS CLEAR SET IT HERE
014730  000401                                      BR      9$               ;SKIP THE CLEARING
014732  000241                              8$:     CLC                      ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
014734  006105                              9$:     ROL     R5               ;GET THE NEW BIT BACK INTO R5
014736  000305                              10$:    SWAB    R5               ;MOVE THE LINE NUMBER TO THE UPPER BYTE
014740  156005  001426                              BISB    TDO(R0),R5       ;ADD CHARACTER
014744  052705  100000                              BIS     #DVALID,R5       ;ADD DATA VALID
014750  005003                                      CLR     R3
014752  105777  165032                      11$:    TSTB    @DZVCSR          ;REC DONE?
014756  100404                                      BMI     12$              ;IF YES GO CHECK CHAR.
014760  104414                                      DELAY                    ;IF NOT WAIT FOR REC.
014762  005203                                      INC     R3               ;DELAY LOOP TIMER
014764  001372                                      BNE     11$              ;DELAY FINISHED?
014766  104004                                      ERROR   4                ;*RDONE FAILED TO SET!
014770  017704  165020                      12$:    MOV     @DZVRBUF,R4      ;LOAD THE VALUE ACTUALLY RECEIVED
014774  020405                                      CMP     R4,R5            ;COMPARE ACTUAL VS EXPECTED. ARE THEY THE SAME?
014776  001401                                      BEQ     13$              ;IF YES, GO DO THE NEXT LINE
015000  104006                                      ERROR   6                ;*NO DATA/CONTENTS DID NOT COMPARE
015002  104401                              13$:    SCOP1                    ;CHECK TO SEE IF SWITCH NINE IS SET
```

```
015004  105260  001426                  INCB    TDO(RO)         ;INCREMENT BINARY PATTERN FOR THIS LINE
015010  001315                          BNE     4$              ;GO 'ROUND AGAIN FOR NEXT CHARACTER
015012  105237  001425                  INCB    DONFLG          ;INDICATE ALL CHAR. SENT
015016  000712                          BR      4$              ;BRANCH TO CLEAR TCR BIT
015020  005077  165000          14$:    CLR     @DZVTCR         ;CLEAR TCR REGISTER
015024  105037  001425                  CLRB    DONFLG          ;INIT FOR NEXT LINE
015030  005237  001374          15$:    INC     SAVLIN          ;INC EXPECTED LINE
015034  104420                          SHIFT                   ;SHIFT THE LINE POINTER. ARE WE ALL DONE?
015036  000672                          BR      3$              ;IF NO, GO AROUND AGAIN FOR NEXT LINE


                                        ;************************ TEST 16 ******************************
                                        ;*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.
                                ;:*  TEST 16
                                ;;***************************************.*************************
015040  000004                  TST16:  SCOPE
015042  012737  000016  001246          MOV     #16,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
015050  012737  015242  001362          MOV     #TST17,NEXT     ;POINT TO THE START OF THE NEXT TEST
015056  012737  015166  001364          MCV     #5$,LOCK        ;SET FOR LOOP
015064  005037  001374                  CLR     SAVLIN          ;INIT LINE INDIC. FOR ERROR PRINTOUT
015070  012702  000001                  MOV     #1,R2           ;LINE POINTER
015074  030237  001366          1$:     BIT     R2,LINE         ;VALID LINE?
015100  001454                          BEQ     9$              ;IF NOT SET FOR NEXT LINE
015102  104417                          DCLASM                  ;SET DCLR IN CSR AND SET MNTFLG
015104  013701  001370                  MOV     PAR,R1          ;PICK UP PARAMETERS
015110  052737  000300  001370          BIS     #ODDPAR!PARITY,PAR ;FORCE ODD PARITY
015116  104421                          LPRSET                  ;LOAD LPR REGISTER
015120  010137  001370                  MOV     R1,PAR          ;RESET PAR TO ORIGINAL VALUE
015124  052777  000040  164656          BIS     #MSENAB,@DZVCSR ;START SCANNER
015132  013705  001374                  MOV     SAVLIN,R5       ;MAKE EXPECTED DATA
015136  005737  001372                  TST     MODE            ;IS THIS TEST IN STAGGERED MODE?
015142  100006                          BPL     4$              ;IF NOT, SKIP STAGGERED SETUP

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

015144  006205                          ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
015146  103402                          BCS     2$              ;IF IT IS SET, GO CLEAR IT
015150  000261                          SEC                     ;IF IT IS CLEAR SET IT HERE
015152  000401                          BR      3$              ;SKIP THE CLEARING
015154  000241                  2$:     CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
015156  006105                  3$:     ROL     R5              ;GET THE NEW BIT BACK INTO R5
015160  000305                  4$:     SWAB    R5              ;PUT LINE NUMBER IN UPPER BYTE
015162  052705  130000                  BIS     #DVALID!PARER!FRMERR,R5 ;ADD EXPECTED
015166  005003                  5$:     CLR     R3              ;INIT DELAY ACCUMULATOR
015170  110277  164642                  MOVB    R2,@HDZVTDR     ;SET BREAK BIT
015174  105777  164610          6$:     TSTB    @DZVCSR         ;RECEIVER DONE?
015200  100404                          BMI     7$              ;BRANCH IF YES
015202  104414                          DELAY                   ;WAIT FOR REC DONE TO SET
015204  005203                          INC     R3              ;INC DELAY LOOP
015206  001372                          BNE     6$              ;DELAY FINISHED?
015210  104004                          ERROR   4               ;*RDONE FAILED TO SET!
015212  017704  164576          7$:     MOV     @DZVRBUF,R4     ;ACTUAL
```

```
015216  020405                                  CMP     R4,R5           ;CMP ACTUAL VS EXPECTED. DO THEY MATCH?
015220  001401                                  BEQ     8$              ;IF YES, GO CLEAN UP
015222  104006                                  ERROR   6               ;*DATA/CONTENTS FAILED TO COMPARE
015224  105077  164606                  8$:     CLRB    @HDZVTDR        ;CLEAR BREAK BITS
015230  104401                                  SCOP1                   ;LOOP?
015232  005237  001374                  9$:     INC     SAVLIN          ;INC LINE #
015236  104420                                  SHIFT                   ;SET R2 TO NEXT LINE
015240  000715                                  BR      1$              ;GO BACK AND TEST NEXT LINE
                                                ;************************ TEST 17 ******************************
                                                ;* THIS TEST VERIFIES THAT THE DEVICE DOES NOT INTERRUPT
                                                ;*WHILE THE PROCESSOR STATUS DOES NOT ALLOW INTERRUPTS
                                                ;*BUT WILL INTERRUPT IF THE PROCESSOR STATUS
                                                ;*ALLOWS INTERRUPTS.
                                        ;;*  TEST 17
                                        ;;*************************************************************
015242  000004                          TST17:  SCOPE
015244  012737  000017  001246                  MOV     #17,$TSTNM      ;LOAD THE NUMBER OF THIS TEST
015252  012737  015636  001362                  MOV     #TST20,NEXT     ;POINT TO THE START OF THE NEXT TEST
015260  104417                                  DCLASM                  ;SET DCLR IN CSR AND SET MAINT BIT
                                                                        ;IF NECESSARY (INTERNAL MODE)
015262  104421                                  LPRSET                  ;SET UP LPR REGISTER
015264  005037  001374                          CLR     SAVLIN          ;INIT LINE INDIC. FOR ERROR
015270  105037  001425                          CLRB    DONFLG          ;INIT TCR BIT HANDLER FLAG
015274  113777  001366  164522                  MOVB    LINE,@DZVTCR    ;SET ALL VALID TCR BITS
015302  106427  000200                          MTPS    #MASK           ;SET CPU STATUS TO DZV11 PRIO,
015306  012777  000200  164526                  MOV     #MASK,@DZVRIS   ;SET RECEIVER STATUS
015314  012777  000200  164524                  MOV     #MASK,@DZVTIS   ;SET TRANSMITTER STATUS
015322                                  1$:
015322  012777  015410  164514                  MOV     #6$,@DZVTIV     ;SET UP THE TRANSMITTER INTERRUPT VECTOR
015330  012777  015432  164502                  MOV     #7$,@DZVRIV     ;SET UP THE RECEIVER INTERRUPT VECTOR
015336  012777  000200  164476                  MOV     #MASK,@DZVRIS   ;SET THE INTERRUPT VECTOR STATUS
015344  012777  000200  164474                  MOV     #MASK,@DZVTIS   ;SET TRANSMITTER INTERRUPT PRIORITY
015352  052777  040040  164430                  BIS     #TIE!MSENAB,@DZVCSR ;ENABLE THE DEVICE
015360  005005                                  CLR     R5              ;INIT DELAY COUNTER
015362  005777  164422                  4$:     TST     @DZVCSR         ;TRDY SET?
015366  100003                                  BPL     5$              ;IF NOT GO DO DELAY
015370  000240                                  NOP                     ;WAIT FOR INTERRUPT
015372  000240                                  NOP
015374  000420                                  BR      8$              ;GO CLEAR TIE BIT
015376  104414                          5$:     DELAY                   ;DELAY ROUTINE CALL
015400  005205                                  INC     R5              ;INC DELAY COUNTER
015402  001367                                  BNE     4$              ;DELAY FINISHED?
015404  104003                                  ERROR   3               ;*TRDY NOT SET!
015406  000413                                  BR      8$              ;GO CLEAR TIE
015410  022626                          6$:     POP2SP                  ;REMOVE THE INTERRUPT FROM THE STACK
015412  042777  040000  164370                  BIC     #TIE,@DZVCSR    ;DON'T LET ANY MORE INTERRUPTS OCCUR
015420  105737  001425                          TSTB    DONFLG          ;PROCESSOR ALLOWING INTER?
015424  001013                                  BNE     10$             ;IF YES NO ERROR
015426  104010                                  ERROR   10              ;IF NOT PRINT ERROR
015430  000413                                  BR      9$              ;RETURN TO THE NORMAL FLOW
015432  104012                          7$:     ERROR   12              ;*RECEIVER SHOULD NOT INTERRUPT
015434  022626                                  POP2SP                  ;POP FOR FAKE RTI
015436  042777  040000  164344          8$:     BIC     #TIE,@DZVCSR    ;RESET TRANSMITTER INTERRUPT ENABLE
015444  105737  001425                          TSTB    DONFLG          ;INTERRUPTS ENABLED?
015450  001403                                  BEQ     9$              ;IF NOT GET OUT
015452  104007                                  ERROR   7               ;IF YES TRANS FAILED TO INTER.
```

```
015454  106427  000000                   10$:    MTPS    #CLEAR                  ;ALLOW INTERRUPTS
015460                                   9$:
015460  012777  015564  164356                   MOV     #11$,@DZVTIV            ;SET UP THE TRANSMITTER INTERRUPT VECTOR
015466  012777  015570  164344                   MOV     #12$,@DZVRIV            ;SET UP THE RECEIVER INTERRUPT VECTOR
015474  012777  000200  164340                   MOV     #MASK,@DZVRIS           ;SET THE INTERRUPT VECTOR STATUS
015502  012777  000200  164336                   MOV     #MASK,@DZVTIS           ;SET TRANSMITTER INTERRUPT PRIORITY
015510  052777  000140  164272                   BIS     #RIE!MSENAB,@DZVCSR ;ENABLE THE DEVICE
015516  113777  001426  164310                   MOVB    TDO,@DZVTDR             ;LOAD BUFFER WITH ANY CHAR.
015524  005005                                   CLR     R5                      ;INIT DELAY ACCUMULATOR
015526  105777  164256                   13$:    TSTB    @DZVCSR                 ;REC. DONE?
015532  100003                                   BPL     14$                     ;IF NOT DELAY
015534  000240                                   NOP                             ;WAIT FOR INTERRUPT
015536  000240                                   NOP
015540  000404                                   BR      18$
015542  104414                           14$:    DELAY                           ;DELAY FOR INTERRUPT
015544  005205                                   INC     R5                      ;INCREMENT DELAY COUNTER
015546  001367                                   BNE     13$                     ;DELAY FINISHED?
015550  104004                                   ERROR   4                       ;*NO RX DONE! (NOT SET)
015552  105737  001425                   18$:    TSTB    DONFLG                  ;PROCESSOR ALLOWING INTERRUPTS?
015556  001411                                   BEQ     15$                     ;IF NOT DON'T PRINT ERROR
015560  104011                                   ERROR   11                      ;RECEIVER FAILED TO INTERRUPT
015562  000407                                   BR      15$                     ;CONTINUE TEST
015564  104010                           11$:    ERROR   10                      ;TRANSMITTER SHOULD NOT INTER.
015566  000404                                   BR      16$                     ;CONT TEST
015570  105737  001425                   12$:    TSTB    DONFLG                  ;PROCESSOR ALLOWING INTERRUPTS?
015574  001001                                   BNE     16$                     ;IF YES DON'T PRINT ERROR
015576  104012                                   ERROR   12                      ;*RECEIVER SHOULD NOT INTERRUPT
015600  022626                           16$:    POP2SP                          ;POP FOR FAKE RTI
015602  042777  040100  164200           15$:    BIC     #RIE!TIE,@DZVCSR              ;CLEAR INTERRUPTS
015610  105737  001425                           TSTB    DONFLG                  ;SECOND TIME THROUGH?
015614  001005                                   BNE     17$                     ;IF YES LEAVE TEST
015616  105237  001425                           INCB    DONFLG                  ;IF NO INDICATE SECOND TEST PASS
015622  106427  000000                           MTPS    #CLEAR                  ;ALLOW INTERRUPTS
015626  000635                                   BR      1$                      ;RESTART TEST
015630  106427  000200                   17$:    MTPS    #MASK                   ;DON'T ALLOW INTERRUPTS
015634  104413                                   DEVICE.CLR                      ;CLEAR DEVICE, LEAVE TEST

                                                 ;*********************** TEST 20 *****************************
                                                 ;*THIS TEST VERIFIES THAT THE RECEIVER WILL
                                                 ;*INTERRUPT BEFORE THE TRANSMITTER EVEN
                                                 ;*THOUGH THE TRANSMITTER WAS ENABLED
                                                 ;*FIRST.  SET PS TO HIGH (MASK INTERRUPTS);
                                                 ;*GET RDONE AND TRDY TO SET;
                                                 ;*SET TX IE AND RX IE;
                                                 ;*CLEAR PS AND EXPECT RX TO INTERRUPT FIRST
                                         ;:*  TEST 20
                                         ;:***********************************************************
015636  000004                           TST20:  SCOPE
015640  012737  000020  001246                   MOV     #20,$TSTNM              ;LOAD THE NUMBER OF THIS TEST
015646  012737  004064  001362                   MOV     #$EOP,NEXT              ;POINT TO THE END-OF-PASS HANDLER
015654  104417                                   DCLASM                          ;SET DCLR IN CSR AND MNTFLG
015656  104421                                   LPRSET                          ;LOAD PAR REGISTER FOR ALL LINES
015660  005037  001374                           CLR     SAVLIN                  ;INIT. ERROR LINE INDIC.
015664  012777  016074  164146                   MOV     #8$,@DZVRIV             ;SETUP INTERRUPT STUFF
015672  012777  000200  164142                   MOV     #MASK,@DZVRIS           ;
015700  012777  016162  164136                   MOV     #12$,@DZVTIV            ;
```

```
015706  012777  000200  164132                  MOV     #MASK,@DZVTIS    ;
015714  052777  000040  164066                  BIS     #MSENAB,@DZVCSR
015722  012702  000001                          MOV     #1,R2            ;LINE POINTER
015726  030237  001366                  3$:     BIT     R2,LINE          ;VALID LINE ?
015732  001515                                  BEQ     14$              ;IF NOT GO TO NEXT LINE
015734  106427  000200                  4$:     MTPS    #MASK
015740  110277  164060                          MOVB    R2,@DZVTCR       ;SET TCR BIT
015744  005777  164044                          TST     @DZVRBUF         ;VALID DATA?
015750  100001                                  BPL     .+4              ;IT BETTER NOT BE SET
015752  104017                                  ERROR   17               ;DATA VALID SHOULD NOT BE SET
015754  105777  164030                  5$:     TSTB    @DZVCSR          ;RECEIVER DONE ?
015760  100001                                  BPL     .+4
015762  104020                                  ERROR   20               ;RECEIVER DONE BIT SHOULD NOT BE SET
015764  005005                                  CLR     R5
015766  005004                                  CLR     R4
015770  005777  164014                  99$:    TST     @DZVCSR          ;WAIT FOR TRDY
015774  100404                                  BMI     100$             ;BR IF READY
015776  104414                                  DELAY                    ;STALL TIME
016000  005204                                  INC     R4               ;
016002  001372                                  BNE     99$
016004  104003                                  ERROR   3                ;TRDY FAILED TO SET
016006  105077  164022                  100$:   CLRB    @DZVTDR          ;SEND A ZERO CHARACTER
016012  005004                                  CLR     R4
016014  105777  163770                  6$:     TSTB    @DZVCSR          ;IS RDONE SET?
016020  100404                                  BMI     7$
016022  104414                                  DELAY
016024  005204                                  INC     R4
016026  001372                                  BNE     6$
016030  104004                                  ERROR   4                ;*RDONE FAILED TO SET!
016032  005777  163752                  7$:     TST     @DZVCSR          ;TRANS DONE BIT = 1 ?
016036  100401                                  BMI     .+4              ;YES
016040  104003                                  ERROR   3                ;*NO   TRANS DONE FAILED TO SET
                                                ;NOW THAT BOTH TRANSMITTER AND RECEIVER DONE BIT =1
                                                ;SET INTERRUPT ENABLES
016042  052777  040000  163740                  BIS     #TIE,@DZVCSR
016050  052777  000100  163732                  BIS     #RIE,@DZVCSR
016056  106427  000000                          MTPS    #CLEAR           ;ALLOW THE INTERRUPTS
016062  000240                                  NOP
016064  000240                                  NOP
016066  104007                                  ERROR   7                ;*TRANSMITTER FAILED TO INTERRUPT
016070  104011                                  ERROR   11               ;*RECEIVER FAILED TO INTERRUPT
016072  000435                                  BR      14$              ;GET OUT

                                                ;RECEIVER INTERRUPT ROUTINE
016074  017704  163714                  8$:     MOV     @DZVRBUF,R4      ;ACTUAL
016100  010403                                  MOV     R4,R3
016102  000303                                  SWAB    R3
016104  042703  177770                          BIC     #↑C<7>,R3        ;STRIP JUNK
016110  005737  001372                          TST     MODE             ;IS THIS TEST IN STAGGERED MODE?
016114  100006                                  BPL     11$              ;IF NOT, SKIP STAGGERED SETUP

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

016116  006203                                  ASR     R3               ;GET THE LAST BIT INTO THE CARRY BIT
016120  103402                                  BCS     9$               ;IF IT IS SET, GO CLEAR IT
016122  000261                                  SEC                      ;IF IT IS CLEAR SET IT HERE
```

```
 2911  016124  000401                          BR      10$              ;SKIP THE CLEARING
 2912  016126  000241                  9$:     CLC                      ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
 2913  016130  006103                  10$:    ROL     R3               ;GET THE NEW BIT BACK INTO R3
 2914  016132  020337  001374          11$:    CMP     R3,SAVLIN        ;IS THIS A VALID LINE
 2915  016136  001401                          BEQ     .+4              ;YES
 2916  016140  104015                          ERROR   15               ;*INVALID LINE
 2917  016142  042704  177400                  BIC     #↑C<377>,R4      ;STRIP JUNK
 2918  016146  120504                          CMPB    R5,R4            ;DATA COMPARE ?
 2919  016150  001401                          BEQ     .+4              ;YES
 2920  016152  104005                          ERROR   5                ;*DATA DOES NOT COMPARE
 2921  016154  040277  163644                  BIC     R2,@DZVTCR       ;CLEAR TCR BIT
 2922  016160  000401                          BR      13$              ;GO GET OUT OF INTERRUPT MODE
 2923                                          ;TRANSMITTER INTERRUPT SVC ROUTINE
 2924  016162  104011                  12$:    ERROR   11               ;THE RECEIVER INTERRUPT FAILED
 2925                                                                   ;TO OVERRIDE THE TRANSMITTER
 2926  016164  022626                  13$:    POP2SP                   ;REMOVE THE INTERRUPT VECTOR FROM THE STACK
 2927  016166  005237  001374          14$:    INC     SAVLIN           ;ADJUST FOR NEXT LINE
 2928  016172  104420                          SHIFT                    ;GET THE NEXT POINTER. IF DONE, ADVANCE
 2929  016174  000137  015726                  JMP     3$               ;OTHERWISE GO DO THE NEXT LINE
```

```
                                                  ;ERROR TABLE
016200  000000                          .ERRTAB:           0          ;ERROR 0
016202  000000                                             0
016204  000000                                             0

016206  016346                                             EM1        ;ERROR
016210  017164                                             DH1
016212  017304                                             DT1

016214  016421                                             EM2        ;ERROR 2
016216  017210                                             DH2
016220  017316                                             DT2

016222  016447                                             EM3        ;ERROR 3
016224  017243                                             DH3
016226  017334                                             DT3

016230  016506                                             EM4        ;ERROR 4
016232  017243                                             DH3
016234  017334                                             DT3

016236  016535                                             EM5        ;ERROR 5
016240  017255                                             DH4
016242  017342                                             DT4

016244  016564                                             EM6        ;ERROR 6
016246  017255                                             DH4
016250  017342                                             DT4

016252  016623                                             EM7        ;ERROR 7
016254  017243                                             DH3
016256  017334                                             DT3

016260  016664                                             EM10       ;ERROR 10
016262  017243                                             DH3
016264  017334                                             DT3

016266  016726                                             EM11       ;ERROR 11
016270  017243                                             DH3
016272  017334                                             DT3

016274  016764                                             EM12       ;ERROR 12
016276  017243                                             DH3
016300  017334                                             DT3

016302  000000                                             0
016304  000000                                             0
016306  000000                                             0

016310  000000                                             0
016312  000000                                             0
016314  000000                                             0

016316  017023                                             EM15       ;ERROR 15
016320  000000                                             0
016322  000000                                             0
```

```
  2986
  2987  016324  000000                                          0
  2988  016326  000000                                          0
  2989  016330  000000                                          0
  2990
  2991  016332  017065                                          EM17     ;ERROR 17
  2992  016334  017243                                          DH3
  2993  016336  017334                                          DT3
  2994
  2995  016340  017123                                          EM20
  2996  016342  017243                                          DH3
  2997  016344  017334                                          DT3
```

```
 2998                                                         ;ERROR MESSAGES
 2999   016346  047200  020117  052502          EM1:    .ASCIZ  <200>/NO BUS REPLY RESPONSE FROM DZV11 REGISTER/
        016421     200  042522  044507          EM2:    .ASCIZ  <200>*REGISTER R/W FAILURE*
        016447     200  051124  047101          EM3:    .ASCIZ  <200>/TRANSMIT READY (TRDY) NOT SET/
        016506  051200  041505  044505          EM4:    .ASCIZ  <200>/RECEIVER DONE NOT SET/
        016535     200  040504  040524          EM5:    .ASCIZ  <200>/DATA COMPARISON ERROR/
        016564  042200  053132  030461          EM6:    .ASCIZ  <200>/DZV11 *RECEIVER BUFFER* ERROR/
        016623     200  051124  047101          EM7:    .ASCIZ  <200>/TRANSMITTER FAILED TO INTERRUPT/
        016664  052600  042516  050130          EM10:   .ASCIZ  <200>/UNEXPECTED TRANSMITTER INTERRUPT/
        016726  051200  041505  044505          EM11:   .ASCIZ  <200>/RECEIVER FAILED TO INTERRUPT/
        016764  052600  042516  050130          EM12:   .ASCIZ  <200>/UNEXPECTED RECEIVER INTERRUPT/
        017023     200  041501  044524          EM15:   .ASCIZ  <200>/ACTION DETECTED ON INVALID LINE./
        017065     200  040504  040524          EM17:   .ASCIZ  <200>/DATA VALID SHOULD NOT BE SET/
        017123     200  042522  042503          EM20:   .ASCIZ  <200>/RECEIVER DONE SHOULD NOT BE SET/

        017164  052200  040522  020120          DH1:    .ASCIZ  <200>/TRAP PC  DZV11 REG/
        017210  042600  050130  041505          DH2:    .ASCIZ  <200>/EXPECTED  FOUND  REGISTER/
        017243     200  044514  042516          DH3:    .ASCIZ  <200>/LINE NO./
        017255     200  054105  042520          DH4:    .ASCIZ  <200>/EXPECTED  FOUND  LINE/

                                                .EVEN
                                                        ;DATA TABLES FOR ERROR MESSAGES
 3000   017304  000002                          DT1:    2
 3001   017306     006     003                          .BYTE   6,3
 3002   017310  001330                                  $REG1
 3003   017312     006     001                          .BYTE   6,1
 3004   017314  001326                                  $REG0
 3005
 3006   017316  000003                          DT2:    3
 3007   017320     006     004                          .BYTE   6,4
 3008   017322  001340                                  $REG5
 3009   017324     006     001                          .BYTE   6,1
 3010   017326  001336                                  $REG4
 3011   017330     006     001                          .BYTE   6,1
 3012   017332  001326                                  $REG0
 3013
 3014   017334  000001                          DT3:    1
 3015   017336     003     001                          .BYTE   3,1
 3016   017340  001374                                  SAVLIN
 3017
 3018   017342  000003                          DT4:    3
 3019   017344     006     004                          .BYTE   6,4
 3020   017346  001340                                  $REG5
 3021   017350     006     001                          .BYTE   6,1
 3022   017352  001336                                  $REG4
 3023   017354     003     001                          .BYTE   3,1
 3024   017356  001374                                  SAVLIN
 3025
 3026                                                           ;TABLE OF DELAY TIMES FOR INDIVIDUAL BAUD RATES
 3027                                                           ;----------------------------------------------
 3028                                                           ;
 3029   017360  002450                          DLYTBL: 2450                    ;TIME FOR   50 BAUD
 3030   017362  001560                                  1560                    ;TIME FOR   75 BAUD
 3031   017364  001120                                  1120                    ;TIME FOR  110 BAUD
 3032   017366  000750                                   750                    ;TIME FOR  134 BAUD
 3033   017370  000660                                   660                    ;TIME FOR  150 BAUD
 3034  017372  000330                      330                 ;TIME FOR  300 BAUD
 3035  017374  000150                      150                 ;TIME FOR  600 BAUD
 3036  017376  000060                       60                 ;TIME FOR 1200 BAUD
 3037  017400  000040                       40                 ;TIME FOR 1800 BAUD
 3038  017402  000030                       30                 ;TIME FOR 2000 BAUD
 3039  017404  000020                       20                 ;TIME FOR 2400 BAUD
 3040  017406  000010                       10                 ;TIME FOR 3600 BAUD
 3041  017410  000001                        1                 ;TIME FOR 4800 BAUD
 3042  017412  000001                        1                 ;TIME FOR 7200 BAUD
 3043  017414  000001                        1                 ;TIME FOR 9600 BAUD
 3044  017416  000001                        1                 ;TIME OF DELAY FOR 19200 BAUD
 3045
 3046                                     ;DELAYS WERE COMPUTED TO ALLOW MAXIMUM TIME AT EACH BAUD RATE
 3047                                     ;FOR ALL TESTS TO FUNCTION CORRECTLY ON A LSI11.
 3048
 3049  017420                       CORMAX:
 3050          000001               .END
```

F07

```
MD-11-DVDZ(-A   MACY11 30(1046)  27-JUL-77  12:52  PAGE 65                                                    PAGE:  0083
DVDZAA.P11    27-JUL-77 12:51            CROSS REFERENCE TABLE -- USER SYMBOLS

ABASE = 16(010             1#      342       383
ACDW1 = 000017             1#      342       385
ACDW2 = 000000           342       386
ACPUOP= 000000           342       357
ACTIVE  001420           500#      756*     1789*     1790      1792*     1796      1797*     1798      1801*
ADDW0 = 017470             1#      342       387
ADDW1 = 017470             1#      342       388
ADDW10= 017470             1#      342       397
ADDW11= 017470             1#      342       398
ADDW12= 017470             1#      342       399
ADDW13= 017470             1#      342       400
ADDW14= 017470             1#      342       401
ADDW15= 017470             1#      342       402
ADDW2 = 017470             1#      342       389
ADDW3 = 017470             1#      342       390
ADDW4 = 017470             1#      342       391
ADDW5 = 017470             1#      342       392
ADDW6 = 017470             1#      342       393
ADDW7 = 017470             1#      342       394
ADDW8 = 017470             1#      342       395
ADDW9 = 017470             1#      342       396
ADEVCT= 000000           342       348
ADEVM = 000001             1#      342       364
ADRCNT  005661          1323*     1360*     1370#
ADVANC= 104400           652#     1500      2109      2381
AENV  = 000000           342       353
AENVM = 000000           342       354
AFATAL= 000000           342       345
AMADR1= 000000           342       370
AMADR2= 000000           342       374
AMADR3= 000000           342       377
AMADR4= 000000           342       380
AMAMS1= 000000           342       364
AMAMS2= 000000           342       372
AMAMS3= 000000           342       375
AMAMS4= 000000           342       378
AMSGAD= 000000           342       350
AMSGLG= 000000           342       351
AMSGTY= 000000           342       344
AMTYP1= 000000           342       365
AMTYP2= 000000           342       373
AMTYP3= 000000           342       376
AMTYP4= 000000           342       379
APASS = 000000           342       347
APRIOR= 000000           342
APTCSU= 000040          1174      1279#
APTENV= 000001          1167      1235      1277#     1587
APTSIZ= 000200          1276#
APTSPO= 000100          1169      1237      1278#
ASWREG= 000000           342       355
ATESTN= 000000           342       346
AUNIT = 000000           342       349
AUSWR = 000000           342       356
AUTO.S  011464           931      1972#
AVECT1= 000300             1#      342       381
AVECT2= 000000           342       382
```

```
BINWRD  006134          1446#
BIT0   = 000001          139#    211     213     250     261    2395
BIT00 = 000001           129#    139
BIT01 = 000002           128#    138
BIT02 = 000004           127#    137
BIT03 = 000010           126#    136
BIT04 = 000020           125#    135
BIT05 = 000040           124#    134
BIT06 = 000100           123#    133
BIT07 = 000200           122#    132
BIT08 = 000400           121#    131
BIT09 = 001000           120#    130
BIT1   = 000002          138#    212     213     251     262
BIT10 = 002000           119#    235     236     237     238     243     244     245     246     256     267     275    2327
BIT11 = 004000           118#    239     240     241     242     243     244     245     246     257     268     276    1107
                        2327
BIT12 = 010000           117#    178     194     229
BIT13 = 020000           116#    179     195
BIT14 = 040000           115#    180     196    1089    2051
BIT15 = 100000           114#    181     197
BIT2   = 000004          137#    252     263     992
BIT3   = 000010          136#    173     216     218     220     222     253     264
BIT4   = 000020          135#    174     217     218     221     222    1497    1512
BIT5   = 000040          134#    175     219     220     221     222     227    1984    1994    2051
BIT6   = 000100          133#    176     224
BIT7   = 000200          132#    165     177     225     929    1616    1630
BIT8   = 000400          131#    186     188     203     205     232     234     236     238     240     242     244     246
                         254     265     273    2257    2267
BIT9   = 001000          130#    187     188     204     205     233     234     237     238     241     242     245     246
                         255     266     274    2257    2267
BPTVEC= 000014           146#
BRK0   = 000400          273#
BRK1   = 001000          274#
BRK2   = 002000          275#
BRK3   = 004000          276#
BRW      004540          997    1127#
BUFSET= 104422           688#   2638
CHRCNT  006132          1412*   1425*   1443#
CLEAR = 000000           166#   2799*   2831*   2891*
CNVRT  = 104412          672#   1020    1022    1025    1028    1566    1568    1570    1623
CONVRT= 104411           670#    941    1584
CORMAX  017420          3049#   3050
C00    = 000400          265#
C01    = 001000          266#
C02    = 002000          267#
C03    = 004000          268#
CR     = 000015           54#   1213    1223
CRLF   = 000200           55#   1184    1223
CSRMAP  011472          1975#
CYCLE   010436           998    1051    1779#
DATABP  006626          1555*   1558    1582    1585#
DATAHD  006614          1554*   1578    1581#
DCLASM= 104417           682#   2473    2552    2635    2714    2761    2849
DCLR   = 000020          174#   1466    1467    2125    2287    2299    2306    2308
DDISP = 177570            61#    434
DELAY = 104414           676#   2270    2404    2412    2444    2489    2515    2565    2593    2608    2653    2678    2738
```

```
                            2782    2813    2872    2880
DEVADR  005656             1321*    1357    1368#
DEVICE= 104413              674#    1473    2167    2229    2255    2285    2324    2348    2383    2431    2453    2611    2834
DH1     017164             2936     2999#
DH2     017210             2940     2999#
DH3     017243             2944     2948    2960    2964    2968    2972    2992    2996    2999#
DH4     017255             2952     2956    2999#
DISPLA  001306              434#     765*   1119*   1597*
DISPRE  000174              329#     765
DLYCNT  006230              916*     974    1479    1484#
DLYTBL  017360              916     3029#
DONFLG  001425              508#    2476*   2493    2495*   2525*   2550*   2580    2603*   2639*   2657    2689*   2692*   2765*
                           2789     2796    2817    2823    2828    2830*
DSWR   = 177570              60#     433
DTR0   = 000400             254#    2221
DTR1   = 001000             255#    2222
DTR2   = 002000             256#    2223
DTR3   = 004000             257#    2224
DT1     017304             2937     3000#
DT2     017316             2941     3006#
DT3     017334             2945     2949    2961    2965    2969    2973    2993    2997    3014#
DT4     017342             2953     2957    3018#
DVALID= 100000              197#    2327    2511    2674    2733
DZCR0   001500              548#     796     799     844    2025
DZCR1   001512              554#     845
DZCR10  001620              596#
DZCR11  001632              602#
DZCR12  001644              608#
DZCR13  001656              614#
DZCR14  001670              620#
DZCR15  001702              626#
DZCR16  001714              632#
DZCR17  001726              638#
DZCR2   001524              560#
DZCR3   00153               566#
DZCR4   0015                572#
DZCR5   0015 2              578#
DZCR6   0015.               584#
DZCR7   00160               590#
DZVACT  001406              493#     944*    954     957     962    1779    1785
DZVCSR  002010              696#    1055    1466*   1467    1474*   1858*   2084    2124    2147    2254    2286    2434    2479*
                           2484     2487    2513    2533    2561*   2563    2589    2614    2641*   2647    2651    2676    2719*
                           2736     2775*   2777    2788*   2795*   2805*   2808    2827*   2856*   2865    2870    2878    2884
                           2889*    2890*
DZVC0   001502              549#     808     811    2067
DZVC1   001514              555#
DZVC10  001622              597#
DZVC11  001634              603#
DZVC12  001646              609#
DZVC13  001660              615#
DZVC14  001672              621#
DZVC15  001704              627#
DZVC16  001716              633#
DZVC17  001730              639#
DZVC2   001526              561#
DZVC3   001540              567#
```

```
DZVC4   001552        573#
DZVC5   001564        579#
DZVC6   001576        585#
DZVC7   001610        591#
DZVLEV  011002       1811      1847#
DZVLPR  002020        700#     1509*    1863*    2328*    2333#
DZVMSR  002030        704#     1872*    2102     2349     2384
DZVNUM  001414        496#      837*     945     1942*    1952*    1980*    2009*    2010     2016     2018
DZVRBU  002014        698#     1862*    2090     2325     2519     2596     2618     2682     2742     2862     2899
DZVRIS  002042        712#     1849*    2768*    2773*    2803*    2853*
DZVRIV  002040        711#     1058     1803*    1847     2772*    2802*    2852*
DZVTCR  002024        702#     1868*    2096     2195     2263*    2439*    2482*    2526*    2560*    2578*    2587*    2644*    2691*
                     2766*     2861*    2921*
DZVTDR  002034        706#     1873*    2352*    2357*    2496*    2535*    2584*    2659*    2806*    2876*
DZVTIS  002046        714#     1853*    2769*    2774*    2804*    2855*
DZVTIV  002044        713#     1851*    2771*    2801*    2854*
DZV.EN  001740        644#      782     1790     1798     1978
DZV.MA  001500        500       546#     756      779      936     1792     1801     1935     1975     1981     2039
EIGHT = 000030        218#
EIGHTS= 000070        222#
EMTVEC= 000030        149#
EM1     016346       2935      2999#
EM10    016664       2963      2999#
EM11    016726       2967      2999#
EM12    016764       2971      2999#
EM15    017023       2983      2999#
EM17    017065       2991      2999#
EM2     016421       2939      2999#
EM20    017123       2995      2999#
EM3     016447       2943      2999#
EM4     016506       2947      2999#
EM5     016535       2951      2999#
EM6     016564       2955      2999#
EM7     016623       2959      2999#
ERRMSG  006602       1553*     1573     1576#
ERRVEC= 000004        142#     1094     1095*    1097*    1100*
EPTAB0  006756       1568      1609#
EVEPAR= 000000        228#
EXITER  006706       1596      1599#
FIVE  = 000000        215#
FIVES = 000040        219#
FRMERR= 020000        195#     2733
HALTS   006632       1538      1587#
HDRFLG  001423        506#      780*     879*     932      934*
HDZVCS  002012        697#     1860*    2300*    2569
HDZVLP  002022        701#     1866*
HDZVMS  002032        705#     1875*
HDZVRB  002016        699#     1865*
HDZVTC  002026        703#     1870*    2403*    2411*
HDZVTD  002036        707#     1876*    2735*    2746*
HILIM   005654       1320*     1348     1367#
HT    = 000011         52#     1182     1223
INBUF   010270       1290      1326     1719*    1720     1725     1730     1748*    1749     1753     1763#    1887
INIFLG  001422        505#      766      771*     784*     865*     927     1974*
INSTER= 104404        660#     1342     1736     1758     1906
INSTR = 104403        658#      791      803      814      821      871      904      949      969     1817
```

```
INSTR2   005454              1297     1309#
IOTVEC=  000020               147#    2068*
LF    =  000012                53#    1217     1223
LIMITS   005602              1336     1348#
LINE     001366               485#    1804*    2387     2437     2480     2559     2642     2712     2766     2858
LINE0    001504               550#     830*     876      886      912
LINE1    001516               556#
LINE10   001624               598#
LINE11   001636               604#
LINE12   001650               610#
LINE13   001662               616#
LINE14   001674               622#
LINE15   001706               628#
LINE16   001720               634#
LINE17   001732               640#
LINE2    001530               562#
LINE3    001542               568#
LINE4    001554               574#
LINE5    001566               580#
LINE6    001600               586#
LINE7    001612               592#
LOBITS   005660              1322*    1352     1369#
LOCK     001364               480#    1135     1137     1490*    1562     2083*    2089*    2095*    2101*    2150*    2157*    2165*    2184*
                             2197*    2205*    2226*    2382*    2410*    2418*    2432*    2472*    2634*    2709*
LOLIM    005652              1319*    1350     1366#
LPRSET=  104421               686#    2474     2556     2636     2717     2763     2850
LP0   =  000000               210#
LP1   =  000001               211#
LP2   =  000002               212#
LP3   =  000003               213#
MAINT =  000010               173#    1125     1728     1810     2178     2288     2301
MANT0    001510               552#     817      884     2026
MANT1    001522               558#
MANT10   001630               600#
MANT11   001642               606#
MANT12   001654               612#
MANT13   001666               618#
MANT14   001700               624#
MANT15   001712               630#
MANT16   001724               636#
MANT17   001736               642#
MANT2    001534               564#
MANT3    001546               570#
MANT4    001560               576#
MANT5    001572               582#
MANT6    001604               588#
MANT7    001616               594#
MASK  =  000200               165#     310      752*     989*    2037     2082     2767*    2768     2769     2773     2774     2803     2804
                             2833*    2853     2855     2860*
MASTEK   007705              1564     1707#
MBADLN   010020               893     1707#
MCSRX    007635              1019     1569     1707#
MDATA    010374              1423     1433     1767#
MEPASS   007453              1018     1707#
MERRPC   007765              1567     1707#
MERRX    007662              1027     1707#
```

```
MERR2   007513              1707#   1781    2030
MERR3   007562               959    1707#
MLOCK   007606               994    1707#
MNEW    007710               950    1707#
MNTFLG  001424               507#   1122*   1125*   1474    1723*   1728*   1733*   1807*   1810*
MODE    001372               487#   1123    1806*   1808    2379    2393    2498    2661    2721    2903
MPASSX  007651              1024    1707#
MPFAIL  007410              1700    1707#
MR      007477              1000    1707#
MSENAB= 000040               175#   2179    2262    2264    2288    2301    2435    2440    2479    2561    2641    2719    2775
                            2805    2856
MTITLE  001000               336#    770
MTSTN   007673              1565    1707#   1818
MVECX   007643              1021    1707#
NEXT    001362               479#   1489    1605    2080*   2123*   2146*   2194*   2253*   2284*   2323*   2347*   2378*   2430*
                            2471*   2549*   2633*   2708*   2760*   2848*
NOLIST= ****** U               1
ODDPAR= 000200               225#   2716
ONESTO= 000000               226#
OVRRUN= 040000               196#
PAR     001370               486#   1507    1805*   2554    2555*   2557*   2604    2715    2716*   2718*
PARAM = 104405               662#    793     805     823     873     906     951     971    1819
PARAM1  005522              1325#   1343
PARER = 010000               194#   2733
PARERR  005576              1328    1330    1332    1342#   1349    1351    1353
PARITY= 000100               224#   2716
PARMD = 104415               678#
PAR0    001506               551#    831*    909     913
PAR1    001520               557#
PAR10   001626               599#
PAR11   001640               605#
PAR12   001652               611#
PAR13   001664               617#
PAR14   001676               623#
PAR15   001710               629#
PAR16   001722               635#
PAR17   001734               641#    923
PAR2    001532               563#
PAR3    001544               569#
PAR4    001556               575#
PAR5    001570               581#
PAR6    001602               587#
PAR7    001614               593#
PAWCH = 104416               680#
PIRQ  = 177772                59#
PIRQVE= 000240               153#
POPR0 = 012600               162#
POP¡SP= 005726               160#
POP¿SP= 022626               164#   1499    2064    2111    2787    2794    2826    2926
PR0   = 000000                76#
PR1   = 000040                77#
PR2   = 000100                78#
PR3   = 000140                79#
PR4   = 000200                80#
PR5   = 000240                81#
PR6   = 000300                82#
```

```
PR7    = 000340        83#
PS     = 177776        56#     57
PSW    = 177776        57#
PUSHR0= 010046        161#
PUSH1S= 005746        159#
PUSH2S= 024646        163#
PWRVEC= 000024        148#   1669*   1670*   1679*   1685*   1697*   1698*
RCVON  = 010000       229#   2555
RDONE  = 000200       177#   2257
REGIST   001402       491#
RESREG   006630      1583    1586#
RESTAR   010776      1702    1843#
RESVEC= 000010        143#
RESOS  = 104410       668#   1586
RIE    = 000100       176#   2181    2288    2301    2805    2827    2890
RING0  = 000001       261#
RING1  = 000002       262#
RING2  = 000004       263#
RING3  = 000010       264#
RL0    = 000000       202#
RL1    = 000400       203#
RL2    = 001000       204#
RL3    = 001400       205#
RUN      001412       495#    757*   1785    1787*   1788*   1794*   1795*
SAVACT   001410       494#    836*    841*    843     944     957    1943*   1951*   2019*   2023*   2028    2047
SAVLIN   001374       488#   2433*   2456*   2475*   2497    2527*   2551*   2600*   2616*   2620*   2637*   2645    2660
                     2693*   2710*   2720    2748*   2764*   2851*   2914    2927*   3016    3024
SAVNO    001416       498#    945*    961*    965*    999    1032
SAVNUM   001415       497#    999*   1030*   1032*
SAVPC    001404       492#   1375*   1611
SAVOS  = 104407       666#   1543
SCOP1  = 104401       654#   2113    2156    2164    2173    2204    2212    2409    2417    2452    2524    2536    2686
                     2747
SERV.G   006772      1121    1531    1600    1615#   1620
SETAPT   011336       774    1935#
SETFLG= 104406        664#    816
SEVEN  = 000020       217#
SEVENS= 000060        221#
SHIFT  = 104420       684#   2390    2455    2528    2694    2749    2928
SILOAL= 020000        179#   2257
SILOEN= 010000        178#   2180    2288
SIX    = 000010       216#
SIXS   = 000050       220#
SPACNT   C06133      1444#
STACK  = 001120        47#    751     988    1606
STKLMT= 177774         58#
STOP     001446       519#   1523
SVOS     005670      1379#
SWR      001304       433#    764*    776     833     861     929     946     967     992    1089    1107    1133    1140
                     1532    1537    1595    1601    1603    1621    1626*   1638*   1639*   1640*   1647*   1651*   1677
                     1690*   1814    1960*
SWREG    000176       330#    764
SW0    = 000001       111#
SW00   = 000001       :01#    111     776
SW01   = 000002       100#    110    1814
SW02   = 000004        99#    109
```

```
SW03   = 000010          98#    108      833      861
SW04   = 000020          97#    107      967
SW05   = 000040          96#    106
SW06   = 000100          95#    105      946
SW07   = 000200          94#    104
SW08   = 000400          93#    103     1601
SW09   = 001000          92#    102     1133
SW1    = 000002         110#
SW10   = 002000          91#   1603
SW11   = 004000          90#
SW12   = 010000          89#   1140     1532
SW13   = 020000          88#   1537
SW14   = 040000          87#
SW15   = 100000          86#
SW2    = 000004         109#
SW3    = 000010         108#
SW4    = 000020         107#
SW5    = 000040         106#
SW6    = 000100         105#
SW7    = 000200         104#
SW8    = 000400         103#
SW9    = 001000         102#
S110   = 001000         233#
S1200  = 003400         238#
S134   = 001400         234#
S150   = 002000         235#
S1800  = 004000         239#
S19200= 007400          246#
S2000  = 004400         240#
S2400  = 005000         241#
S300   = 002400         236#
S3600  = 005400         242#
S4800  = 006000         243#
S50    = 000000         231#
S600   = 003000         237#
S7200  = 006400         244#
S75    = 000400         232#
S9600  = 007000         245#
TBITVE= 000014          144#
TCR0   = 000001         250#   2217
TCR1   = 000002         251#   2218
TCR2   = 000004         252#   2219
TCR3   = 000010         253#   2220
TD0      001426         511#   1521     2659     2673     2687*    2806
TD1      001430         512#
TD2      001432         513#
TD3      001434         514#
TEIGHT   002106         737#
TEMP     010332        1765#
TFIVE    002114         740#
TIE    = 040000         180#   2182     2288     2775     2788     2795     2827     2889
TKVEC  = 000060         151#
TLAST  = 015636        1837    3026#
TL0    = 000000         185#
TL1    = 000400         186#
TL2    = 001000         187#
```

```
TL3    = 001400            188#
TMTBL    002050            721#
TPVEC  = 000064            152#
TRAPVE=  000034            150#
TRDY   = 100000            181#    2257    2262    2435
TRTVEC=  000014            145#
TR0      001436            515#
TR1      001440            516#
TR2      001442            517#
TR3      001444            518#
TSEVEN   002110            738#
TSIX     002112            739#
TST1     012170           1083     1325    1841    2078#
TST10    013324           2323     2345#
TST11    013410           2347     2376#
TST12    013604           2378     2428#
TST13    013736           2430     2469#
TST14    014226           2471     2547#
TST15    014550           2549     2631#
TST16    015040           2633     2706#
TST17    015242           2708     2758#
TST2     012360           2080     2121#
TST20    015636           2760     2846#   3026
TST21  = ****** U         2848
TST3     012424           2123     2144#
TST4     012602           2146     2192#
TST5     013006           2194     2251#
TST6     013110           2253     2282#
TST7     013240           2284     2321#
TTST     004312            995*     997*   1085#
TWOSTO=  000040            227#
TYPDAT   006616           1559     1579    1582#
TYPE   = 104402            656#     770     892     893     935     959     994    1000    1018    1019    1021    1024    1027
                          1187     1288    1307    1400    1433    1560    1561    1564    1565    1567    1569    1571    1575
                          1580     1622    1624    1652    1699    1781    1816    1834    1839    2030
TYPMSG   006506           1557     1560#
T110     002054            724#
T1200    002066            729#
T134     002056            725#
T150     002060            726#
T1800    002070            730#
T2000    002072            731#
T2400    002074            732#
T300     002062            727#
T3600    002076            733#
T4800    002100            734#
T50      002050            722#
T600     002064            728#
T7200    002102            735#
T75      002052            723#
T9600    002104            736#
VECMAP   012000           2029     2037#
WRDCNT   006130           1408*    1434*   1442#
WTBS.F   006604           1574     1577#
XBX      006374           1533     1535    1537#
XCSR     004250           1020     1053#   1570
```

```
XERR    004272           1028    1062#
XHEAD   007772            935    1707#
XMTCNT  001400            490#
XMTLIN  001376            489#
XPASS   004264           1025    1059#
XSTATQ  010062            942    1707#
XTSTN   006764           1566    1612#
XVEC    004256           1022    1056#
XX    = 160210            547#    553#    559#    565#    571#    577#    583#    589#    595#    601#    607#    613#    619#
                          625#    631#    637#    643#
YY    = 000500            547#    553#    559#    565#    571#    577#    583#    589#    595#    601#    607#    613#    619#
                          625#    631#    637#    643#
ZZ    = 000020            547#    553#    559#    565#    571#    577#    583#    589#    595#    601#    607#    613#    619#
                          625#    631#    637#    643#
$APTHD  001446            529     535#
$ASTAT= ****** U         1257    1272
$ATYC   005114           1228    1230#
$ATY1   005070           1226#
$ATY3   005076           1172    1227#
$ATY4   005106           1229#   1590
$AUTOB  001300            430#
$BASE   001174            383#    799*   1802*   1857    1936    2025*
$BDADR  001266            425#
$BDDAT  001272            427#
$CDW1   001200            385#   1955
$CDW2   001202            386#   1957    2026*
$CHARC  005064           1189*   1199*   1206    1215*   1220#
$CMTAG  001244            413#
$CM1  = 000006            445#    446#    447#    448#    449#    450#    451#
$CM2  = 000014            445#    446#    447#    448#    449#    450#    451#
$CM3  = 000006            443#    445
$CM4  = 000005            451#    452#    453#    454#    455#    456#
$CPUOP  001146            357#
$CRLF   001357            458#   1188    1223    1400    1560    1561    1571    1652    1816    1834
$DOW0   001204            387#    846    1940    1976
$DOW1   001206            388#
$DOW10  001230            397#
$DOW11  001232            398#
$DOW12  001234            399#
$DOW13  001236            400#
$DOW14  001240            401#
$DOW15  001242            402#
$DOW2   001210            389#
$DOW3   001212            390#
$DOW4   001214            391#
$DOW5   001216            392#
$DOW6   001220            393#
$DOW7   001222            394#
$DOW8   001224            395#
$DOW9   001226            396#
$DEVCT  001130            348#   1029*
$DEVM   001176            384#   1941    2028*
$DOAGN  004244           1031    1038    1043    1049#
$E    = 000022              1#   2080    2081#   2123    2124#   2146    2147#   2194    2195#   2253    2254#   2284    2285#
                         2323    2324#   2347    2348#   2378    2379#   2430    2431#   2471    2473#   2549    2550#   2633
                         2635#   2708    2709#   2760    2761#   2848    2849#
```

```
MD-11-DVDZA-A   MACY11 30(1046)  27-JUL-77  12:52  PAGE 75
DVDZAA.P11    27-JUL-77 12:51          CROSS REFERENCE TABLE -- USER SYMBOLS

$ENDAD  004234            323     768    1045#   1593
$ENOCT  004220           1040#
$ENV    001140            353#   1167    1235    1259    1587
$ENVM   001141            354#    772    1169    1174    1237
$EOP    004064           1014#   2848
$EOPCT  004212           1037#   1041
$ERFLG  001247            416#    755*   1017*   1074    1103    1105*   1129    1542*   1556    1572*
$ERMAX  001261            422#   1129
$ERROR  006344            313    1531#
$ERRPC  001262            423#    759*   1016*   1082*   1535    1541*
$ERRTB  001362            474#
$ERTTL  001256            420#    758*   1064    1599*
$ETABL  001140            352#
$ETEND  001244            405#    541
$FATAL  001122            345#   1263*
$FFLG   005334           1226*   1229*   1257    1266*   1274#
$FILLC  001322            441#   1192    1223
$FILLS  001321            440#   1223
$FLIP = 177777              1#   2072#   2076#   2115#   2119#   2134#   2142#   2185#   2190#   2243#   2249#   2275#   2280#
                         2315#   2319#   2339#   2343#   2364#   2373#   2421#   2426#   2458#   2467#   2539#   2545#   2624#
                         2629^   2698#   2704#   2751#   2756#   2836#   2844#
$GDADR  001264            424#
$GDDAT  001270            426#
$GET42  004224           1042#
$HD   = 000001             10      11
$HIBTS  001446            536#
$ICNT   001250            417#   1111*   1112    1114*   1128
$ILLUP  007402           1669    1685    1704#
$INTAG  001301            431#
$ITEMB  001260            421#   1547*   1589
$LF     001360            459#   1223
$LFLG   005333           1267*   1273#
$LPADR  001252            418#    761*    998*   1001    1118*   1120    1128    1605*   1607    1832*   1833*   1841*   1843
$LPERR  001254            419#
$MADR1  001152            370#
$MADR2  001156            374#
$MADR3  001162            377#
$MADR4  001166            380#
$MAIL   001120            343#    537     541    1117    1167
$MAMS1  001150            364#
$MAMS2  001154            372#
$MAMS3  001160            375#
$MAMS4  001164            378#
$MBADR  001450            537#
$MFLG   005332           1227*   1233    1268*   1272#
$MSGAD  001134            350#   1243*   1246
$MSGLG  001136            351#   1248*
$MSGTY  001120            344#   1241    1249*   1261    1265*
$MTYP1  001151            365#
$MTYP2  001155            373#
$MTYP3  00116!            376#
$MTYP4  001165            379#
$MXCNT  004542           1115    1128#   1784
$N    = 000020              1#   2072    2076    2081#   2115    2119    2124#   2134    2142    2147#   2185    2190    2195#
                         2243    2249    2254#   2275    2280    2285#   2315    2319    2324#   2339    2343    2348#   2364
                         2374    2379#   2421    2426    2431#   2458    2467    2473#   2539    2545    2550#   2624    2629
```

```
                           2635#   2698    2704    2709#   2751    2756    2761#   2836    2844    2849#   3026#
$NULL    001320             439#   1194    1223
$NWTST=  000000            2077#   2120#   2143#   2191#   2250#   2281#   2320#   2344#   2375#   2427#   2468#   2546#   2630#
                           2705#   2757#   2845#
$OVER    004500            1087    1090    1101    1113    1119#
$PASS    001126             347#    754*   1023*   1026*   1034*   1035*   1052    1061    1109    1129
$PASTM   001454             539#
$PWRAD   007376            1702#
$PWRDN   007236             311     753    1669#   1697
$PWRMG   007372            1700#
$PWRUP   007310            1679    1685#
$QUES    001356             457#    892    1223    1307    1839
$REGAD   001324             443#
$REG0    001326             445#   1384*   1389    3004    3012
$REG1    001330             446#   1383*   1390    3002
$REG2    001332             447#   1382*   1391
$REG3    001334             448#   1381*   1392
$REG4    001336             449#   1380*   1393    3010    3022
$REG5    001340             450#   1379*   1394    3008    3020
$RTNAD   004246            1051#
$SAVR6   007406            1678*   1686    1687*   1688*   1706#
$SCOPE   004300            1081#
$SETUP=  000000            1033    1082
$SVLAD   004462            1098    1116#
$SVPC =  000040             321#    326
$SWR  =  164000               1#     10     456     457    1011    1033    1044    1050    1052    1075    1076    1077    1078
                           1089    1101    1103    1104    1105    1106    1107    1119    1128    1703    2079    2122    2145
                           2193    2252    2283    2322    2346    2377    2429    2470    2548    2632    2707    2759    2847
$SWREG   001142             355#   1960
$SWRMK=  000000            1078
$TESTN   001124             346#   1117*
$TIMES   001354             456#   1033*   1106*   1112    1115*   1128    1784*
$TKB     001312             436#   1088    1294    1300    1615    1629
$TKS     001310             435#   1086    1292    1619    1627
$TMP0    001342             451#
$TMP1    001344             452#    826     837     838*    937*    962*    963*   1709
$TMP2    001346             453#    843*    848*    849*    938*    939    1711
$TMP3    001350             454#
$TMP4    001352             455#
$TN   =  000021              10#   2077    2079#   2120    2122#   2143    2145#   2191    2193#   2250    2252#   2281    2283#
                           2320    2322#   2344    2346#   2375    2377#   2427    2429#   2468    2470#   2546    2548#   2630
                           2632#   2705    2707#   2757    2759#   2845    2847#
$TPB     001316             438#   1212*   1223    1300*   1536*   1636*
$TPFLG   001323             442#   1161    1223
$TPS     001314             437#   12'0    1223    1298    1534    1633
$TSTM    001452             538#
$TSTNM   001246             415#    760*   1074    1116*   1117    1119    1129    1614    1822    1830    2079*   2122*   2145*
                           2193*   2252*   2283*   2322*   2346*   2377*   2429*   2470*   2548*   2632*   2707*   2759*   2847*
$TYPE    004606            1141    1161#   1254
$TYPEC   005020            1191    1198    1205    1210#   1211
$TYPEX   005066            1216    1218    1221#
$UNIT    001132             349#
$UNITM   001456             540#
$USWR    001144             356#
$VECT1   001170             381#    811*   1937    2067*
$VECT2   001172             382#
```

```
$XTSTR  004340              1084    1092#
$Y    = 000023               645#    652     654#    656#    658#    660#    662#    664#    666#    668#    670#    672#    674#
                             676#    678#    680#    682#    684#    686#    688#    690#
$$GET4= 000000              1044#
$4OCAT= ****** U            1089
.     = 017420               304#    305     308#    321     322#    324#    326#    328#    331#    335#    337#    460     493#
                             494#    496#    498#    499#    525     526#    528#    530#    545#    548#    549#    550#    551#
                             552#    554#    555#    556#    557#    558#    560#    561#    562#    563#    564#    566#    567#
                             568#    569#    570#    572#    573#    574#    575#    576#    578#    579#    580#    581#    582#
                             584#    585#    586#    587#    588#    590#    591#    592#    593#    594#    596#    597#    598#
                             599#    600#    602#    603#    604#    605#    606#    608#    609#    610#    611#    612#    614#
                             615#    616#    617#    618#    620#    621#    622#    623#    624#    626#    627#    628#    629#
                             630#    632#    633#    634#    635#    636#    638#    639#    640#    641#    642#    926#   1052
                            1128    1129    1223    1275#   1664#   1684    1705    1764#   1766#   1768#   1783    2033    2056
                            2863    2866    2885    2915    2919
.ADVAN  006232               653    1489#
.BEGIN  003770               988#
.BUFSE  006322               689    1520#
.CNVRT  005760               673    1401#
.CONVR  005754               671    1400#
.DCLAS  006200               683    1473#
.DELAY  006212               677    1477#
.DEVIC  006160               675    1465#
.ERRTA  016200              1552    2931#
.INSTE  005442               661    1305#
.INSTR  005336               659    1284#
.INST1  005356              1288#   1308
.LPRSE  006262               687    1505#
.MSG    005360              1286*   1289#
.PARAM  005462               663    1316#
.PARMD  011142               679    1879#
.PAWCH  010214               681    1747#   1759
.RESOS  005722               669    1389#
.SAVOS  005662               667    1375#
.SCOPE  004300               309    1082#   2068
.SCOP1  004544               655    1133#
.SETFL  010074               665    1718#   1737
.SHIFT  006244               685    1496#
.START  002116               332     749#    761
.TRPSR  006136               315    1454#
.TRPTA  001742               651#   1459
.TYPE   004570               657    1140#
.$ASTA= ****** U            1227    1230
.$X   = 001446               525#    530
```

```
                                                                    F08
MD-11-DVDZA-A   MACY11 30(1046)  27-JUL-77  12:52  PAGE 79                                                        PAGE:  0096
DVDZAA.P11    27-JUL-77 12:51            CROSS REFERENCE TABLE -- MACRO NAMES

COMMEN      154#
ENDCOM      154#
ERROR        48#   2112    2132    2155    2162    2171    2203    2210    2233    2241    2261    2273    2292    2298    2305
           2313    2332    2337    2356    2361    2408    2416    2447    2451    2486    2492    2518    2522    2568    2591
           2601    2617    2621    2649    2656    2681    2685    2741    2745    2785    2791    2793    2798    2816    2819
           2821    2325    2864    2867    2875    2883    2886    2894    2895    2916    2920    2924
ESCAPE      154#
GETPRI      154#
GETSWR      154#
MULT        154#
NEWTST      154#   2077    2120    2143    2191    2250    2281    2320    2344    2375    2427    2468    2546    2630    2705
           2757    2845
PASEND        1#   1015
POP         154#   1269    1270    1690    1691
PRGEND        1#   1002
PRGFRT        1#
PUSH        154#   1230    1232    1253    1671    1677
REPORT        1#    154#
SC            1#   1082
SCOPE        49#   1015    2078    2121    2144    2192    2251    2282    2321    2345    2376    2428    2469    2547    2631
           2706    2758    2846
SC1           1#   1121
SETPRI      154#
SETUP       154#
SKIP        154#
SLASH       154#
SPACE       154#
STARS       154#    319     338     341     409     522     524     531    1009    1071    1146    1225    1667    1683    2077
           2120    2143    2191    2250    2281    2320    2344    2375    2427    2468    2546    2630    2705    2757    2845
SWRSU       154#
TYPBIN      154#
TYPDEC      154#
TYPNAM      154#
TYPNUM      154#
TYPOCS      154#
TYPOCT      154#
TYPTXT      154#
$BUFFE        1#   1760
$CYCLE        1#   1770
$EOP          1#   1002
$GETFL        1#    814
$GETPA        1#    791     803     821     870     903     949     969    1817
$HEADE        1#     11
$INTSE        1#   2770    2800
$JUNK         1#    547     553     559     565     571     577     583     589     595     601     607     613     619     625
            631     637
$LINEU        1#
$LVLTS        1#   2751
$MRESE        1#
$MRR          1#   2243
$MRRW         1#   2134
$MRWD         1#   2315    2339
$MSG          1#   1707
$SCOPE        1#   1065
$SETFL        1#   1713
$STAG         1#   2498    2661    2721    2903
```

```
$STAGF       1#
$TCR         1#   2185
$TLINE       1#   2421
$TRPDE       1#    652     654     656     658     660     662     664     666     668     670     672     674     676     678
           680     682     684     686     688
$TSTN        1#   2076    2119    2142    2190    2249    2280    2319    2343    2374    2426    2467    2545    2629    2704
          2756    2844
$UNIBU       1#   2072
$VARIA       1#    334
$XZ          1#   2072    2076    2115    2119    2134    2142    2185    2190    2243    2249    2275    2280    2315    2319
          2339    2343    2364    2373    2421    2426    2458    2467    2539    2545    2624    2629    2698    2704    2751
          2756    2836    2844
$$CMRE     337#    445     446     447     448     449     450
$$CMT''    337#    451     452     453     454     455
$$ESCA     154#
$$NEWT     154#   2077    2120    2143    2191    2250    2281    2320    2344    2375    2427    2468    2546    2630    2705
          2757    2845
$$SKIP     154#
.EQUAT       1#     44
.HEADE       1#
.SETUP       1#
.$ACT1       1#    317
.$APTB       1#    339#
.$APTH       1#    520
.$APTY       1#   1223
.$CATC       1#
.$CMTA     337#
.$EGP        1#   1007
.$ERRO       1#
.$POWE       1#   1665
.$SCOP       1#   1069
.$TRAP       1#
.$TYPE       1#   1144


. ABS.  017420     000


 ERRORS DETECTED:  0

 DVDZAA,DVDZAA.SEQ=DVDZAA.P11
 RUN-TIME: 22 13 1 SECONDS
 RUN-TIME RATIO: 218/36=5.9
 CORE USED:  36K  (71 PAGES)
```