DZV-11

DZV11 CABLE + ECHO TESTS

MD-11-DVDZC-A

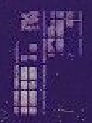
EP-DZVDZC-A-DL-A

FICHE 1 OF 1

OCT 1977

digital

MADE IN USA



```
EOF1DVDZBASEQ          00010000     770920          PDP10 411          EWHDR1DVDZCASEQ          00010000     770920
     PDP10 PAGE:  0001
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DVDZC-A-D

PRODUCT NAME: DZV11 CABLE AND ECHO TESTS

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.

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.

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)
ASR 33 (or equivalent) to run DZV11 ECHO TEST
DZV11 INTERFACE MODULE
H325 Cable turnaround connector.

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 the operator if the parameters have been already built by running either the DVDZA or DVDZB diagnostics. Loading this diagnostic will preserve these locations.

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 the SWR to allow the desired program options to function.
NOTE: Loc. 000176 is used as a software Switch Register in all of the DZV11 diagnostics. (see Sec. 4.1)

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).

C. The program will then ask for the Device Address, the Vector and the Line no. of the DZV11 to be tested. Type these values on the console terminal followed by a <CR>. The program will then ask for which test is desired, Echo or Cable. Type either E or C and a <CR>. The diagnostic will type out the name of the test that is now running (see Sec. 5.1).

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:	Reserved
SW 13	Set:	Inhibit error print out
SW 12	Set:	Inhibit **ALL** type out/bell on error.
SW 11	Set:	Reserved
SW 10	Set:	Go to End of Pass after an error
SW 09	Set:	Loop with current data (see Sec. 4.1.1)
SW 08	Set:	Restart test after an error
SW 07	Set:	Reserved
SW 06	Set:	Reserved
SW 05	Set:	Reserved
SW 04	Set:	Reserved
SW 03	Set:	Reserved
SW 02	Set:	Reserved
SW 01	Set:	Reserved
SW 00	Set:	Reserved

4.1.1 SWITCH REGISTER RESTRICTIONS

SW 09 LOOP ON CURRENT DATA: this switch is only used in the Cable test to lock on testing if setting the DTR bit for the desired line in the Transmit Control Register of the DZV11 will cause the CO and RING bits to set for that line in the Modem Status Register. 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.

4.1.2 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 Restart the test after an error
5. SW 10 Go to the End of Pass after an error

SCOPE SWITCHES

1. SW 09 (if enabled by 'SCOP1'). If an '*' is printed in front of the test no. on an error report then SW09 is incorporated in that test. This switch provides the operator with the ability to lock on a specific test operation.
   If the program user is technically trained to electronically isolate signal problems on the DZV11 module, this switch might prove to be a useful aid.
   Presently this switch is only used in this diagnostic for the Cable test to lock on checking that if DTR is set for an active line the CO and RING will become set for that line.

4.2 STARTING ADDRESS

SA 200 - The starting address for any DZV11 diagnostic is Loc. 200

NOTE: This diagnostic is not designed to run in an automatic chain mode because of the operator intervention required to run it.

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 HOW TO RUN THE "CABLE/ECHO" TESTS.

Normal starting proceedure for the first time would be:
Load the diagnostic, set the SWR at loc. 176 to whatever settings are desired, then start the program at loc. 200.
The program will print out on the console terminal:

"VECTOR ADDRESS-"

You type a vector followed by a <CR>.

"CONTROL REGISTER ADDRESS-"

You type in the DZVCSR address under test followed by a <CR>.

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

Lets do the CABLE TEST first. Type "C" and a <CR>.

"BAUD RATE- "

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

"LINE: "

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

"CABLE TEST"

and if everything is working, the End of Pass message will be printed after each pass.

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

"LINE: "

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

"CABLE TEST"

and begin running the diagnostic.
Continue this operation until all lines are tested.

5.2 ECHO TEST

Start the program at loc. 200 and enter the values for the CSR address and the device vector. The program will then print out on the console:

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

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

"BAUD RATE-"

Type the BAUD RATE. Baud rate choices are: 50,75, 110, 135, 150 300, 600 , 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600. The program will then print:

LINE: "

Type the line number which the terminal is connected to. Then the program will print:

"TERMINAL ECHO TEST"

*** AT THIS POINT THE MESSAGE:

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

Should be printed on the terminal connected to the DZV11. If this message is desired to be printed continuously, type a Control G <↑G> on the CONSOLE terminal while the message is printing. The program will print a prompt on the console asking for a new SWR setting. By setting the SWR to 377 the QUICK BROWN FOX message will be continuously printed on the DZV terminal. A Control G can then be typed on the console terminal at any time to reset the SWR and return to the flow of the diagnostic. The program will then print on the console terminal:

"TYPE A CHAR. ON DZV11 TERMINAL"

Any printable character which is typed on the DZV11 terminal will be echoed back on the terminal.
If you type Control C <↑C> on the DZV11 terminal the program will print the End of Pass message on the console terminal and the "QUICK BROWN FOX" message will begin printing on the DZV11 terminal again, the echo test will be resumed.

TO CHANGE LINES:

Type any printable character on the CONSOLE TERMINAL (not the DZV11 terminal). The program will again type "LINE: " and wait for a response.

5.3 PROGRAM AND/OR OPERATOR ACTION

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

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

6. ERRORS

As described previously there will always be a TEST NUMBER and PC typed out at the time of an error (providing SW 13=0 and SW 12=0). in most cases additional information will be supplied to the the error message which is to give the operator an indication of the error.

6.1 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 the operator to regain control of the CPU. It will then be necessary to check the PC processor register and refer to this location in the program listing to find out what the program was doing at the time of the error.

7. OPERATING RESTRICTIONS

When running the Cable test, the line that is declared active must be terminated by an H325 test connector which will turn the transmitted signal around to the receiver on the same line.
The diagnostic is not designed to determine a logic problem with the DZV interface. It is designed only to verify that the interface cable is providing a true link to the terminals which are connected to the DZV11.

8. MISCELLANEOUS

8.1 EXECUTION TIME

The execution time for the Cable test depends upon the desired baud rate given at start up time. At 9600. baud the End Pass message will print out before 10 seconds have elapsed.
The execution time for the Echo test is entirely dependent upon the number of characters the operator wishes to send.

8.2 PASS COMPLETE

When the diagnostic has completed a pass the following is an example of the print out to be expected.

```
END PASS DVDZC-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

After the base device address and the base vector have been typed in, locations 2010 through 2046 will contain the various device register addresses and the device vectors. Location 1374 (SAVLIN) will contain the line number that was declared active.

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

Only one variable in the region subtitled "APT Mailbox-Etable" needs to be set up before running under APT. This variable is:

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

9.1.3 RUNNING UNDER APT

$SWREG (loc. 1142) should be set up prior to running the diagnostic.

DOCUMENT
****
DVDZCA SEQ
****

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

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

11       STARTING PROCEDURE
         LOAD PROGRAM
         START THE PROGRAM AT LOC. 000200
         PROGRAM WILL TYPE DZV11 ECHO/CABLE TEST
         PROGRAM WILL TYPE  WHICH TEST- ECHO OR CABLE
         TYPE IN  E  OR  C   RESPECTIVELY
         PROGRAM WILL TYPE "VECTOR ADDRESS-"
         TYPE IN THE ADDRESS OF THE RECEIVER INTERRUPT VECTOR
         FOR THE DZV11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
         PROGRAM WILL TYPE "CONTROL REGISTER ADDRESS-"
         TYPE IN THE ADDRESS OF THE SYSTEM CONTROL REGISTER
         FOR THE DZV11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
         PROGRAM WILL TYPE "LINE NUMBER-"
         TYPE IN THE LINE NUMBER TO BE TESTED (IN OCTAL)
          FOLLOWED BY <CARRIAGE RETURN>
         PROGRAM WILL TYPE "BAUD RATE-"
         TYPE IN THE BAUD RATE OF THE DZV11 TERMINAL
          FOLLOWED BY <CARRIAGE RETURN>
         THE FOLLOWING BAUD RATES ARE ACCEPTED IN DECIMAL
                           50
                           75
                           110
                           135     (ROUNDED OFF   134.5)
                           150
                           300
                           600
                           1200
                           1800
                           2000
                           2400
                           3600
                           4800
                           7200
                           9600
         ALL OTHERS ARE REJECTED

47       PROGRAM WILL TYPE "ECHO"  OR "CABLE TEST" TO INDICATE THAT TESTING HAS STARTED

74       INITIAL ADDRESS OF THE STACK POINTER *** 1120 ***
```

```
79      MISCELLANEOUS DEFINITIONS

91      GENERAL PURPOSE REGISTER DEFINITIONS

103     PRIORITY LEVEL DEFINITIONS

113     "SWITCH REGISTER" SWITCH DEFINITIONS

141     DATA BIT DEFINITIONS (BIT00 TO BIT15)

169     BASIC "CPU" TRAP VECTOR ADDRESSES

384                             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

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

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

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

488     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:

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

873     INCREMENT THE PASS NUMBER ($PASS)
        IF THERES A MONITOR GO TO IT
        IF THERE ISN'T JUMP TO XBEGIN

995     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

         1728     *********    ECHO TEST    **********
                  THIS TEST WILL ACCEPT 1 CHARACTER AT A TIME
                  (IN INTERRUPT MODE) AND TRANSMIT THAT SAME CHARACTER,
                  ONE LINE AT A TIME, ANY LINE 0 THRU 7 (OCTAL)

         1799     *********    CABLE TEST    **********
                  THIS TEST TRANSMITS A BINARY COUNT PATTERN
                  VIA INTERRUPT MODE TO THE RECEIVER
                  ...THE LINE UNDER TEST MUST BE TERMINATED WITH THE TEST CONNECTOR

         1808     TEST TO VERIFY THAT SETTING DTR FOR A GIVEN LINE
                  WILL BRING UP "CO" AND "RING" FOR THE SAME LINE
                  JUMPERS W1,W2,W3 AND W4 MUST BE INSTALLED ON THE
                  INTERFACE MODULE OTHERWISE AN ERROR REPORT WILL RESULT.
```

```
                              .TITLE  MD-11-DVDZC-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
                                      ;*START THE PROGRAM AT LOC. 000200
                                      ;*PROGRAM WILL TYPE DZV11 ECHO/CABLE TEST
                                      ;*PROGRAM WILL TYPE  WHICH TEST- ECHO OR CABLE
                                      ;*TYPE IN  E  OR  C   RESPECTIVELY
                                      ;*PROGRAM WILL TYPE "VECTOR ADDRESS-"
                                      ;*TYPE IN THE ADDRESS OF THE RECEIVER INTERRUPT VECTOR
                                      ;*FOR THE DZV11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
                                      ;*PROGRAM WILL TYPE "CONTROL REGISTER ADDRESS-"
                                      ;*TYPE IN THE ADDRESS OF THE SYSTEM CONTROL REGISTER
                                      ;*FOR THE DZV11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
                                      ;*PROGRAM WILL TYPE "LINE NUMBER-"
                                      ;*TYPE IN THE LINE NUMBER TO BE TESTED (IN OCTAL)
                                      ;*,FOLLOWED BY <CARRIAGE RETURN>
                                      ;*PROGRAM WILL TYPE "BAUD RATE-"
                                      ;*TYPE IN THE BAUD RATE OF THE DZV11 TERMINAL
                                      ;*,FOLLOWED BY <CARRIAGE RETURN>
                                           ;*THE FOLLOWING BAUD RATES ARE ACCEPTED IN DECIMAL
                                      ;*              50
                                      ;*              75
                                      ;*              110
                                      ;*              135     (ROUNDED OFF   134.5)
                                      ;*              150
                                      ;*              300
                                      ;*              600
                                      ;*              1200
                                      ;*              1800
                                      ;*              2000
                                      ;*              2400
                                      ;*              3600
                                      ;*              4800
                                      ;*              7200
                                      ;*              9600
                                      ;*ALL OTHERS ARE REJECTED

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

                              .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
                      .EQUIV  PS,PSW
        177774        STKLMT= 177774        ;;STACK LIMIT REGISTER
        177772        PIRQ=   177772        ;;PROGRAM INTERRUPT REQUEST REGISTER
        177570        DSWR=   177570        ;;HARDWARE SWITCH REGISTER
        177570        DDISP=  177570        ;;HARDWARE DISPLAY REGISTER

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

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

```

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

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

```

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


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

        005746                  PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
        005726                  POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
        010046                  PUSHR0=10046    ;SAVE R0 ON STACK
        012600                  POPR0=12600     ;RESTORE R0 FROM STACK
        024646                  PUSH2SP=24646   ;DECREMENT STACK TWICE
        022626                  POP2SP=22626    ;INCREMENT STACK TWICE
        000200                  MASK=BIT7       ;SET INTERRUPT MASK (INHIBIT FURTHER INTERRUPTS)
        000000                  CLEAR=0         ;ALLOW INTERRUPTS (CLEAR PROCESSOR STATUS)


                                        ;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
```

```
    225        100000             DVALID=BIT15    ;DATA VALID
    226
    227                                   ;DZVRBUF WORD DEFINITIONS
    228                                   ;------------------------
    229
    230        000000             RL0=0               ;RECEIVER LINE 0
    231        000400             RL1=BIT8            ;RECEIVER LINE 1
    232        001000             RL2=BIT9            ;RECEIVER LINE 2
    233        001400             RL3=BIT9!BIT8       ;RECEIVER LINE 3
    234
    235                                   ;DZVLPR WORD DEFINITIONS
    236                                   ;-----------------------
    237
    238        000000             LP0=0               ;LINE PARAMETER 0
    239        000001             LP1=BIT0            ;LINE PARAMETER 1
    240        000002             LP2=BIT1            ;LINE PARAMETER 2
    241        000003             LP3=BIT1!BIT0       ;LINE PARAMETER 3
    242
    243        000000             FIVE=0              ;FIVE BITS/CHAR,1 STOP BIT
    244        000010             SIX=BIT3            ;SIX BITS/CHAR,1 STOP BIT
    245        000020             SEVEN=BIT4          ;SEVEN BITS/CHAR,1 STOP BIT
    246        000030             EIGHT=BIT4!BIT3     ;EIGHT BITS/CHAR,1 STOP BIT
    247        000040             FIVES=BIT5          ;FIVE BITS/CHAR,2 STOP BITS
    248        000050             SIXS=BIT5!BIT3      ;SIX BITS/CHAR,2 STOP BITS
    249        000060             SEVENS=BIT5!BIT4            ;SEVEN BITS/CHAR, 2 STOP BITS
    250        000070             EIGHTS=BIT5!BIT4!BIT3       ;EIGHT BITS/CHAR, 2 STOP BITS
    251
    252        000100             PARITY=BIT6                 ;PARITY ENABLED
    253        000200             ODDPAR=BIT7                 ;ODD PARITY ENABLED
    254        000000             ONESTOP=0                   ;ONE STOP BIT ENABLED
    255        000040             TWOSTOP=BIT5                ;TWO STOP BITS ENABLED
    256        000000             EVEPAR=0                    ;EVEN PARITY ENABLED
    257        010000             RCVON=BIT12                 ;ENABLE RECEIVER (RECEIVER ON)
    258
    259        000000             S50=0                       ;SPEED 50 BAUD
    260        000400             S75=BIT8                    ;SPEED 75 BAUD
    261        001000             S110=BIT9                   ;SPEED 110 BAUD
    262        001400             S134=BIT9!BIT8              ;SPEED 134.5 BAUD
    263        002000             S150=BIT10                  ;SPEED 150 BAUD
    264        002400             S300=BIT10!BIT8             ;SPEED 300 BAUD
    265        003000             S600=BIT10!BIT9             ;SPEED 600 BAUD
    266        003400             S1200=BIT10!BIT9!BIT8       ;SPEED 1200 BAUD
    267        004000             S1800=BIT11                 ;SPEED 1800 BAUD
    268        004400             S2000=BIT11!BIT8            ;SPEED 2000 BAUD
    269        005000             S2400=BIT11!BIT9            ;SPEED 2400 BAUD
    270        005400             S3600=BIT11!BIT9!BIT8       ;SPEED 3600 BAUD
    271        006000             S4800=BIT11!BIT10           ;SPEED 4800 BAUD
    272        006400             S7200=BIT11!BIT10!BIT8      ;SPEED 7200 BAUD
    273        007000             S9600=BIT11!BIT10!BIT9      ;SPEED 9600 BAUD
    274        007400             S19200=BIT11!BIT10!BIT9!BIT8     ;SPEED 19200 BAUD
    275
    276                                   ;DZVTCR BIT DEFINITIONS
    277                                   ;----------------------
    278        000001             TCR0=BIT0                   ;ENABLE TRANSMISSION ON LINE 0
    279        000002             TCR1=BIT1                   ;ENABLE TRANSMISSION ON LINE 1
    280        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
                                                ;-------------------------

        000024                                  .=24
000024  005576                                  $PWRDN                  ;POWER FAIL HANDLER
000026  000340                                  340                     ;SERVICE AT PRIORITY LEVEL 7
000030  004704                                  $ERROR                  ;ERROR HANDLER
000032  000340                                  340                     ;SERVICE AT PRIORITY LEVEL 7
000034  004476                                  .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  002644                                  $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-DVDZCA/<200>/DZV11 ECHO AND CABLE TESTS  /<200>
```

```
        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" ABOVE
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  000000                   $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  000000                   $DEVM:  .WORD    ADEVM      ;;DEVICE MAP
001200  000000                   $CDW1:  .WORD    ACDW1      ;;CONTROLLER DESCRIPTION WORD#1
001202  000000                   $CDW2:  .WORD    ACDW2      ;;CONTROLLER DESCRIPTION WORD#2
001204  000000                   $DDW0:  .WORD    ADDW0      ;;DEVICE DESCRIPTOR WORD#0
001206  000000                   $DDW1:  .WORD    ADDW1      ;;DEVICE DESCRIPTOR WORD#1
001210  000000                   $DDW2:  .WORD    ADDW2      ;;DEVICE DESCRIPTOR WORD#2
001212  000000                   $DDW3:  .WORD    ADDW3      ;;DEVICE DESCRIPTOR WORD#3
001214  000000                   $DDW4:  .WORD    ADDW4      ;;DEVICE DESCRIPTOR WORD#4
001216  000000                   $DDW5:  .WORD    ADDW5      ;;DEVICE DESCRIPTOR WORD#5
```

```
 001220  000000               $DDW6:  .WORD   ADDW6    ;;DEVICE DESCRIPTOR WORD#6
 001222  000000               $DDW7:  .WORD   ADDW7    ;;DEVICE DESCRIPTOR WORD#7
 001224  000000               $DDW8:  .WORD   ADDW8    ;;DEVICE DESCRIPTOR WORD#8
 001226  000000               $DDW9:  .WORD   ADDW9    ;;DEVICE DESCRIPTOR WORD#9
 001230  000000               $DDW10: .WORD   ADDW10   ;;DEVICE DESCRIPTOR WORD#10
 001232  000000               $DDW11: .WORD   ADDW11   ;;DEVICE DESCRIPTOR WORD#11
 001234  000000               $DDW12: .WORD   ADDW12   ;;DEVICE DESCRIPTOR WORD#12
 001236  000000               $DDW13: .WORD   ADDW13   ;;DEVICE DESCRIPTOR WORD#13
 001240  000000               $DDW14: .WORD   ADDW14   ;;DEVICE DESCRIPTOR WORD#14
 001242  000000               $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 ERRCR 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,NO PARIT
        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 NUMBER 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  000000                $TSTM:  .WORD   0.              ;;RUN TIM OF LONGEST TEST
001454  000000                $PASTM: .WORD   0.              ;;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 ADDITIONAL UNIT
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
```

```
   583  001520  000001                 PAR1:     .BLKW    1               ;PARAMETERS
   584  001522  000001                 MANT1:    .BLKW    1               ;MAINTENANCE MODE FOR THIS DEVICE
   585
   586  001524  000001                 DZCR2:    .BLKW    1               ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 2
   587  001526  000001                 DZVC2:    .BLKW    1               ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 2
   588  001530  000001                 LINE2:    .BLKW    1               ;ALL LINES SELECTED
   589  001532  000001                 PAR2:     .BLKW    1               ;PARAMETERS
   590  001534  000001                 MANT2:    .BLKW    1               ;MAINTENANCE MODE FOR THIS DEVICE
   591
   592  001536  000001                 DZCR3:    .BLKW    1               ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 3
   593  001540  000001                 DZVC3:    .BLKW    1               ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 3
   594  001542  000001                 LINE3:    .BLKW    1               ;ALL LINES SELECTED
   595  001544  000001                 PAR3:     .BLKW    1               ;PARAMETERS
   596  001546  000001                 MANT3:    .BLKW    1               ;MAINTENANCE MODE FOR THIS DEVICE
   597
   598  001550  000001                 DZCR4:    .BLKW    1               ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 4
   599  001552  000001                 DZVC4:    .BLKW    1               ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 4
   600  001554  000001                 LINE4:    .BLKW    1               ;ALL LINES SELECTED
   601  001556  000001                 PAR4:     .BLKW    1               ;PARAMETERS
   602  001560  000001                 MANT4:    .BLKW    1               ;MAINTENANCE MODE FOR THIS DEVICE
   603
   604  001562  000001                 DZCR5:    .BLKW    1               ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 5
   605  001564  000001                 DZVC5:    .BLKW    1               ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 5
   606  001566  000001                 LINE5:    .BLKW    1               ;ALL LINES SELECTED
   607  001570  000001                 PAR5:     .BLKW    1               ;PARAMETERS
   608  001572  000001                 MANT5:    .BLKW    1               ;MAINTENANCE MODE FOR THIS DEVICE
   609
   610  001574  000001                 DZCR6:    .BLKW    1               ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 6
   611  001576  000001                 DZVC6:    .BLKW    1               ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 6
   612  001600  000001                 LINE6:    .BLKW    1               ;ALL LINES SELECTED
   613  001602  000001                 PAR6:     .BLKW    1               ;PARAMETERS
   614  001604  000001                 MANT6:    .BLKW    1               ;MAINTENANCE MODE FOR THIS DEVICE
   615
   616  001606  000001                 DZCR7:    .BLKW    1               ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 7
   617  001610  000001                 DZVC7:    .BLKW    1               ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 7
   618  001612  000001                 LINE7:    .BLKW    1               ;ALL LINES SELECTED
   619  001614  000001                 PAR7:     .BLKW    1               ;PARAMETERS
   620  001616  000001                 MANT7:    .BLKW    1               ;MAINTENANCE MODE FOR THIS DEVICE
   621
   622  001620  000001                 DZCR10:   .BLKW    1               ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 10
   623  001622  000001                 DZVC10:   .BLKW    1               ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 10
   624  001624  000001                 LINE10:   .BLKW    1               ;ALL LINES SELECTED
   625  001626  000001                 PAR10:    .BLKW    1               ;PARAMETERS
   626  001630  000001                 MANT10:   .BLKW    1               ;MAINTENANCE MODE FOR THIS DEVICE
   627
   628  001632  000001                 DZCR11:   .BLKW    1               ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 11
   629  001634  000001                 DZVC11:   .BLKW    1               ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 11
   630  001636  000001                 LINE11:   .BLKW    1               ;ALL LINES SELECTED
   631  001640  000001                 PAR11:    .BLKW    1               ;PARAMETERS
   632  001642  000001                 MANT11:   .BLKW    1               ;MAINTENANCE MODE FOR THIS DEVICE
   633
   634  001644  000001                 DZCR12:   .BLKW    1               ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 12
   635  001646  000001                 DZVC12:   .BLKW    1               ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 12
   636  001650  000001                 LINE12:   .BLKW    1               ;ALL LINES SELECTED
   637  001652  000001                 PAR12:    .BLKW    1               ;PARAMETERS
   638  001654  000001                 MANT12:   .BLKW    1               ;MAINTENANCE MODE FOR THIS DEVICE
```

```
 639
 640  001656  000001               DZCR13: .BLKW   1          ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 13
 641  001660  000001               DZVC13: .BLKW   1          ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 13
 642  001662  000001               LINE13: .BLKW   1          ;ALL LINES SELECTED
 643  001664  000001               PAR13:  .BLKW   1          ;PARAMETERS
 644  001666  000001               MANT13: .BLKW   1          ;MAINTENANCE MODE FOR THIS DEVICE
 645
 646  001670  000001               DZCR14: .BLKW   1          ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 14
 647  001672  000001               DZVC14: .BLKW   1          ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 14
 648  001674  000001               LINE14: .BLKW   1          ;ALL LINES SELECTED
 649  001676  000001               PAR14:  .BLKW   1          ;PARAMETERS
 650  001700  000001               MANT14: .BLKW   1          ;MAINTENANCE MODE FOR THIS DEVICE
 651
 652  001702  000001               DZCR15: .BLKW   1          ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 15
 653  001704  000001               DZVC15: .BLKW   1          ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 15
 654  001706  000001               LINE15: .BLKW   1          ;ALL LINES SELECTED
 655  001710  000001               PAR15:  .BLKW   1          ;PARAMETERS
 656  001712  000001               MANT15: .BLKW   1          ;MAINTENANCE MODE FOR THIS DEVICE
 657
 658  001714  000001               DZCR16: .BLKW   1          ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 16
 659  001716  000001               DZVC16: .BLKW   1          ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 16
 660  001720  000001               LINE16: .BLKW   1          ;ALL LINES SELECTED
 661  001722  000001               PAR16:  .BLKW   1          ;PARAMETERS
 662  001724  000001               MANT16: .BLKW   1          ;MAINTENANCE MODE FOR THIS DEVICE
 663
 664  001726  000001               DZCR17: .BLKW   1          ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 17
 665  001730  000001               DZVC17: .BLKW   1          ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 17
 666  001732  000001               LINE17: .BLKW   1          ;ALL LINES SELECTED
 667  001734  000001               PAR17:  .BLKW   1          ;PARAMETERS
 668  001736  000001               MANT17: .BLKW   1          ;MAINTENANCE MODE FOR THIS DEVICE
 669
 670  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
       001742   004572                 .ADVANCE
                104401         SCOP1=TRAP+1                ;CALL TO LOOP ON CURRENT DATA HANDLER
       001744   003104                 .SCOP1
                104402         TYPE=TRAP+2                 ;CALL TO TELETYPE OUTPUT ROUTINE
       001746   003130                 .TYPE
                104403         INSTR=TRAP+3                ;CALL TO ASCII STRING INPUT ROUTINE
       001750   003676                 .INSTR
                104404         INSTER=TRAP+4               ;CALL TO INPUT ERROR HANDLER
       001752   004002                 .INSTER
                104405         PARAM=TRAP+5                ;CALL TO NUMERICAL DATA INPUT ROUTINE
       001754   004022                 .PARAM
                104406         SETFLG=TRAP+6               ;CALL TO  SET FLAG ROUTINE
       001756   006432                 .SETFLG
                104407         SAVOS=TRAP+7                ;CALL TO REGISTER SAVE ROUTINE
       001760   004222                 .SAVOS
                104410         RESOS=TRAP+10               ;CALL TO REGISTER RESTORE ROUTINE
       001762   004262                 .RESOS
                104411         CONVRT=TRAP+11              ;CALL TO DATA OUTPUT ROUTINE
       001764   004314                 .CONVRT
                104412         CNVRT=TRAP+12               ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
       001766   004320                 .CNVRT
                104413         DEVICE.CLR=TRAP+13                  ;CALL TO ISSUE A DEVICE CLEAR
       001770   004520                 .DEVICE.CLR
                104414         DELAY=TRAP+14               ;CALL TO DELAY FOR FAST CPU'S
       001772   004552                 .DELAY
                104415         PARMD=TRAP+15               ;CONVERT DECIMAL STRING TO OCTAL
       001774   002710                 .PARMD
                104416         PAWCH=TRAP+16               ;SET FLAG    ECHO OR  CABLE
       001776   006552                 .PAWCH
                104417         DCLASM=TRAP+17              ;CLEAR DEVICE, SET MAINT. BIT IF I MODE
       002000   004540                 .DCLASM
                104420         SHIFT=TRAP+20               ;CALL TO ROTATE LINE POINTER
       002002   004604                 .SHIFT
                104421         LPRSET=TRAP+21              ;CALL TO SET UP LPR DEVICE REGISTER
       002004   004622                 .LPRSET
                104422         BUFSET=TRAP+22              ;CALL TO ZERO BUFFER AREA
       002006   004662                 .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 ONLY
 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


```

```
  743
  744                                            ;TIME TABLE FOR RELATIVE TIMING TESTS
  745                                            ;------------------------------------
  746                                            ;
  747  002050                           TMTBL:
  748  002050   000000                  T50:     0
  749  002052   000000                  T75:     0
  750  002054   000000                  T110:    0
  751  002056   000000                  T134:    0
  752  002060   000000                  T150:    0
  753  002062   000000                  T300:    0
  754  002064   000000                  T600:    0
  755  002066   000000                  T1200:   0
  756  002070   000000                  T1800:   0
  757  002072   000000                  T2000:   0
  758  002074   000000                  T2400:   0
  759  002076   000000                  T3600:   0
  760  002100   000000                  T4800:   0
  761  002102   000000                  T7200:   0
  762  002104   000000                  T9600:   0
  763  002106   000000                  TEIGHT:0
  764  002110   000000                  TSEVEN:  0
  765  002112   000000                  TSIX:    0
  766  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  000005                  .START: RESET                     ;CLEAR THE WORLD
002120  012706  001120                  MOV     #STACK,SP         ;SET UP PROCESSOR STACK
002124  106427  000200                  MTPS    #MASK             ;LOCK OUT INTERRUPTS
002130  012737  005576  000024          MOV     #$PWRDN,@#24      ;SET UP FOR POWER FAIL
002136  012737  002116  001252          MOV     #.START,$LPADR    ;SET UP IN CASE OF POWER FAIL
002144  105737  001141                  TSTB    $ENVM             ;RUNNING UNDER APT?
002150  100004                          BPL     1$                ;IF NOT SKIP SWR SETUP FOR APT
002152  012737  001142  001304          MOV     #$SWREG,SWR       ;SETUP FOR APT SWR
002160  000403                          BR      2$                ;SKIP SOFTWARE SWR SETUP
002162  012737  000176  001304  1$:     MOV     #SWREG,SWR        ;SETUP SOFTWARE SWITCH REGISTER OTHERWISE
002170  012737  000174  001306  2$:     MOV     #DISPREG,DISPLAY
                                                                     ;SETUP DISPLAY REGISTER
002176  005037  007024                  CLR     STFLG             ;CLEAR TEST START FLAG
002202  005037  001126                  CLR     $PASS             ;CLEAR PASS COUNT
002206  005037  001256                  CLR     $ERTTL            ;CLEAR ERROR COUNT
002212  105037  001247                  CLRB    $ERFLG            ;CLEAR ERROR FLAG
002216  005037  001246                  CLR     $TSTNM            ;CLEAR TEST NO. INDICATOR
002222  005037  007030                  CLR     LAST              ;CLEAR LAST ERROR PC
002226  105737  001422                  TSTB    INIFLG            ;HAS TITLE BEEN TYPED YET?
002232  001010                          BNE     VEC1              ;IF YES SKIP PRINTING AGAIN
002234  023727  000042  002644          CMP     @#42,#$ENDAD      ;RUNNING UNDER ACT?
002242  001402                          BEQ     3$                ;IF YES DON'T PRINT TITLE
002244  104402  001000                  TYPE    ,MTITLE           ;PRINT TITLE
002250  105337  001422          3$:     DECB    INIFLG            ;INDICATE TITLE ALREADY TYPED
002254  012701  000300          VEC1:   MOV     #300,R1
002260  012702  000302                  MOV     #302,R2
002264  010221                  1$:     MOV     R2,(R1)+          ;RESTORE TRAPCATCHER
002266  005022                          CLR     (R2)+             ;IN FLOATING VECTOR AREA
002270  022122                          CMP     (R1)+,(R2)+       ;UPDATE THE POINTERS
002272  020127  001000                  CMP     R1,#1000
002276  001372                          BNE     1$
002300  104403                          INSTR                     ;INPUT ADDRESS OF DEVICE VECTOR
002302  007056                          MVECTOR                   ;MESSAGE "VECTOR ADDRESS-"
002304  104405                          PARAM                     ;CONVERT STRING TO OCTAL
002306  000300                          300                       ;LOW LIMIT
002310  000770                          770                       ;HIGH  LIMIT
002312  002040                          DZVRIV                    ;LOCATIONS TO BE FILLED
002314     003                  .BYTE   3                         ;LSB MASK
002315     004                  .BYTE   4                         ;NUMBER OF LOCATIONS
002316  104403                          INSTR                     ;INPUT ADDRESS OF DEVICE CSR
002320  007100                          MREGAD                    ;MESSAGE "CONTROL REGISTER ADDRESS-"
002322  104405                          PARAM                     ;CONVERT STRING TO OCTAL
002324  160000                          160000                    ;LOW LIMIT
002326  163770                          163770                    ;HIGH LIMIT
002330  001174                          $BASE                     ;LOCATION TO BE FILLED
002332     007                  .BYTE   7                         ;LSB MASK
002333     001                  .BYTE   1                         ;NUMBER OF LOCATIONS
002334  004737  007660                  JSR     PC,DZVLEV         ;GO BUILD DEVICE POINTERS
002340  104403                          INSTR                     ;INPUT WHICH TEST YOU ARE RUNNING
```

```
002342  007265                            MWHICH                  ;ECHO OR CABLE
002344  104416                            PAWCH                   ;SET FLAG
002346  007022                            WCHFLG                  ;THIS FLAG
002350  104403                  BAUD:     INSTR                   ;INPUT BAUD RATE
002352  007206                            MSPEED                  ;MESSAGE "BAUD RATE-"
002354  104415                            PARMD                   ;CONVERT DECIMAL STRING TO OCTAL
002356  000062                            50.                     ;LOW LIMIT
002360  022600                            9600.                   ;HIGH LIMIT
002362  007040                            LINESP                  ;LOCATION TO BE FILLED
002364     000                  .BYTE     0                       ;LSB MASK
002365     001                  .BYTE     1                       ;NUMBER OF LOCATIONS
002366  104413                  LINEX:    DEVICE.CLR              ;CLEAR DEVICE
002370  005037  007024                    CLR     STFLG           ;CLEAR PROGRAM START FLAG
002374  104403                            INSTR                   ;INPUT LINE NUMBER
002376  007176                            MLINE                   ;MESSAGE "LINE NUMBER-"
002400  104405                            PARAM                   ;CONVERT  STRING TO OCTAL
002402  000000                            0                       ;LOW LIMIT
002404  000003                            3                       ;HIGH LIMIT
002406  001374                            SAVLIN                  ;LOCATION TO BE FILLED
002410     000                  .BYTE     0                       ;LSB MASK
002411     001                            .BYTE   1               ;NUMBER OF LOCATIONS
002412  004537  006626                    JSR     R5,SET

002416  106427  000200          XBEGIN:   MTPS    #MASK           ;LOCK OUT INTERRUPTS
002422  012706  001120                    MOV     #STACK,SP       ;SET UP PROCESSOR STACK
002426  005037  007026                    CLR     LOCKUP          ;CLEAR TIMEOUT
002432  005737  007022                    TST     WCHFLG          ;ECHO OR CABLE TEST ?
002436  001413                            BEQ     2$              ;ECHO
002440  012737  010374  001252            MOV     #TST2,$LPADR    ;CABLE TEST
002446  005737  007024                    TST     STFLG           ;ARE YOU LOOPING ?
002452  001017                            BNE     1$              ;YES
002454  005137  007024                    COM     STFLG           ;NO
002460  104402  007360                    TYPE    ,MCABLE         ;TYPE CABLE TEST
002464  000412                            BR      1$
002466  012737  010020  001252  2$:       MOV     #TST1,$LPADR    ;SET UP ECHO TEST
002474  005737  007024                    TST     STFLG           ;ARE YOU LOOPING ?
002500  001004                            BNE     1$              ;YES
002502  005137  007024                    COM     STFLG           ;NO
002506  104402  007333                    TYPE    ,MTERM          ;TYPE ECHO TEST
002512                          1$:
002512  000177  176534          RESTART:JMP       @$LPADR         ;START TESTING,THIS LOCATION IS ALSO
                                                                  ;USED BY THE POWER UP ROUTINE
```

```
                                                ;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 XBEGIN

002516                                  $EOP:
002516  005037  001262                          CLR     $ERRPC                  ;CLEAR LAST ERROR PC
002522  105037  001247                          CLRB    $ERFLG                  ;CLEAR ERROR FLAG
002526  104402  006013                          TYPE    ,MEPASS                 ;TYPE END PASS
002532  104402  006175                          TYPE    ,MCSRX                  ;TYPE CSR
002536  104412  002660                          CNVRT   ,XCSR                   ;SHOW IT
002542  104402  006203                          TYPE    ,MVECX                  ;TYPE VECTOR
002546  104412  002666                          CNVRT   ,XVEC                   ;SHOW IT
002552  005237  001126                          INC     $PASS                   ;RAISE PASS COUNT
002556  104402  006211                          TYPE    ,MPASSX                 ;TYPE PASSES
002562  104412  002674                          CNVRT   ,XPASS                  ;SHOW IT
002566  005337  001126                          DEC     $PASS                   ;RESTORE PASS COUNT
002572  104402  006222                          TYPE    ,MERRX                  ;TYPE ERRORS
002576  104412  002702                          CNVRT   ,XERR                   ;SHOW IT
002602  005037  001354                          CLR     $TIMES                  ;;ZERO THE NUMBER OF ITERATIONS
002606  005237  001126                          INC     $PASS                   ;;INCREMENT THE PASS NUMBER
002612  042737  100000  001126                  BIC     #100000,$PASS           ;;DON'T ALLOW A NEG. NUMBER
002620  005327                                  DEC     (PC)+                   ;;LOOP?
002622  000001                          $EOPCT: .WORD   1
002624  003013                                  BGT     $DOAGN                  ;;YES
002626  012737                                  MOV     (PC)+,@(PC)+            ;;RESTORE COUNTER
002630  000001                          $ENDCT: .WORD   1
002632  002622                                  $EOPCT
002634  013700  000042                  $GET42: MOV     @#42,R0                 ;;GET MONITOR ADDRESS
002640  001405                                  BEQ     $DOAGN                  ;;BRANCH IF NO MONITOR
002642  000005                                  RESET                           ;;CLEAR THE WORLD
002644  004710                          $ENDAD: JSR     PC,(R0)                 ;;GO TO MONITOR
002646  000240                                  NOP                             ;;SAVE ROOM
002650  000240                                  NOP                             ;;FOR
002652  000240                                  NOP                             ;;ACT11
002654                                  $DOAGN:
002654  000137                                  JMP     @(PC)+                  ;;RETURN
002656  002416                          $RTNAD: .WORD   XBEGIN

002660  000001                          XCSR:   1
002662     006     002                          .BYTE   6,2
002664  002010                                  DZVCSR
002666  000001                          XVEC:   1
002670     003     002                          .BYTE   3,2
002672  002040                                  DZVRIV
002674  000001                          XPASS:  1
002676     006     002                          .BYTE   6,2
002700  001126                                  $PASS
002702  000001                          XERR:   1
```

```
002704     006     002                   .BYTE   6,2
002706  001256                           $ERTTL

                                         ;CONVERT DECIMAL ASCII STRING TO OCTAL
002710  011605                  .PARMD:  MOV     (SP),R5
002712  012537  003074                   MOV     (R5)+,6$
002716  012537  003076                   MOV     (R5)+,7$
002722  012537  003100                   MOV     (R5)+,8$
002726  112537  003102                   MOVB    (R5)+,9$
002732  112537  003103                   MOVB    (R5)+,10$
002736  010516                           MOV     R5,(SP)
002740  005005                  2$:      CLR     R5
002742  012704  007512                   MOV     #INBUF,R4
002746  122714  000015                   CMPB    #15,(R4)
002752  001424                           BEQ     3$
002754  121427  000060          1$:      CMPB    (R4),#'0
002760  002421                           BLT     3$
002762  121427  000071                   CMPB    (R4),#'9
002766  003016                           BGT     3$
002770  142714  000060                   BICB    #'0,(R4)
002774  005002                           CLR     R2
002776  152402                           BISB    (R4)+,R2
003000  060205                           ADD     R2,R5
003002  122714  000015                   CMPB    #15,(R4)
003006  001410                           BEQ     4$
003010  006305                           ASL     R5          ;X2
003012  010502                           MOV     R5,R2       ;SAVE X2
003014  006305                           ASL     R5          ;X4
003016  006305                           ASL     R5          ;X8
003020  060205                           ADD     R2,R5       ;TIMES 10
003022  000754                           BR      1$
003024  104404                  3$:      INSTER
003026  000744                           BR      2$

                                         ;TEST TO SEE IF NUMBER IS WITHIN LIMITS

003030  020537  003076          4$:      CMP     R5,7$
003034  101373                           BHI     3$
003036  020537  003074                   CMP     R5,6$
003042  103770                           BLO     3$
003044  133705  003102                   BITB    9$,R5
003050  001365                           BNE     3$

                                         ;STORE NUMBER AT SPECIFIED ADDRESS

003052  013704  003100                   MOV     8$,R4
003056  010524                  5$:      MOV     R5,(R4)+
003060  062705  000002                   ADD     #2,R5
003064  105337  003103                   DECB    10$
003070  001372                           BNE     5$
003072  000002                           RTI
003074  000000                  6$:      0
003076  000000                  7$:      0
003100  000000                  8$:      0
003102     000                  9$:      .BYTE 0
003103     000                  10$:     .BYTE 0
```

```

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

003104  032777  001000  176172  .SCOP1: BIT     #SW09,@SWR              ;IS SW09=1(SET)?
003112  001405                          BEQ     1$                      ;BR IF NOT SET.
003114  005737  001364                  TST     LOCK                    ;IS THERE A TIGHT LOOP SPECIFIED?
003120  001402                          BEQ     1$                      ;IF NO, RETURN
003122  013716  001364                  MOV     LOCK,(SP)               ;IF YES, GOTO THE ADDRESS IN LOCK.
003126  000002                  1$:     RTI                             ;GO BACK.

003130  032777  010000  176146  .TYPE:  BIT     #SW12,@SWR              ;INHIBIT ALL PRINTOUT??
003136  001403                          BEQ     $TYPE                   ;IF NOT, GO TYPE
003140  062716  000002                  ADD     #2,(SP)                 ;SKIP OVER MESSAGE POINTER
003144  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
                                ;*

003146  105737  001323          $TYPE:  TSTB    $TPFLG                  ;;IS THERE A TERMINAL?
003152  100002                          BPL     1$                      ;;BR IF YES
003154  000000                          HALT                            ;;HALT HERE IF NO TERMINAL
003156  000430                          BR      3$                      ;;LEAVE
003160  010046                  1$:     MOV     R0,-(SP)                ;;SAVE R0
003162  017600  000002                  MOV     @2(SP),R0               ;;GET ADDRESS OF ASCIZ STRING
003166  122737  000001  001140          CMPB    #APTENV,$ENV            ;;RUNNING IN APT MODE
003174  001011                          BNE     62$                     ;;NO,GO CHECK FOR APT CONSOLE
003176  132737  000100  001141          BITB    #APTSPOOL,$ENVM         ;;SPOOL MESSAGE TO APT
003204  001405                          BEQ     62$                     ;;NO,GO CHECK FOR CONSOLE
003206  010037  003216                  MOV     R0,61$                  ;;SETUP MESSAGE ADDRESS FOR APT
003212  004737  003436                  JSR     PC,$ATY3                ;;SPOOL MESSAGE TO APT
003216  000000                  61$:    .WORD   0                       ;;MESSAGE ADDRESS
003220  132737  000040  001141  62$:    BITB    #APTCSUP,$ENVM          ;;APT CONSOLE SUPPRESSED
003226  001003                          BNE     60$                     ;;YES,SKIP TYPE OUT
003230  112046                  2$:     MOVB    (R0)+,-(SP)             ;;PUSH CHARACTER TO BE TYPED ONTO STACK
003232  001005                          BNE     4$                      ;;BR IF IT ISN'T THE TERMINATOR
003234  005726                          TST     (SP)+                   ;;IF TERMINATOR POP IT OFF THE STACK
003236  012600                  60$:    MOV     (SP)+,R0                ;;RESTORE R0
003240  062716  000002          3$:     ADD     #2,(SP)                 ;;ADJUST RETURN PC
003244  000002                          RTI                             ;;RETURN
003246  122716  000011          4$:     CMPB    #HT,(SP)                ;;BRANCH IF <HT>
003252  001430                          BEQ     8$
003254  122716  000200                  CMPB    #CRLF,(SP)              ;;BRANCH IF NOT <CRLF>
```

```
003260  001006                                  BNE     5$
003262  005726                                  TST     (SP)+               ;;POP  <CR><LF> EQUIV
003264  104402                                  TYPE                        ;;TYPE A CR AND LF
003266  001357                                  $CRLF
003270  105037  003424                          CLRB    $CHARCNT            ;;CLEAR CHARACTER COUNT
003274  000755                                  BR      2$                  ;;GET NEXT CHARACTER
003276  004737  003360                  5$:     JSR     PC,$TYPEC           ;;GO TYPE THIS CHARACTER
003302  123726  001322                  6$:     CMPB    $FILLC,(SP)+        ;;IS IT TIME FOR FILLER CHARS.?
003306  001350                                  BNE     2$                  ;;IF NO GO GET NEXT CHAR.
003310  013746  001320                          MOV     $NULL,-(SP)         ;;GET # OF FILLER CHARS. NEEDED
                                                                            ;;AND THE NULL CHAR.
003314  105366  000001                  7$:     DECB    1(SP)               ;;DOES A NULL NEED TO BE TYPED?
003320  002770                                  BLT     6$                  ;;BR IF NO--GO POP THE NULL OFF OF STACK
003322  004737  003360                          JSR     PC,$TYPEC           ;;GO TYPE A NULL
003326  105337  003424                          DECB    $CHARCNT            ;;DO NOT COUNT AS A COUNT
003332  000770                                  BR      7$                  ;;LOOP

                                        ;HORIZONTAL TAB PROCESSOR

003334  112716  000040                  8$:     MOVB    #' ,(SP)            ;;REPLACE TAB WITH SPACE
003340  004737  003360                  9$:     JSR     PC,$TYPEC           ;;TYPE A SPACE
003344  132737  000007  003424                  BITB    #7,$CHARCNT         ;;BRANCH IF NOT AT
003352  001372                                  BNE     9$                  ;;TAB STOP
003354  005726                                  TST     (SP)+               ;;POP SPACE OFF STACK
003356  000724                                  BR      2$                  ;;GET NEXT CHARACTER
003360  105777  175730                  $TYPEC: TSTB    @$TPS               ;;WAIT UNTIL PRINTER IS READY
003364  100375                                  BPL     $TYPEC
003366  116677  000002  175722                  MOVB    2(SP),@$TPB         ;;LOAD CHAR TO BE TYPED INTO DATA REG.
003374  122766  000015  000002                  CMPB    #CR,2(SP)           ;;IS CHARACTER A CARRIAGE RETURN?
003402  001003                                  BNE     1$                  ;;BRANCH IF NO
003404  105037  003424                          CLRB    $CHARCNT            ;;YES--CLEAR CHARACTER COUNT
003410  000406                                  BR      $TYPEX              ;;EXIT
003412  122766  000012  000002          1$:     CMPB    #LF,2(SP)           ;;IS CHARACTER A LINE FEED?
003420  001402                                  BEQ     $TYPEX              ;;BRANCH IF YES
003422  105227                                  INCB    (PC)+               ;;COUNT THE CHARACTER
003424  000000                          $CHARCNT:.WORD  0                   ;;CHARACTER COUNT STORAGE
003426  000207                          $TYPEX: RTS     PC

                                        .SBTTL  APT COMMUNICATIONS ROUTINE

                                        ;;********************************************************************
003430  112737  000001  003674          $ATY1:  MOVB    #1,$FFLG            ;;TO REPORT FATAL ERROR
003436  112737  000001  003672          $ATY3:  MOVB    #1,$MFLG            ;;TO TYPE A MESSAGE
003444  000403                                  BR      $ATYC
003446  112737  000001  003674          $ATY4:  MOVB    #1,$FFLG            ;;TO ONLY REPORT FATAL ERROR
003454                                  $ATYC:
003454  010046                                  MOV     R0,-(SP)            ;;PUSH R0 ON STACK
003456  010146                                  MOV     R1,-(SP)            ;;PUSH R1 ON STACK
003460  105737  003672                          TSTB    $MFLG               ;;SHOULD TYPE A MESSAGE?
003464  001450                                  BEQ     5$                  ;;IF NOT: BR
003466  122737  000001  001140                  CMPB    #APTENV,$ENV        ;;OPERATING UNDER APT?
003474  001031                                  BNE     3$                  ;;IF NOT: BR
003476  132737  000100  001141                  BITB    #APTSPOOL,$ENVM     ;;SHOULD SPOOL MESSAGES?
003504  001425                                  BEQ     3$                  ;;IF NOT: BR
003506  017600  000004                          MOV     @4(SP),R0           ;;GET MESSAGE ADDR.
003512  062766  000002  000004                  ADD     #2,4(SP)                    ;;BUMP RETURN ADDR.
```

```
003520  005737  001120          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
003524  001375                          BNE     1$              ;;IF NOT:  WAIT
003526  010037  001134                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
003532  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
003534  001376                          BNE     2$
003536  163700  001134                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
003542  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
003544  010037  001136                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
003550  012737  000004  001120          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
003556  000413                          BR      5$
003560  017637  000004  003604  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
003566  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
003574  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
003600  004737  003146                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
003604  000000                  4$:     .WORD   0
003606                          5$:
003606  105737  003674          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
003612  001416                          BEQ     12$             ;;IF NOT:  BR
003614  005737  001140                  TST     $ENV            ;;RUNNING UNDER APT?
003620  001413                          BEQ     12$             ;;IF NOT:  BR
003622  005737  001120          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
003626  001375                          BNE     11$             ;;IF NOT:  WAIT
003630  017637  000004  001122          MOV     @4(SP),$FATAL   ;;GET ERROR #
003636  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
003644  005237  001120                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
003650  105037  003674          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
003654  105037  003673                  CLRB    $LFLG           ;;CLEAR LOG FLAG
003660  105037  003672                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
003664  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
003666  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
003670  000207                          RTS     PC              ;;RETURN
003672     000                  $MFLG:  .BYTE   0               ;;MESSG. FLAG
003673     000                  $LFLG:  .BYTE   0               ;;LOG FLAG
003674     000                  $FFLG:  .BYTE   0               ;;FATAL FLAG
        003676                          .EVEN
        000200                  APTSIZE=200
        000001                  APTENV=001
        000100                  APTSPOOL=100
        000040                  APTCSUP=040


                                        ;STRING INPUT ROUTINE
                                        ;-------------------------

003676  010346                  .INSTR: MOV     R3,-(SP)        ;SAVE R3 ON STACK
003700  010446                          MOV     R4,-(SP)        ;SAVE R4 ON STACK
003702  017637  000004  003720          MOV     @4(SP),.MSG     ;GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
003710  062766  000002  000004          ADD     #2,4(SP)        ;POINT TO INSTRUCTION AFTER ADDRESS POINTER
003716  104402                  .INST1: TYPE                    ;PRINT THE MESSAGE
003720  000000                  .MSG:   0                       ;MESSAGE IS POINTED TO FROM HERE
003722  012704  007512                  MOV     #INBUF,R4       ;POINT R4 TO THE INPUT BUFFER
003726  012703  000007                  MOV     #7,R3           ;SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
003732  105777  175352          1$:     TSTB    @$TKS           ;HAS A CHARACTER BEEN RECEIVED?
003736  100375                          BPL     1$              ;IF NO, KEEP WAITING FOR IT
003740  117714  175346                  MOVB    @$TKB,(R4)      ;IF YES, SAVE IT IN THE INPUT BUFFER
003744  142714  000200                  BICB    #200,(R4)       ;KEEP ONLY THE 7-BIT ASCII INFORMATION
003750  122427  000015                  CMPB    (R4)+,#15       ;IS THIS CHARACTER A LINE FEED?
```

```
003754  001417                           BEQ     INSTR2           ;IF SO, TERMINATE THE INPUT SEQUENCE
003756  105777  175332          2$:      TSTB    @$TPS            ;IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
003762  100375                           BPL     2$               ;IF WE CAN'T, WAIT UNTIL WE CAN
003764  017777  175322  175324           MOV     @$TKB,@$TPB      ;ECHO THE CHARACTER BACK
003772  005303                           DEC     R3               ;REDUCE THE NUMBER OF CHARACTERS RECEIVED
003774  001356                           BNE     1$               ;IF WE DON'T HAVE 7, GO GET SOME MORE
003776  012604                           MOV     (SP)+,R4         ;IF WE HAVE 7, RESTORE R4
004000  012603                           MOV     (SP)+,R3         ;RESTORE R3
004002  010346                  .INSTE:  MOV     R3,-(SP)         ;SAVE R3 ON THE STACK
004004  010446                           MOV     R4,-(SP)         ;SAVE R4 ON THE STACK
004006  104402  001356                   TYPE    ,$QUES           ;PRINT A QUESTION MARK... WHAT'S GOING ON?
004012  000741                           BR      .INSTI           ;GO PRINT THE MESSAGE AGAIN
004014  012604                  INSTR2:  MOV     (SP)+,R4         ;RESTORE R4
004016  012603                           MOV     (SP)+,R3         ;RESTORE R3
004020  000002                           RTI                      ;RETURN TO THE MAIN PROCEDURE

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

004022  010546                  .PARAM:  MOV     R5,-(SP)         ;SAVE R5 ON THE STACK
004024  010446                           MOV     R4,-(SP)         ;SAVE R4 ON THE STACK
004026  016605  000004                   MOV     4(SP),R5         ;GET THE SETUP INFORMATION POINTER
004032  012537  004212                   MOV     (R5)+,LOLIM      ;SET THE LOW LIMIT FOR THE INPUT
004036  012537  004214                   MOV     (R5)+,HILIM      ;SET THE HIGH LIMIT FOR THE INPUT
004042  012537  004216                   MOV     (R5)+,DEVADR     ;SAVE THE ADDRESS WHERE THE RESULT WILL BE STORED
004046  112537  004220                   MOVB    (R5)+,LOBITS     ;GET THE MASK OF THE INCORRECT BITS
004052  112537  004221                   MOVB    (R5)+,ADRCNT     ;GET THE COUNT OF ITEMS TO BE STORED
004056  010566  000004                   MOV     R5,4(SP)         ;POINT TO WHERE MAIN LINE PROGRAM WILL RESUME
004062  005005                  PARAM1:  CLR     R5               ;INITIALIZE THE ASCII TO OCTAL RESULT WORD
004064  012704  007512                   MOV     #INBUF,R4        ;POINT TO THE INPUT BUFFER
004070  122714  000015                   CMPB    #15,(R4)         ;IS THIS CHARACTER A CARRIAGE RETURN?
004074  001420                           BEQ     PARERR           ;IF SO, PRINT THE MESSAGE AGAIN
004076  121427  000060          1$:      CMPB    (R4),#60         ;IS THIS CHARACTER BELOW THE NUMERIC RANGE?
004102  002415                           BLT     PARERR           ;IF SO, GO PRINT THE MESSAGE AGAIN
004104  121427  000067                   CMPB    (R4),#67         ;IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
004110  003012                           BGT     PARERR           ;IF SO, GO PRINT THE MESSAGE AGAIN
004112  142714  000060                   BICB    #60,(R4)         ;ISOLATE THE NUMBER THE CHARACTER REPRESENTS
004116  152405                           BISB    (R4)+,R5         ;CONCATENATE THESE BITS TO THE ALREADY EXISTING STRING
004120  122714  000015                   CMPB    #15,(R4)         ;IS THE NEXT CHARACTER A CARRIAGE RETURN?
004124  001406                           BEQ     LIMITS           ;IF SO, GO SEE IF NUMBER IS WITHIN LIMITS
004126  006305                           ASL     R5               ;CLEAR BIT POSITION 0, MOVE EXISTING STRING TO LEFT
004130  006305                           ASL     R5               ;CLEAR POSITION 1, MOVE STRING TO LEFT AGAIN
004132  006305                           ASL     R5               ;MOVE THE STRING ONE MORE TIME TO MAKE ROOM FOR
                                                                  ;NEXT THREE BITS
004134  000760                           BR      1$               ;GO GET THE NEXT CHARACTER
004136  104404                  PARERR:  INSTER                   ;THERE WAS AN ERROR... GO PRINT MESSAGE AGAIN
004140  000750                           BR      PARAM1           ;TRY GETTING THE PARAMETERS AGAIN

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

004142  020537  004214          LIMITS:  CMP     R5,HILIM         ;DOES RESULT EXCEED ITS MAXIMUM CORRECT VALUE?
004146  101373                           BHI     PARERR           ;IF YES, GO PRINT THE MESSAGE AGAIN
004150  020537  004212                   CMP     R5,LOLIM         ;IS THE RESULT LOWER THAN ALLOWED?
004154  103770                           BLO     PARERR           ;IF YES, GO PRINT THE MESSAGE AGAIN
004156  133705  004220                   BITB    LOBITS,R5        ;ARE ANY INCORRECT BITS SET IN THE RESULT?
```

```
004162  001365                                  BNE     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN

                                                ;STORE NUMBER AT SPECIFIED ADDRESS

004164  013704  004216                          MOV     DEVADR,R4       ;POINT TO THE LOCATION WHERE THE RESULT WILL BE STORED
004170  010524                          1$:     MOV     R5,(R4)+        ;STORE THE RESULT
004172  062705  000002                          ADD     #2,R5           ;CALCULATE THE NEXT DATUM
004176  105337  004221                          DECB    ADRCNT          ;REDUCE COUNT OF STORED RESULTS. IS IT EXCEEDED?
004202  001372                                  BNE     1$              ;IF NOT, GO STORE THE NEXT DATUM
004204  012604                                  MOV     (SP)+,R4        ;RESTORE R4
004206  012605                                  MOV     (SP)+,R5        ;RESTORE R5
004210  000002                                  RTI                     ;RETURN TO THE MAIN PROGRAM

004212  000000                          LOLIM:  0                       ;LOWEST ACCEPTABLE VALUE
004214  000000                          HILIM:  0                       ;HIGHEST ACCEPTABLE
004216  000000                          DEVADR: 0                       ;LOCATION WHERE RESULT WILL BE STORED
004220     000                          LOBITS: .BYTE   0               ;INCORRECT BITS MASK
004221     000                          ADRCNT: .BYTE   0               ;COUNT OF ITEMS TO BE STORED

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

004222  016637  000004  001404          .SAV05: MOV     4(SP),SAVPC     ;SAVE R7 (PC)

                                                ;SAVE R0-R5

004230  010537  001340                  SV05:   MOV     R5,$REG5        ;SAVE R5
004234  010437  001336                          MOV     R4,$REG4        ;SAVE R4
004240  010337  001334                          MOV     R3,$REG3        ;SAVE R3
004244  010237  001332                          MOV     R2,$REG2        ;SAVE R2
004250  010137  001330                          MOV     R1,$REG1        ;SAVE R1
004254  010037  001326                          MOV     R0,$REG0        ;SAVE R0
004260  000002                                  RTI                     ;LEAVE.

                                                ;RESTORE R0-R5

004262  013700  001326                  .RES05: MOV     $REG0,R0        ;RESTORE R0
004266  013701  001330                          MOV     $REG1,R1        ;RESTORE R1
004272  013702  001332                          MOV     $REG2,R2        ;RESTORE R2
004276  013703  001334                          MOV     $REG3,R3        ;RESTORE R3
004302  013704  001336                          MOV     $REG4,R4        ;RESTORE R4
004306  013705  001340                          MOV     $REG5,R5        ;RESTORE R5
004312  000002                                  RTI                     ;LEAVE

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

004314  104402  001357                  .CONVR: TYPE    ,$CRLF          ;PRINT A CARRIAGE RETURN
004320  010046                          .CNVRT: MOV     R0,-(SP)        ;SAVE R0
004322  010146                                  MOV     R1,-(SP)        ;SAVE R1
004324  010346                                  MOV     R3,-(SP)        ;SAVE R3
004326  010446                                  MOV     R4,-(SP)        ;SAVE R4
004330  010546                                  MOV     R5,-(SP)        ;SAVE R5
004332  017601  000012                          MOV     @12(SP),R1      ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
004336  062766  000002  000012                  ADD     #2,12(SP)       ;POINT TO WHERE MAIN PROGRAM WILL RESUME
004344  012137  004470                          MOV     (R1)+,WRDCNT    ;GET NUMBER OF WORDS TO BE PRINTED
```

```
004350  112105                  1$:      MOVB    (R1)+,R5           ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
004352  112100                           MOVB    (R1)+,R0           ;GET THE NUMBER OF SPACES TO PRINT
004354  013104                           MOV     @(R1)+,R4          ;COPY THE WORD TO BE CONVERTED
004356  110537  004472                   MOVB    R5,CHRCNT          ;COPY THE CHARACTER COUNT
004362  010403                  3$:      MOV     R4,R3              ;COPY THE ARGUMENT WORD AGAIN
004364  042703  177770                   BIC     #^C<7>,R3          ;ISOLATE THREE BITS TO BE TREATED AS A CHARACTER
004370  062703  000060                   ADD     #060,R3            ;MAKE AN ASCII CHARACTER OUT OF THEM
004374  110346                           MOVB    R3,-(SP)           ;SAVE THAT CHARACTER
004376  006004                           ROR     R4                 ;MOVE THE NEXT THREE BITS INTO PLACE
004400  006204                           ASR     R4                 ;MOVE THEM AGAIN
004402  006204                           ASR     R4                 ;AND FINALLY A THIRD TIME
004404  005305                           DEC     R5                 ;REDUCE CHARACTER COUNT.ARE ALL CHARACTERS
                                                                    ;BUILT?
004406  001365                           BNE     3$                 ;IF NO, GO BUILD THE NEXT ONE.
004410  012703  007616                   MOV     #MDATA,R3          ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
004414  112623                  4$:      MOVB    (SP)+,(R3)+        ;STORE THE CHARACTER, STARTING WITH THE MOST
004416  105337  004472                   DECB    CHRCNT             ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
004422  001374                           BNE     4$                 ;IF NO, GO TRANSFER ANOTHER
004424  105700                           TSTB    R0                 ;ARE ANY SPACES TO BE PRINTED?
004426  001404                           BEQ     6$                 ;IF NO, DON'T SET UP ANY
004430  112723  000040          5$:      MOVB    #040,(R3)+         ;ADD A SPACE TO THE OUTPUT BUFFER
004434  105300                           DECB    R0                 ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
004436  001374                           BNE     5$                 ;IF YES, GO ADD ANOTHER SPACE
004440  105013                  6$:      CLRB    (R3)               ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
004442  104402  007616                   TYPE    ,MDATA             ;PRINT THE STRING WE JUST BUILT
004446  005337  004470                   DEC     WRDCNT             ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
004452  001336                           BNE     1$                 ;IF YES, GO CONVERT THEM
004454  012605                           MOV     (SP)+,R5           ;RESTORE R5
004456  012604                           MOV     (SP)+,R4           ;RESTORE R4
004460  012603                           MOV     (SP)+,R3           ;RESTORE R3
004462  012601                           MOV     (SP)+,R1           ;RESTORE R1
004464  012600                           MOV     (SP)+,R0           ;RESTORE R0
004466  000002                           RTI                        ;RETURN TO THE MAIN PROGRAM
004470  000000                  WRDCNT:  0
004472     000                  CHRCNT:  .BYTE                      ;NUMBER OF CHARACTERS TO PRINT
004473     000                  SPACNT:  .BYTE   0                  ;NUMBER OF SPACES TO PRINT

004474  000000                  BINWRD:  0



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

004476  010046                  .TRPSR:  MOV     R0,-(SP)           ;SAVE R0. USE R0 TO FIND TRAP ROUTINE
004500  016600  000002                   MOV     2(SP),R0           ;GET TRAP ADDRESS
004504  005740                           TST     -(R0)              ;GET TRAP
004506  111000                           MOVB    (R0),R0            ;GET RIGHT BYTE OF TRAP(TRAP OFFSET)
004510  006300                           ASL     R0                 ;POSITION OFFSET FOR TABLE INDEXING
004512  016000  001742                   MOV     .TRPTAB(R0),R0     ;PLACE INDEXED ADDRESS OF TABLE IN R0
004516  000200                           RTS     R0                 ;TRANSFER TO THAT ADDRESS AND RESTORE OLD R0

                                         ;DEVICE CLEAR ROUTINE
                                         ;ISSUE A DEVICE CLEAR
                                         ;-----------------------
```

```
004520                                   .DEVICE.CLR:
004520  052777  000020  175262                   BIS     #DCLR,@DZVCSR       ;SET DCLR
004526  032777  000020  175254           1$:     BIT     #DCLR,@DZVCSR       ;DID IT CLEAR?
004534  001374                                   BNE     1$                  ;BR IF NO
004536  000002                                   RTI                         ;EXIT ROUTINE

                                                 ;ROUTINE TO HANDLE MAINTENANCE BIT SETTING WITH DEVICE CLEAR
                                                 ;-----------------------------------------------------------
004540  104413                           .DCLASM:DEVICE.CLR                  ;ISSUE A DEVICE CLEAR
004542  153777  001424  175240                   BISB    MNTFLG,@DZVCSR      ;LOAD THE MAINTENANCE BIT IF IT IS I MODE
004550  000002                                   RTI                         ;RETURN TO CALLING ROUTINE

004552                                   .DELAY:
004552  010046                                   MOV     R0,-(SP)            ;SAVE R0
004554  013700  004570                           MOV     DLYCNT,R0           ;SET COUNT
004560  005300                           1$:     DEC     R0                  ;DELAY
004562  001376                                   BNE     1$                  ;
004564  012600                                   MOV     (SP)+,R0            ;RESTORE R0
004566  000002                                   RTI                         ;LEAVE ROUTINE
004570  000001                           DLYCNT: .WORD     1                 ;PATCHABLE LOC FOR MORE TIME

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

004572  013716  001362                   .ADVANCE:MOV    NEXT,(SP)           ;CRUNCH STACK WITH ADDRESS OF NEXT TEST
004576  005037  001364                           CLR     LOCK                ;RESET TIGHT LOOP ADDRESS
004602  000002                                   RTI                         ;CHECK TO SEE IF OLD TEST GETS REPEATED

                                                 ;ROUTINE TO SHIFT LINE POINTER
                                                 ;AND SWITCH TESTS IF NECESSARY
                                                 ;----------------------------
004604  106302                           .SHIFT: ASLB    R2                  ;POINT TO THE NEXT LINE
004606  032702  000020                           BIT     #BIT4,R2            ;HAVE WE PASSED ALL LINE POINTERS?
004612  001402                                   BEQ     1$                  ;IF NOT, RETURN TO THE TEST
004614  022626                                   POP2SP                      ;REMOVE THE TRAP CALL FROM THE STACK
004616  104400                                   ADVANCE                     ;GO TO THE NEXT TEST
004620  000002                           1$:     RTI                         ;RETURN TO THE PRESENT TEST

```

```
                                      ;LINE PARAMETER REGISTER SETUP ROUTINE

004622  010146                      .LPRSET:MOV     R1,-(SP)        ;SAVE CONTENTS OF R1
004624  010246                              MOV     R2,-(SP)        ;SAVE CONTENTS OF R2
004626  013701  001370                      MOV     PAR,R1          ;MOVE DEFAULT PARAM. INTO R1
004632  012702  000001                      MOV     #1,R2   ;INIT. FOR LINE 1
004636  010177  175156              1$:     MOV     R1,@DZVLPR      ;LOAD PARAM. REGISTER
004642  005201                              INC     R1              ;SET R1 FOR NEXT LINE
004644  106302                              ASLB    R2              ;SET R2 FOR NEXT LINE
004646  032702  000020                      BIT     #BIT4,R2        ;ALL LINES DONE?
004652  001771                              BEQ     1$              ;IF NO LOAD NEXT LINE
004654  012602                              MOV     (SP)+,R2        ;RELOAD R2
004656  012601                              MOV     (SP)+,R1        ;RELOAD R1
004660  000002                              RTI                     ;RETURN

                                      ;ROUTINE TO ZERO DATA BUFFER

004662  010046                      .BUFSET:MOV     R0,-(SP)        ;SAVE CONTENTS OF R0
004664  012700  001426                      MOV     #TDO,R0         ;SET R0 TO TOP OF BUFFER
004670  005020                      1$:     CLR     (R0)+           ;CLEAR BUFFER LOCATION
004672  022700  001446                      CMP     #STOP,R0        ;IS BUFFER ALL CLEARED
004676  001374                              BNE     1$              ;IF NOT CLEAR NEXT LOCATION
004700  012600                              MOV     (SP)+,R0        ;RELOAD R0
004702  000002                              RTI                     ;RETURN

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

004704  004737  005332              $ERROR: JSR     PC,SERV.G       ;FIND OUT IF <↑G> WAS HIT
004710  032777  010000  174366              BIT     #SW12,@SWR      ;BELL ON ERROR?
004716  001406                              BEQ     XBX             ;BR IF NO BELL
004720  105777  174370                      TSTB    @$TPS           ;TTY READY.
004724  100003                              BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
004726  112777  000207  174362              MOVB    #207,@$TPB      ;PUSH A BELL AT THE TTY.
004734  032777  020000  174342      XBX:    BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
004742  001113                              BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
004744  021637  001262                      CMP     (SP),$ERRPC     ;WAS THIS ERROR FOUND LAST TIME?
004750  001404                              BEQ     1$              ;BR IF YES
004752  011637  001262                      MOV     (SP),$ERRPC     ;RECORD BEING HERE
004756  105037  001247                      CLRB    $ERFLG          ;PREPARE HEADER
004762  104407                      1$:     SAV05                   ;SAVE ALL PROC REGISTERS
004764  011605                              MOV     (SP),R5         ;GET THE PC OF ERROR
004766  162705  000002                      SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
004772  011504                              MOV     (R5),R4         ;GET ERROR INSTRUCTION
004774  110437  001260                      MOVB    R4,$ITEMB       ;COPY TEST NUMBER FOR APT HANDLING
005000  006304                              ASL     R4              ;MULT BY TWO
005002  061504                              ADD     (R5),R4         ;DOUBLE IT
005004  006304                              ASL     R4              ;MULT AGAIN
005006  042704  177001                      BIC     #177001,R4      ;CLEAR JUNK
005012  062704  011064                      ADD     #.ERRTAB,R4     ;GET POINTER
005016  012437  005142                      MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
005022  012437  005154                      MOV     (R4)+,DATAHD    ;GET DATA HEADRER
005026  011437  005166                      MOV     (R4),DATABP     ;GET DATA TABLE
005032  105737  001247                      TSTB    $ERFLG          ;TYPE HEADER
005036  001403                              BEQ     TYPMSG          ;BR IF YES
005040  005737  005166                      TST     DATABP          ;DOES DATA TABLE EXIST?
```

```
005044  001044                          BNE     TYPDAT          ;BR IF YES.
005046  104402  001357          TYPMSG: TYPE    ,$CRLF          ;TYPE A CARRIAGE RETURN
005052  104402  001357                  TYPE    ,$CRLF          ;AND TYPE ANOTHER
005056  005737  001364                  TST     LOCK
005062  001402                          BEQ     1$
005064  104402  006245                  TYPE    ,MASTEK
005070  104402  006233          1$:     TYPE    ,MTSTN
005074  104412  005324                  CNVRT   ,XTSTN          ;SHOW IT
005100  104402  006323                  TYPE    ,MERRPC         ;TYPE PC.
005104  104412  005316                  CNVRT   ,ERTABO         ;SHOW IT
005110  104402  006175                  TYPE    ,MCSRX
005114  104412  002660                  CNVRT   ,XCSR
005120  104402  001357                  TYPE    ,$CRLF          ;GIVE A CR/LF
005124  112737  177777  001247          MOVB    #-1,$ERFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
005132  005737  005142                  TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
005136  001402                          BEQ     WTBS.FM         ;BR IF NO.
005140  104402                          TYPE                    ;TYPE
005142  000000                  ERRMSG: 0                       ;    ERROR MESSAGE
005144                          WTBS.FM:                        ;
005144  005737  005154                  TST     DATAHD          ;DATA HEADER?
005150  001402                          BEQ     TYPDAT          ;BR IF NO
005152  104402                          TYPE                    ;TYPE
005154  000000                  DATAHD: 0                       ;    DATA HEADER
005156  005737  005166          TYPDAT: TST     DATABP          ;DATA TABLE?
005162  001402                          BEQ     RESREG          ;BR IF NO.
005164  104411                          CONVRT                  ;SHOW
005166  000000                  DATABP: 0                       ;    DATA TABLE
005170  104410                  RESREG: RESOS                   ;RESTORE PROC REGISTERS
005172  122737  000001  001140  HALTS:  CMPB    #APTENV,$ENV    ;IS APT RUNNING?
005200  001007                          BNE     15$             ;SKIP APT CALL IF NOT
005202  113737  001260  005214          MOVB    $ITEMB,5$       ;COPY ERROR NUMBER
005210  004737  003446                  JSR     PC,$ATY4        ;CALL APT SERVICE
005214  000000                  5$:     .WORD   0               ;ERROR NUMBER STUCK HERE
005216  000777                  10$:    BR      10$             ;LOCK UP HERE
005220  022737  002644  000042  15$:    CMP     #$ENDAD,@#42    ;CHECK TO SEE IF IN ACT-11 MODE
005226  001403                          BEQ     20$             ;IF SO, HANDLE ACCORDINGLY
005230  005777  174050                  TST     @SWR            ;HALT ON ERROR?
005234  100004                          BPL     EXITER          ;BR IF NO HALT ON ERROR
005236  016677  000002  174042  20$:    MOV     2(SP),@DISPLAY  ;SHOW ERROR PC IN DATA DISPLAY
005244  000000                          HALT                    ;HALT
005246  005237  001256          EXITER: INC     $ERTTL          ;UPDATE ERROR COUNT
005252  004737  005332                  JSR     PC,SERV.G       ;FIND OUT IF ↑G WAS TYPED
005256  032777  000400  174020          BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
005264  001007                          BNE     1$              ;BR IF YES
005266  032777  002000  174010          BIT     #SW10,@SWR      ;GOTO NEXT TEST?
005274  001407                          BEQ     2$              ;BR IF NO
005276  013737  001362  001252          MOV     NEXT,$LPADR     ;SET FOR NEXT TEST
005304  012706  001120          1$:     MOV     #STACK,SP       ;RESET SP
005310  000177  173736                  JMP     @$LPADR         ;GOTO SPECIFIED TEST
005314  000002                  2$:     RTI                     ;RETURN
005316  000001                  ERTABO: 1
005320     006     002                  .BYTE   6,2
005322  001404                          SAVPC
005324  000001                  XTSTN:  1
005326     002     002                  .BYTE   2,2
005330  001246                          $TSTNM
```

```
005332  017746  173754          SERV.G: MOV     @$TKB,-(SP)      ;OTHERWISE, GET THE LAST CHARACTER TYPED
005336  042716  000200                  BIC     #BIT7,(SP)       ;STRIP PARITY(EIGHTH) BIT
005342  122726  000007                  CMPB    #7,(SP)+         ;IS IT ↑G?
005346  001076                          BNE     6$               ;IF NOT, IGNORE INPUT
005350  032777  004000  173732          BIT     #4000,@$TKS      ;RX BUSY?
005356  001365                          BNE     SERV.G           ;BR IF YES
005360  017737  173720  005566          MOV     @SWR,90$         ;SAVE (SWR).
005366  104402  005546          1$:     TYPE    ,89$             ;TYPE HEADER FOR OLD SWITCH REGISTER
005372  104412  005560                  CNVRT   ,88$             ;TYPE THE NUMBER ITSELF
005376  104402  005570                  TYPE    ,91$             ;AFTER HAVING CONVERTED IT TO ASCII
005402  105037  005574                  CLRB    92$              ;CLEAR SWR CHANGE FLAG
005406  005077  173672                  CLR     @SWR             ;CLEAR THE SOFTWARE SWITCH REGISTER
005412  105777  173672          3$:     TSTB    @$TKS            ;WAIT FOR DONE.
005416  100375                          BPL     3$               ;CONTINUE WAITING FOR IT
005420  017746  173666                  MOV     @$TKB,-(SP)      ;PUT THE CHARACTER ON THE STACK
005424  042716  000200                  BIC     #BIT7,(SP)       ;STRIP PARITY BIT
005430  122726  000015                  CMPB    #15,(SP)+        ;IS IT THE CARRIAGE RETURN CHAR?
005434  001433                          BEQ     4$               ;IF SO, GO PRINT CRLF
005436  105777  173652          2$:     TSTB    @$TPS            ;IS THE OUTPUT BUFFER AVAILABLE
005442  100375                          BPL     2$               ;IF NOT, WAIT FOR IT TO BE READY
005444  105237  005574                  INCB    92$              ;INDICATE THAT THE SWR WAS CHANGED
005450  014677  173642                  MOV     -(SP),@$TPB      ;PLACE THE CHARACTER THERE(ECHO BACK)
005454  000241                          CLC                      ;GET READY TO ROTATE
005456  006177  173622                  ROL     @SWR             ;MOVE THE EXISTING BITS OVER
005462  006177  173616                  ROL     @SWR             ;TO MAKE ROOM FOR THE INCOMING
005466  006177  173612                  ROL     @SWR             ;THREE BITS FROM THIS CHARACTER
005472  103735                          BCS     1$               ;ERROR
005474  022627  000060                  CMP     (SP)+,#60        ;IS IT LOWER THAN 0?
005500  002732                          BLT     1$               ;IF SO, GO ASK AGAIN
005502  026627  177776  000067          CMP     -2(SP),#67       ;IS IT HIGHER THAN 7?
005510  003326                          BGT     1$               ;IF SO, GO ASK AGAIN
005512  042746  177770                  BIC     #↑C<7>,-(SP)     ;ISOLATE INFORMATION BITS
005516  052677  173562                  BIS     (SP)+,@SWR       ;ADD THEM TO THE SWITCH REGISTER
005522  000733                          BR      3$               ;GO CHECK FOR THE NEXT CHARACTER
005524  105737  005574          4$:     TSTB    92$              ;HAS THE SWR BEEN CHANGED?
005530  001003                          BNE     5$               ;IF YES GO TYPE CRLF
005532  013777  005566  173544          MOV     90$,@SWR         ;IF NOT RESTORE SWR
005540  104402  001357          5$:     TYPE    ,$CRLF           ;TYPE A CARRIAGE RETURN AND LINE FEED
005544  000207                  6$:     RTS     PC               ;RETURN TO CALLING PROCEDURE

005546  020200  051450  051127  89$:    .ASCIZ  <200>? (SWR)=/?
005554  036451  000057
                                .EVEN
005560  000001                  88$:    1
005562     006     000                  .BYTE   6,0
005564  005566                          90$
005566  000000                  90$:    .WORD   0
005570  036457  000057          91$:    .ASCIZ  ?/=/?
005574     000                  92$:    .BYTE   0
        005576                  .EVEN
                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;*****************************************************************
                                ;POWER DOWN ROUTINE
005576  012737  005742  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
005604  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
```

```
     005612  010046                             MOV     R0,-(SP)          ;;PUSH R0 ON STACK
     005614  010146                             MOV     R1,-(SP)          ;;PUSH R1 ON STACK
     005616  010246                             MOV     R2,-(SP)          ;;PUSH R2 ON STACK
     005620  010346                             MOV     R3,-(SP)          ;;PUSH R3 ON STACK
     005622  010446                             MOV     R4,-(SP)          ;;PUSH R4 ON STACK
     005624  010546                             MOV     R5,-(SP)          ;;PUSH R5 ON STACK
     005626  017746  173452                     MOV     @SWR,-(SP)        ;;PUSH @SWR ON STACK
     005632  010637  005746                     MOV     SP,$SAVR6         ;;SAVE SP
     005636  012737  005650  000024             MOV     #$PWRUP,@#PWRVEC  ;;SET UP VECTOR
     005644  000000                             HALT
     005646  000776                             BR      .-2               ;;HANG UP

                                        ;;*************************************************************
                                        ;POWER UP ROUTINE
     005650  012737  005742  000024   $PWRUP:   MOV     #$ILLUP,@#PWRVEC  ;;SET FOR FAST DOWN
     005656  013706  005746                     MOV     $SAVR6,SP         ;;GET SP
     005662  005037  005746                     CLR     $SAVR6            ;;WAIT LOOP FOR THE TTY
     005666  005237  005746           1$:       INC     $SAVR6            ;;WAIT FOR THE INC
     005672  001375                             BNE     1$                ;;OF  WORD
     005674  012677  173404                     MOV     (SP)+,@SWR        ;;POP STACK INTO @SWR
     005700  012605                             MOV     (SP)+,R5          ;;POP STACK INTO R5
     005702  012604                             MOV     (SP)+,R4          ;;POP STACK INTO R4
     005704  012603                             MOV     (SP)+,R3          ;;POP STACK INTO R3
     005706  012602                             MOV     (SP)+,R2          ;;POP STACK INTO R2
     005710  012601                             MOV     (SP)+,R1          ;;POP STACK INTO R1
     005712  012600                             MOV     (SP)+,R0          ;;POP STACK INTO R0
     005714  012737  005576  000024             MOV     #$PWRDN,@#PWRVEC  ;;SET UP THE POWER DOWN VECTOR
     005722  012737  000340  000026             MOV     #340,@#PWRVEC+2   ;;PRIO:7
     005730  104402                             TYPE                      ;;REPORT THE POWER FAILURE
     005732  005750                   $PWRMG:   .WORD   MPFAIL            ;;POWER FAIL MESSAGE POINTER
     005734  012716                             MOV     (PC)+,(SP)        ;;RESTART AT RESTART
     005736  002512                   $PWRAD:   .WORD   RESTART           ;;RESTART ADDRESS
     005740  000002                             RTI
     005742  000000                   $ILLUP:   HALT                      ;;THE POWER UP SEQUENCE WAS STARTED
     005744  000776                             BR      .-2               ;; BEFORE THE POWER DOWN WAS COMPLETE
     005746  000000                   $SAVR6:   0                         ;;PUT THE SP HERE
     005750  050200  051127  043040   MPFAIL:   .ASCIZ  <200>/PWR FAILED. RESTART AT LAST TEST /
 (2) 006013     200  047105  020104   MEPASS:   .ASCIZ  <200>/END PASS DVDZC-A  /
 (2) 006037     200  052522  047116   MR:       .ASCIZ  <200>/RUNNING   /
 (2) 006053     200  051120  043517   MERR2:    .ASCIZ  <200>/PROGRAM INDICATES NO DEVICES PRESENT./
 (2) 006122  044600  051516  043125   MERR3:    .ASCIZ  <200>/INSUFFICIENT DATA!/
 (2) 006146  046200  041517  020113   MLOCK:    .ASCIZ  <200>/LOCK ON SELECTED TEST/
 (2) 006175     103  051123  020072   MCSRX:    .ASCIZ   /CSR: /
 (2) 006203     126  041505  020072   MVECX:    .ASCIZ  /VEC: /
 (2) 006211     120  051501  042523   MPASSX:   .ASCIZ  /PASSES: /
 (2) 006222  051105  047522  051522   MERRX:    .ASCIZ  /ERRORS: /
 (2) 006233     124  051505  020124   MTSTN:    .ASCIZ  /TEST NO: /
 (2) 006245     052  000040           MASTEK:   .ASCIZ  /* /
 (2) 006250  051600  052105  051440   MNEW:     .ASCIZ  <200>/SET SWITCH REG TO DZV11'S DESIRED ACTIVE./
 (2) 006323     120  035103  000040   MERRPC:   .ASCIZ  /PC: /
 (2) 006330  046600  050101  047440   XHEAD:    .ASCIZ  <200>/MAP OF DZV11 STATUS/<200>
 (2) 006356  044600  046114  043505   MBADLN:   .ASCIZ  <200>/ILLEGAL ENTRY IN STAGGERED MODE/<200>
 (2)                                  .EVEN
 (2) 006420  000002                   XSTATQ:   2
     006422     006     003                     .BYTE   6,3
     006424  001344                             $TMP1
```

```
  1558  006426      006      002                .BYTE   6,2
  1559  006430  001346                          $TMP2
  1560                                  .EVEN
```

```
                                                  ;THIS ROUTINE ESTABLISHES WHICH MAINTENANCE MODE THE DEVICE IS IN
                                                  ;----------------------------------------------------------
                                                  ;E=EXTERNAL LOOP BACK
                                                  ;I=INTERNAL LOOP BACK
                                                  ;S=STAGGERED LOOP BACK
006432  017605  000000          .SETFLG:MOV     @(SP),R5                ;PICK UP ADDRESS OF TAG
006436  042737  000040  007512          BIC     #40,INBUF               ;STRIP LOWER CASE
006444  122737  000105  007512          CMPB    #'E,INBUF               ;IS IT EXTERNAL LOOP BACK ?
006452  001005                          BNE     4$                      ;NO
006454  013715  006544                  MOV     1$,(R5)                 ;YES STORE INFO
006460  105037  001424                  CLRB    MNTFLG                  ;SET MAINT BIT =0
006464  000422                          BR      7$                      ;GET OUT
006466  122737  000111  007512  4$:     CMPB    #'I,INBUF               ;IS IT INTERNAL LOOP BACK ?
006474  001006                          BNE     5$                      ;NO
006476  013715  006546                  MOV     2$,(R5)                 ;YES STORE INFO
006502  112737  000010  001424          MOVB    #MAINT,MNTFLG           ;SET UP THE MAINTENANCE FLAG LOADER
006510  000410                          BR      7$                      ;GET OUT
006512  122737  000123  007512  5$:     CMPB    #'S,INBUF               ;IS IT STAGGERED LOOP BACK ?
006520  001007                          BNE     6$                      ;WHAT ?
006522  013715  006550                  MOV     3$,(R5)                 ;YES STORE INFO
006526  105037  001424                  CLRB    MNTFLG                  ;ZERO BITS
006532  062716  000002          7$:     ADD     #2,(SP)                 ;POP AROUND
006536  000002                          RTI
006540  104404                  6$:     INSTER                          ;RETRY
006542  000733                          BR      .SETFLG                 ;DITTO
006544  000200                  1$:     .WORD   200                     ;EXTERNAL =  E
006546  000000                  2$:     .WORD   0                       ;INTERNAL = I
006550  100000                  3$:     .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

006552  017605  000000          .PAWCH: MOV     @(SP),R5
006556  142737  000040  007512          BICB    #40,INBUF               ;SET FOR LOWER CASE INPUT
006564  122737  000105  007512          CMPB    #'E,INBUF               ;IS IT "E" ?
006572  001002                          BNE     1$
006574  105015                          CLRB    (R5)                    ;000
006576  000406                          BR      2$
006600  122737  000103  007512  1$:     CMPB    #'C,INBUF               ;IS IT "C" ?
006606  001005                          BNE     3$
006610  112715  177777                  MOVB    #-1,(R5)                ;3177
006614  062716  000002          2$:     ADD     #2,(SP)
006620  000002                          RTI
006622  104404                  3$:     INSTER                          ;RETRY
006624  000752                          BR      .PAWCH
```

```
                                        ;THIS ROUTINE CONVERTS LINE SPEED (LINESP) AND
                                        ;LINE NUMBER (SAVLIN) FOR DZVLPR, DZVTCR AND DZVCSR
                                        ;REGISTER USAGE.

006626  013737  001374  007044  SET:    MOV     SAVLIN,NUMLIN   ;SAVE SAVLIN
006634  013700  001374                  MOV     SAVLIN,R0       ;COPY THE LINE NUMBER FOR LOOP CONTROL
006640  005037  007046                  CLR     NUMTCR          ;SET A DEFAULT OF LINE 0 OR NO LINES
006644  012702  000001                  MOV     #1,R2           ;SET A BIT POINTER TO THE FIRST LINE
006650  005300                  XTCR1:  DEC     R0              ;REDUCE THE INDICATOR.IS IT MINUS YET?
006652  100402                          BMI     SET1            ;IF SO, R2 POINTS TO THE RIGHT LINE
006654  006302                          ASL     R2              ;IF NOT, MOVE THE POINTER TO THE NEXT LINE
006656  000774                          BR      XTCR1           ;GO SEE IF THIS LINE IS THE ONE
006660  012701  006722          SET1:   MOV     #TABLE2,R1
006664  010237  007046                  MOV     R2,NUMTCR       ;COPY THE CORRECT BIT POINTER
006670  022137  007040          1$:     CMP     (R1)+,LINESP
006674  001407                          BEQ     2$
006676  005721                          TST     (R1)+           ;IS IT THE END OF TABLE?
006700  001373                          BNE     1$              ;NO
006702  104402  007150                  TYPE    ,MINVAL         ;INVALID BAUD RATE,BEGIN AGAIN
006706  012705  002350                  MOV     #BAUD,R5        ;JUMP TO BAUD THRU R5
006712  000402                          BR      3$
006714  011137  007042          2$:     MOV     (R1),SPEED      ;SET UP BAUD RATE
006720  000205                  3$:     RTS     R5



                                        ;THE FOLLOWING IS A TABLE OF LEGAL BAUD RATES (8 BITS/CHAR)
006722  000062                  TABLE2: .WORD   50.             ;50 BAUD
006724  010070                          .WORD   10070           ;
006726  000113                          .WORD   75.             ;75 BAUD
006730  010470                          .WORD   10470           ;
006732  000156                          .WORD   110.            ;110 BAUD
006734  011070                          .WORD   11070           ;TWO STOP BITS
006736  000207                          .WORD   135.            ;134.5 BAUD
006740  011470                          .WORD   11470           ;TWO STOP BITS
006742  000226                          .WORD   150.            ;150 BAUD
006744  012070                          .WORD   12070           ;TWO STOP BITS
006746  000454                          .WORD   300.            ;300 BAUD
006750  012430                          .WORD   12430           ;ONE STOP BIT
006752  001130                          .WORD   600.            ;600 BAUD
006754  013030                          .WORD   13030           ;ONE STOP BIT
006756  002260                          .WORD   1200.           ;1200 BAUD
006760  013430                          .WORD   13430           ;ONE STOP BIT
006762  003410                          .WORD   1800.           ;1800 BAUD
006764  014030                          .WORD   14030           ;ONE STOP BIT
006766  003720                          .WORD   2000.           ;2000 BAUD
006770  014430                          .WORD   14430           ;ONE STOP BIT
006772  004540                          .WORD   2400.           ;2400 BAUD
006774  015030                          .WORD   15030           ;ONE STOP BIT
006776  007020                          .WORD   3600.           ;3600 BAUD
007000  015430                          .WORD   15430           ;ONE STOP BIT
007002  011300                          .WORD   4800.           ;4800 BAUD
007004  016030                          .WORD   16030           ;ONE STOP BIT
007006  016040                          .WORD   7200.           ;7200 BAUD
007010  016430                          .WORD   16430           ;ONE STOP BIT
007012  022600                          .WORD   9600.           ;9600 BAUD
```

```
  1664  007014  017030                          .WORD   17030           ;
  1665  007016  177777  000000                  .WORD   -1,0            ;TABLE TERMINATOR
  1666
  1667
  1668  007022  000000                  WCHFLG:0                        ;ECHO OR CABLE FLAG
  1669  007024  000000                  STFLG: 0                        ;PROGRAM START FLAG
  1670  007026  000000                  LOCKUP: 0                       ;TIMEOUT FLAG
  1671  007030  000000                  LAST:   0                       ;LAST ERROR PC
  1672  007032  000000                  TDATA:  0
  1673  007034  000000                  RDATA:  0
  1674  007036  000000                  BYTCNT: 0
  1675  007040  000156                  LINESP: 110.                    ;DEFAULT BAUD RATE
  1676  007042  011070                  SPEED:  11070                   ;DEFAULT 110 BAUD, 8 BITS/CHAR,
  1677                                                                  ;FDX, 2 STOP BITS
  1678  007044  000000                  NUMLIN: 0
  1679
  1680  007046  000001                  NUMTCR: 1                       ;DEFAULT VALUE,TCR BIT 0
  1681  007050  000200                  PRIO:   200                     ;DEFAULT DEVICE PRIORITY
  1682                                                                  ;MASK OUT INTERRUPTS
  1683  007052  000C00                  RECDAT: 0
  1684  007054  000000                  TBUF:   0
  1685  007056  053200  041505  047524  MVECTO: .ASCIZ  <200>/VECTOR ADDRESS- /
   (2)  007100  041600  047117  051124  MREGAD: .ASCIZ  <200>/CONTROL REGISTER ADDRESS- /
   (2)  007134  050200  051501  020123  MPASS:  .ASCIZ  <200>/PASS DONE./
   (2)  007150  044600  053116  046101  MINVAL: .ASCIZ  <200>/INVALID BAUD RATE - /
   (2)  007176  046200  047111  035105  MLINE:  .ASCIZ  <200>/LINE: /
   (2)  007206  041200  052501  020104  MSPEED: .ASCIZ  <200>/BAUD RATE - /
   (2)  007224  052200  050131  020105  MCHAR:  .ASCIZ  <200>/TYPE A CHAR. ON DZV11 TERMINAL /
   (2)  007265     200  044127  041511  MWHICH: .ASCIZ  <200>/WHICH TEST ? ECHO OR CABLE (E OR C) /
   (2)  007333     200  042524  046522  MTERM:  .ASCIZ  <200>/TERMINAL ECHO TEST /
   (2)  007360  041600  041101  042514  MCABLE: .ASCIZ  <200>/CABLE TEST /
   (2)  007375     377  177415  005377  MQUICK: .ASCII  <377><15><377><377><12><377><377>
   (2)  007404  044124  020105  052521          .ASCII  /THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 0123456789/
   (2)  007501     377  177415  005377          .ASCII  <377><15><377><377><12><377><377><377><0>
   (2)                                  .EVEN
   (2)
  1686                                          ;BUFFERS FOR INPUT-OUTPUT
  1687
  1688  007512  000000                  INBUF:  0
  1689          007554                  .=.+40
  1690  007554  000000                  TEMP:   0
  1691          007616                  .=.+40
  1692  007616  000000                  MDATA:  0
  1693          007660                  .=.+40
  1694
```

```
                                  ;THIS UTILITY SETS UP CSR'S,SETS UP VECTORS.
007660  013700  002040            DZVLEV: MOV     DZVRIV,R0       ;PLACE THE BASE VECTOR ADDRESS IN R0
007664  062700  000002                    ADD     #2,R0           ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
007670  010037  002042                    MOV     R0,DZVRIS       ;STORE IT HERE
007674  062700  000002                    ADD     #2,R0           ;CALCULATE THE TRANSMITTER INTERRUPT VECTOR
007700  010037  002044                    MOV     R0,DZVTIV       ;STORE IT HERE
007704  062700  000002                    ADD     #2,R0           ;CALCULATE THE TRANSMITTER VECTOR STATUS ADDRESS
007710  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
007714  013700  001174                    MOV     $BASE,R0        ;COPY THE ADDRESS BEING LOADED
007720  010037  002010                    MOV     R0,DZVCSR       ;XXX0
007724  005200                            INC     R0
007726  010037  002012                    MOV     R0,HDZVCSR      ;XXX1
007732  005200                            INC     R0
007734  010037  002014                    MOV     R0,DZVRBUF      ;XXX2
007740  010037  002020                    MOV     R0,DZVLPR       ;XXX2
007744  005200                            INC     R0
007746  010037  002016                    MOV     R0,HDZVRBUF     ;XXX3
007752  010037  002022                    MOV     R0,HDZVLPR      ;XXX3
007756  005200                            INC     R0
007760  010037  002024                    MOV     R0,DZVTCR       ;XXX4
007764  005200                            INC     R0
007766  010037  002026                    MOV     R0,HDZVTCR      ;XXX5
007772  005200                            INC     R0
007774  010037  002030                    MOV     R0,DZVMSR       ;XXX6
010000  010037  002034                    MOV     R0,DZVTDR       ;XXX6
010004  005200                            INC     R0
010006  010037  002032                    MOV     R0,HDZVMSR      ;XXX7
010012  010037  002036                    MOV     R0,HDZVTDR      ;XXX7
010016  000207                            RTS     PC
```

```

                                                ;**********   ECHO TEST   **********
                                                ;*THIS TEST WILL ACCEPT 1 CHARACTER AT A TIME
                                                ;*(IN INTERRUPT MODE) AND TRANSMIT THAT SAME CHARACTER,
                                                ;*ONE LINE AT A TIME, ANY LINE 0 THRU 7 (OCTAL)

 010020  104413                       TST1:     DEVICE.CLR                     ;CLEAR DZV11
 010022  012737  000001  001246                 MOV     #1,$TSTNM
 010030  013777  007046  171766                 MOV     NUMTCR,@DZVTCR         ;SET TCR BIT
 010036  013737  007044  001370                 MOV     NUMLIN,PAR             ;SET PARAMETERS
 010044  053737  007042  001370                 BIS     SPEED,PAR              ;SET BAUD RATE
 010052  013777  001370  171740                 MOV     PAR,@DZVLPR            ;LOAD PARAM.
 010060  012777  000040  171722                 MOV     #MSENAB,@DZVCSR        ;SET SCANN ENABLE
 010066  005004                                 CLR     R4
 010070  012705  007375               4$:       MOV     #MQUICK,R5             ;SET MESSAGE BUFFER
 010074  005777  171710               3$:       TST     @DZVCSR                ;TRDY?
 010100  100404                                 BMI     2$                     ;BR IF YES
 010102  104414                                 DELAY                          ;WAIT
 010104  005304                                 DEC     R4                     ;
 010106  001372                                 BNE     3$                     ;
 010110  104003                                 ERROR   3                      ;NO TRDY SET! WHY?
 010112  005004                       2$:       CLR     R4                     ;RESET COUNTER TO 0
 010114  112577  171714                         MOVB    (R5)+,@DZVTDR          ;LOAD CHAR
 010120  001365                                 BNE     3$
 010122  004737  005332                         JSR     PC,SERV.G              ;<↑G>?
 010126  122777  000377  171150                 CMPB    #377,@SWR              ;SWR SET TO 377?
 010134  001755                                 BEQ     4$                     ;IF YES LOOP ON QUICK MESSAGE
 010136  012737  002516  001362                 MOV     #$EOP,NEXT
 010144  012777  010220  171666                 MOV     #INTSVC,@DZVRIV        ;SET UP INTERRUPT SERVICE
 010152  012777  000200  171662                 MOV     #MASK,@DZVRIS          ;AND LEVEL
 010160  106427  000000                         MTPS    #CLEAR                 ;ALLOW INTERRUPTS
 010164  012777  000140  171616                 MOV     #RIE!MSENAB,@DZVCSR    ;SET RECEIVER INTERRUPT ENABLE
 010172  104402  007224                         TYPE    ,MCHAR                 ;TYPE "ANY CHARACTER"
 010176  105777  171106               1$:       TSTB    @$TKS                  ;IF SOMBODY HITS A KEY- GET NEW LINE #
 010202  100375                                 BPL     1$                     ;LOOP HERE
 010204  106427  000200                         MTPS    #MASK                  ;MASK FURTHER INTERRUPTS
 010210  004737  005332                         JSR     PC,SERV.G              ;MAKE SURE IT WASN'T <↑G>
 010214  000137  002366                         JMP     LINEX                  ;



                                                ;THE FOLLOWING IS THE RECEIVER INTERRUPT SVC ROUTINE
 010220  105777  171564               INTSVC:   TSTB    @DZVCSR                ;TEST REC. FLAG
 010224  100401                                 BMI     .+4
 010226  104004                                 ERROR   4                      ;ERROR - INTERRUPT NOT CAUSED BY FLAG
 010230  017737  171560  007052                 MOV     @DZVRBUF,RECDAT
 010236  100401                                 BMI     .+4
 010240  104023                                 ERROR   23                     ;NON- VALID CHARACTER
 010242  032737  020000  007052                 BIT     #BIT13,RECDAT          ;CHECK FOR FRAMING ERROR
 010250  001401                                 BEQ     .+4                    ;BR IF NO ERROR
 010252  104025                                 ERROR   25                     ;EITHER SOMBODY HIT THE
                                                                               ;"BREAK KEY" OR YOU HAVE AN ERROR!
 010254  113737  007052  007054                 MOVB    RECDAT,TBUF            ;MOVE CHARACTER TO OUTPUT AREA
 010262  113737  007052  007512                 MOVB    RECDAT,INBUF           ;MOVE CHARACTER TO CHECK FOR ↑C
 010270  042737  177600  007512                 BIC     #↑C<177>,INBUF         ;STRIP JUNK PLUS PARITY
 010276  042737  176377  007052                 BIC     #176377,RECDAT         ;SAVE ONLY LINE  NUMBER
 010304  000337  007052                         SWAB    RECDAT
```

```
010310  023737  001374  007052          CMP     SAVLIN,RECDAT    ;DOES THE LINE # COMPARE?
010316  001407                          BEQ     2$
010320  013737  007052  001374          MOV     RECDAT,SAVLIN    ;ADJUST LINE NO. FOR ERROR
010326  104015                          ERROR   15               ;*WRONG LINE NUMBER
010330  013737  007044  001374          MOV     NUMLIN,SAVLIN    ;CORRECT LINE NO. INDICATOR
010336  123727  007512  000003  2$:     CMPB    INBUF,#3                 ;IS IT A ↑C ?
010344  001004                          BNE     1$               ;NO
010346  104413                          DEVICE.CLR
010350  012716  002516                  MOV     #$EOP,(SP)       ;CRUNCH STACK
010354  000002                          RTI
010356  005777  171426          1$:     TST     @DZVCSR          ;TRDY SET
010362  100375                          BPL     1$               ;IF NOT THEN WAIT
010364  113777  007054  171442          MOVB    TBUF,@DZVTDR     ;TRANSMIT THE CHARACTER
010372  000002                          RTI


                                        ;**********   CABLE TEST    *********
                                        ;*THIS TEST TRANSMITS A BINARY COUNT PATTERN
                                        ;*VIA INTERRUPT MODE TO THE RECEIVER
                                        ;*...THE LINE UNDER TEST MUST BE TERMINATED WITH THE TEST CONNECTOR

010374  106427  000200          TST2:   MTPS    #MASK            ;DISABLE INTERRUPTS
010400  012737  000002  001246          MOV     #2,$TSTNM
010406  012737  002516  001362          MOV     #$EOP,NEXT
010414  104413                          DEVICE.CLR               ;
                                        ;*TEST TO VERIFY THAT SETTING DTR FOR A GIVEN LINE
                                        ;*WILL BRING UP "CO" AND "RING" FOR THE SAME LINE
                                        ;*JUMPERS W1,W2,W3 AND W4 MUST BE INSTALLED ON THE
                                        ;*INTERFACE MODULE OTHERWISE AN ERROR REPORT WILL RESULT.
010416  012737  010424  001364          MOV     #1$,LOCK         ;LOOP
010424  113777  007046  171374  1$:     MOVB    NUMTCR,@HDZVTCR  ;SET DTR
010432  005005                          CLR     R5               ;
010434  153705  007046                  BISB    NUMTCR,R5        ;BUILD EXPECTED
010440  000305                          SWAB    R5               ;PUT IN HIGH BYTE
010442  153705  007046                  BISB    NUMTCR,R5        ;
010446  104414                          DELAY                    ;WAIT FOR CABLE DELAY
010450  017704  171354                  MOV     @DZVMSR,R4       ;READY MODEM BITS
010454  020504                          CMP     R5,R4            ;ARE THEY OK?
010456  001401                          BEQ     2$               ;BR IF YES
010460  104022                          ERROR   22               ;IS THE TEST CONNECTOR ON?
                                                                 ;HAS RIGHT LINE BEEN SELECTED?
                                                                 ;IF SO- YOU HAVE A PROBLEM!
                                                                 ;MODEM BITS NOT RIGHT
010462  104401                  2$:     SCOP1                    ;LOOP
010464  104413                  3$:     DEVICE.CLR               ;INIT DZV11
010466  005037  001364                  CLR     LOCK             ;CLEAR SCOP1 LOCK ADDRESS
010472  013737  007042  001370          MOV     SPEED,PAR        ;SET LINE SPEED
010500  053737  007044  001370          BIS     NUMLIN,PAR       ;SELECT LINE #
010506  052737  010000  001370          BIS     #RCVON,PAR       ;ENABLE THE RECEIVER FOR THIS LINE
010514  013777  001370  171276          MOV     PAR,@DZVLPR      ;SET THE PARAMETERS AND TURN ON RECEIVER
010522  012777  010644  171310          MOV     #INTREC,@DZVRIV  ;SET UP INTR SERVICE
010530  012777  000200  171304          MOV     #MASK,@DZVRIS    ;SET UP LEVEL
010536  012777  011034  171300          MOV     #INTRAN,@DZVTIV  ;SET UP INTR SERVICE
010544  012777  000200  171274          MOV     #MASK,@DZVTIS    ;SET UP LEVEL
010552  012777  040140  171230          MOV     #TIE!RIE!MSENAB,@DZVCSR ;SET TRANSMITTER INTERRUPT ENABLE
010560  105037  001425                  CLRB    DONFLG           ;INIT INTERRUPT DONE INDICATOR
```

```
010564  005001                           CLR     R1                 ;RX DATA POINTER- SET TO 0
010566  005002                           CLR     R2                 ;TX DATA POINTER- SET TO 0
010570  013777  007046  171226           MOV     NUMTCR,@DZVTCR     ;SET UP TCR BIT
010576  106427  000000                   MTPS    #CLEAR             ;ALLOW INTERRUPTS

                                ;YOU RETURN HERE AFTER EVERY RECEIVER INTERRUPT
010602  105777  170502          SPIN:    TSTB    @STKS              ;IF SOMEBODY HITS A KEY- GET A NEW LINE #
010606  100004                           BPL     1$                 ;BRANCH IF NO KEY HIT
010610  004737  005332                   JSR     PC,SERV.G          ;MAKE SURE IT WASN'T <↑G>
010614  000137  002366                   JMP     LINEX              ;SW02=1
010620  105737  001425          1$:      TSTB    DONFLG             ;ARE ALL RECEIVER INTER. DONE
010624  001004                           BNE     QUITS              ;IF YES GET OUT OF TIMING LOOP
010626  005237  007026                   INC     LOCKUP             ;INC TIMEOUT FLAG
010632  001363                           BNE     SPIN               ;IF NOT 0  RETURN SPINNING
010634  104011                           ERROR   11                 ;*RECEIVER FAILED TO INTERRUPT  CHECK CABLE/TERMINATOR
010636  104413                  QUITS:   DEVICE.CLR
010640  000137  002516                   JMP     $EOP               ;CALL FOR END OF PASS
010644  005037  007026          INTREC:  CLR     LOCKUP             ;CLEAR TIMEOUT FLAG
010650  105777  171134                   TSTB    @DZVCSR            ;TEST REC DONE
010654  100401                           BMI     .+4                ;YES
010656  104004                           ERROR   4                  ;*FALSE INTERRUPT
010660  017737  171130  007052           MOV     @DZVRBUF,RECDAT    ;SAVE WORD
010666  100401                           BMI     .+4
010670  104023                           ERROR   23                 ;*NON VALID CHARACTER
010672  032737  040000  007052           BIT     #BIT14,RECDAT      ;DATA OVERRUN ?
010700  001401                           BEQ     .+4                ;NO
010702  104024                           ERROR   24                 ;*YES
010704  032737  020000  007052           BIT     #BIT13,RECDAT      ;FRAMING ERROR ?
010712  001401                           BEQ     .+4                ;NO
010714  104025                           ERROR   25                 ;*YES
010716  032737  010000  007052           BIT     #BIT12,RECDAT      ;PARITY ERROR ?
010724  001401                           BEQ     .+4                ;NO
010726  104026                           ERROR   26                 ;*YES
010730  110105                           MOVB    R1,R5              ;SET EXPECTED
010732  113704  007052                   MOVB    RECDAT,R4          ;GET FOUND
010736  042704  177400                   BIC     #↑C<377>,R4        ;CLEAR HIGH BYTE
010742  042705  177400                   BIC     #↑C<377>,R5        ;CLEAR HIGH BYTE
010746  020504                           CMP     R5,R4              ;OK?
010750  001401                           BEQ     .+4
010752  104005                           ERROR   5                  ;DATA ERROR
010754  042737  176377  007052           BIC     #176377,RECDAT     ;SAVE ONLY LINE NUMBER
010762  000337  007052                   SWAB    RECDAT
010766  023737  001374  007052           CMP     SAVLIN,RECDAT      ;DOES THE LINE # COMPARE ?
010774  001407                           BEQ     4$                 ;YES
010776  013737  007052  001374           MOV     RECDAT,SAVLIN      ;ADJUST LINE NO. FOR ERROR
011004  104015                           ERROR   15                 ;*WRONG LINE #
011006  013737  007044  001374           MOV     NUMLIN,SAVLIN      ;READJUST LINE NO.
011014  120127  000377          4$:      CMPB    R1,#377            ;LAST CHARACTER ?
011020  001003                           BNE     1$                 ;NO
011022  105237  001425                   INCB    DONFLG             ;INDICATE RECEIVER INTERRUPTS DONE
011026  000401                           BR      2$
011030  105201                  1$:      INCB    R1                 ;UPDATE EXPECTED DATA
011032  000002                  2$:      RTI

011034  005777  170750          INTRAN:  TST     @DZVCSR ;TEST TRANSMIT FLAG
011040  100401                           BMI     .+4
```

```
011042 104003                          ERROR   3               ;#FALSE INTERRUPT
011044 110277 170764                   MOVB    R2,@DZVTDR      ;TRANSMIT A CHARACTER
011050 105202                          INCB    R2              ;UPDATE TX DATA
011052 001003                          BNE     1$              ;BIT PATTERN DONE?
011054 042777 040000 170726            BIC     #TIE,@DZVCSR    ;IF YES THEN CLEAR TIE
011062 000002                   1$:    RTI                     ;IF NOT THEN RETURN
```

```
                                        ;ERROR TABLE
011064  000000              .ERRTAB:            0       ;ERROR 0
011066  000000                                  0
011070  000000                                  0

011072  011312                                  EM1     ;ERROR
011074  012642                                  DH1
011076  013042                                  DT1

011100  011365                                  EM2     ;ERROR 2
011102  012666                                  DH2
011104  013054                                  DT2

011106  011413                                  EM3     ;ERROR 3
011110  012721                                  DH3
011112  013072                                  DT3

011114  011452                                  EM4     ;ERROR 4
011116  012721                                  DH3
011120  013072                                  DT3

011122  011501                                  EM5     ;ERROR 5
011124  012733                                  DH4
011126  013100                                  DT4

011130  011530                                  EM6     ;ERROR 6
011132  012733                                  DH4
011134  013100                                  DT4

011136  011567                                  EM7     ;ERROR 7
011140  012721                                  DH3
011142  013072                                  DT3

011144  011630                                  EM8     ;ERROR 10
011146  012721                                  DH3
011150  013072                                  DT3

011152  011672                                  EM9     ;ERROR 11
011154  012721                                  DH3
011156  013072                                  DT3

011160  011730                                  EM10    ;ERROR 12
011162  012721                                  DH3
011164  013072                                  DT3

011166  011767                                  EM13    ;ERROR 13
011170  012721                                  DH3
011172  013072                                  DT3

011174  012020                                  EM14    ;ERROR 14
011176  012721                                  DH3
011200  013072                                  DT3

011202  012052                                  EM15    ;ERROR 15
011204  000000                                  0
011206  000000                                  0
```

```
 1957
 1958  011210  012114                                   EM16
 1959  011212  012721                                   DH3
 1960  011214  013072                                   DT3
 1961
 1962  011216  012166                                   EM17      ;ERROR 17
 1963  011220  012721                                   DH3
 1964  011222  013072                                   DT3
 1965
 1966  011224  012224                                   EM20
 1967  011226  012721                                   DH3
 1968  011230  013072                                   DT3
 1969
 1970  011232  012265                                   EM21      ;ERROR 21
 1971  011234  012762                                   DH5
 1972  011236  013116                                   DT5
 1973
 1974  011240  012315                                   EM22      ;ERROR 22
 1975  011242  012733                                   DH4
 1976  011244  013100                                   DT4
 1977
 1978  011246  012357                                   EM23      ;ERROR 23
 1979  011250  012721                                   DH3
 1980  011252  013072                                   DT3
 1981
 1982  011254  012407                                   EM24
 1983  011256  012721                                   DH3
 1984  011260  013072                                   DT3
 1985
 1986  011262  012435                                   EM25
 1987  011264  012721                                   DH3
 1988  011266  013072                                   DT3
 1989
 1990  011270  012465                                   EM26
 1991  011272  012721                                   DH3
 1992  011274  013072                                   DT3
 1993
 1994  011276  012514                                   EM27
 1995  011300  012721                                   DH3
 1996  011302  013072                                   DT3
 1997
 1998  011304  012562                                   EM30
 1999  011306  012721                                   DH3
 2000  011310  013072                                   DT3
```

```
2001                                             ;ERROR MESSAGES
2002 011312 047200 020117 052502 EM1:     .ASCIZ  <200>/NO BUS REPLY RESPONSE FROM DZV11 REGISTER/
     011365    200 042522 044507 EM2:     .ASCIZ  <200>?REGISTER R/W FAILURE?
     011413    200 051124 047101 EM3:     .ASCIZ  <200>/TRANSMIT READY (TRDY) NOT SET/
     011452 051200 041505 044505 EM4:     .ASCIZ  <200>/RECEIVER DONE NOT SET/
     011501    200 040504 040524 EM5:     .ASCIZ  <200>/DATA COMPARISON ERROR/
     011530 042200 053132 030461 EM6:     .ASCIZ  <200>/DZV11 *RECEIVER BUFFER* ERROR/
     011567    200 051124 047101 EM7:     .ASCIZ  <200>/TRANSMITTER FAILED TO INTERRUPT/
     011630 052600 042516 050130 EM8:     .ASCIZ  <200>/UNEXPECTED TRANSMITTER INTERRUPT/
     011672 051200 041505 044505 EM9:     .ASCIZ  <200>/RECEIVER FAILED TO INTERRUPT/
     011730 052600 042516 050130 EM10:    .ASCIZ  <200>/UNEXPECTED RECEIVER INTERRUPT/
     011767    200 044523 047514 EM13:    .ASCIZ  <200>/SILO ALARM SET TOO SOON/
     012020 051600 046111 020117 EM14:    .ASCIZ  <200>/SILO ALARM FAILED TO SET/
     012052 040600 052103 047511 EM15:    .ASCIZ  <200>/ACTION DETECTED ON INVALID LINE./
     012114 051200 040505 044504 EM16:    .ASCIZ  <200>/READING DZVRBUF DID NOT CLEAR SILO ALARM/
     012166 042200 052101 020101 EM17:    .ASCIZ  <200>/DATA VALID SHOULD NOT BE SET/
     012224 051200 041505 044505 EM20:    .ASCIZ  <200>/RECEIVER DONE SHOULD NOT BE SET/
     012265    200 042522 040514 EM21:    .ASCIZ  <200>/RELATIVE TIMING ERROR./
     012315    200 047515 042504 EM22:    .ASCIZ  <200>/MODEM SIGNAL ERROR ON CABLE TEST/
     012357    200 040504 040524 EM23:    .ASCIZ  <200>/DATA VALID IS NOT SET!/
     012407    200 040504 040524 EM24:    .ASCIZ  <200>/DATA OVERRUN IS SET!/
     012435    200 051106 046501 EM25:    .ASCIZ  <200>/FRAMING ERROR OCCURRED/
     012465    200 040520 044522 EM26:    .ASCIZ  <200>/PARITY ERROR OCCURRED/
     012514 051600 046111 020117 EM27:    .ASCIZ  <200>/SILO ALARM FAILED TO CAUSE INTERRUPT/
     012562 046200 047111 020105 EM30:    .ASCIZ  <200>/LINE DID NOT RECEIVE FULL BINARY COUNT PATTERN/

     012642 052200 040522 020120 DH1:     .ASCIZ  <200>/TRAP PC  DZV11 REG/
     012666 042600 050130 041505 DH2:     .ASCIZ  <200>/EXPECTED  FOUND  REGISTER/
     012721    200 044514 042516 DH3:     .ASCIZ  <200>/LINE NO./
     012733    200 054105 042520 DH4:     .ASCIZ  <200>/EXPECTED  FOUND  LINE/
     012762 052200 020130 044514 DH5:     .ASCIZ  <200>/TX LINE PREVIOUS TIME  ACTUAL TIME  PARAMETER/

            013042               .EVEN
                                          ;DATA TABLES FOR ERROR MESSAGES
2003 013042 000002               DT1:     2
2004 013044    006    003                 .BYTE   6,3
2005 013046 001330                        $REG1
2006 013050    006    001                 .BYTE   6,1
2007 013052 001326                        $REG0
2008
2009 013054 000003               DT2:     3
2010 013056    006    004                 .BYTE   6,4
2011 013060 001340                        $REG5
2012 013062    006    001                 .BYTE   6,1
2013 013064 001336                        $REG4
2014 013066    006    001                 .BYTE   6,1
2015 013070 001326                        $REG0
2016
2017 013072 000001               DT3:     1
2018 013074    003    001                 .BYTE   3,1
2019 013076 001374                        SAVLIN
2020
2021 013100 000003               DT4:     3
2022 013102    006    004                 .BYTE   6,4
2023 013104 001340                        $REG5
2024 013106    006    001                 .BYTE   6,1
```

```
 013110  001336                           $REG4
 013112     003     001                   .BYTE   3,1
 013114  001374                           SAVLIN

 013116  000004                   DT5:    4
 013120     003     005                   .BYTE   3,5
 013122  001374                           SAVLIN
 013124     006     011                   .BYTE   6,9.
 013126  001340                           $REG5
 013130     006     007                   .BYTE   6,7
 013132  001344                           $TMP1
 013134     006     001                   .BYTE   6,1
 013136  001402                           REGIST
                                                  ;TABLE OF DELAY TIMES FOR INDIVIDUAL BAUD RATES
                                                  ;----------------------------------------------

 013140  002450                   DLYTBL: 2450                    ;TIME FOR   50 BAUD
 013142  001560                           1560                    ;TIME FOR   75 BAUD
 013144  001120                           1120                    ;TIME FOR  110 BAUD
 013146  000750                            750                    ;TIME FOR  134 BAUD
 013150  000660                            660                    ;TIME FOR  150 BAUD
 013152  000330                            330                    ;TIME FOR  300 BAUD
 013154  000150                            150                    ;TIME FOR  600 BAUD
 013156  000060                             60                    ;TIME FOR 1200 BAUD
 013160  000040                             40                    ;TIME FOR 1800 BAUD
 013162  000030                             30                    ;TIME FOR 2000 BAUD
 013164  000020                             20                    ;TIME FOR 2400 BAUD
 013166  000010                             10                    ;TIME FOR 3600 BAUD
 013170  000001                              1                    ;TIME FOR 4800 BAUD
 013172  000001                              1                    ;TIME FOR 7200 BAUD
 013174  000001                              1                    ;TIME FOR 9600 BAUD
 013176  000001                              1                    ;TIME OF DELAY FOR 19200 BAUD

                                          ;DELAYS WERE COMPUTED TO ALLOW MAXIMUM TIME AT EACH BAUD RATE
                                          ;FOR ALL TESTS TO FUNCTION CORRECTLY ON A LSI11.


 013200                           CORMAX:
         000001                   .END
```

```
ABASE = 160010                1#     368      409
ACDW1 = 000000              368      411
ACDW2 = 000000              368      412
ACPUOP= 000000              368      383
ACTIVE  001420              526#
ADDW0 = 000000              368      413
ADDW1 = 000000              368      414
ADDW10= 000000              368      423
ADDW11= 000000              368      424
ADDW12= 000000              368      425
ADDW13= 000000              368      426
ADDW14= 000000              368      427
ADDW15= 000000              368      428
ADDW2 = 000000              368      415
ADDW3 = 000000              368      416
ADDW4 = 000000              368      417
ADDW5 = 000000              368      418
ADDW6 = 000000              368      419
ADDW7 = 000000              368      420
ADDW8 = 000000              368      421
ADDW9 = 000000              368      422
ADEVCT= 000000              368      374
ADEVM = 000000              368      410
ADRCNT  004221             1171*    1208*    1218#
ADVANC= 104400              678#    1348
AENV  = 000000              368      379
AENVM = 000000              368      380
AFATAL= 000000              368      371
AMADR1= 000000              368      396
AMADR2= 000000              368      400
AMADR3= 000000              368      403
AMADR4= 000000              368      406
AMAMS1= 000000              368      390
AMAMS2= 000000              368      398
AMAMS3= 000000              368      401
AMAMS4= 000000              368      404
AMSGAD= 000000              368      376
AMSGLG= 000000              368      377
AMSGTY= 000000              368      370
AMTYP1= 000000              368      391
AMTYP2= 000000              368      399
AMTYP3= 000000              368      402
AMTYP4= 000000              368      405
APASS = 000000              368      373
APRIOR= 000000              368
APTCSU= 000040             1022     1127#
APTENV= 000001             1015     1083     1125#    1435
APTSIZ= 000200             1124#
APTSPO= 000100             1017     1085     1126#
ASWREG= 000000              368      381
ATESTN= 000000              368      372
AUNIT = 000000              368      375
AUSWR = 000000              368      382
AVECT = 000300                1#
AVECT1= 000000              368      407
AVECT2= 000000              368      408
```

```
BAUD     002350            826#   1627
BINWRD   004474           1294#
BIT0   = 000001            167#    239     241     278     289
BIT00  = 000001            157#    167
BIT01  = 000002            156#    166
BIT02  = 000004            155#    165
BIT03  = 000010            154#    164
BIT04  = 000020            153#    163
BIT05  = 000040            152#    162
BIT06  = 000100            151#    161
BIT07  = 000200            150#    160
BIT08  = 000400            149#    159
BIT09  = 001000            148#    158
BIT1   = 000002            166#    240     241     279     290
BIT10  = 002000            147#    263     264     265     266     271     272     273     274     284     295     303
BIT11  = 004000            146#    267     268     269     270     271     272     273     274     285     296     304
BIT12  = 010000            145#    206     222     257    1869
BIT13  = 020000            144#    207     223    1774    1866
BIT14  = 040000            143#    208     224    1863
BIT15  = 100000            142#    209     225
BIT2   = 000004            165#    280     291
BIT3   = 000010            164#    201     244     246     248     250     281     292
BIT4   = 000020            163#    202     245     246     249     250    1345    1360
BIT5   = 000040            162#    203     247     248     249     250     255
BIT6   = 000100            161#    204     252
BIT7   = 000200            160#    193     205     253    1464    1478
BIT8   = 000400            159#    214     216     231     233     260     262     264     266     268     270     272     274
                           282     293     301
BIT9   = 001000            158#    215     216     232     233     261     262     265     266     269     270     273     274
                           283     294     302
BPTVEC= 000014             174#
BRK0   = 000400            301#
BRK1   = 001000            302#
BRK2   = 002000            303#
BRK3   = 004000            304#
BUFSET= 104422             714#
BYTCNT   007036           1674#
CHRCNT   004472           1260*   1273*   1291#
CLEAR  = 000000            194#   1757*   1842*
CNVRT  = 104412            698#    882     884     887     890    1414    1416    1418    1471
CONVRT= 104411             696#   1432
CORMAX   013200           2061#   2062
C00    = 000400            293#
C01    = 001000            294#
C02    = 002000            295#
C03    = 004000            296#
CR     = 000015             82#   1061    1071
CRLF   = 000200             83#   1032    1071
DATABP   005166           1403*   1406    1430    1433#
DATAHD   005154           1402*   1426    1429#
DCLASM= 104417             708#
DCLR   = 000020            202#   1314    1315
DDISP  = 177570             89#    460
DELAY  = 104414            702#   1744    1818
DEVADR   004216           1169*   1205    1216#
DEVICE= 104413             700#    834    1321    1733    1790    1807    1827    1854
```

```
DH1       012642            1907    2002#
DH2       012666            1911    2002#
DH3       012721            1915    1919    1931    1935    1939    1943    1947    1951    1959    1963    1967    1979    1983
                            1987    1991    1995    1999    2002#
DH4       012733            1923    1927    1975    2002#
DH5       012762            1971    2002#
DISPLA    001306             460#    785*   1445*
DISPRE    000174             355#    785
DLYCNT    004570            1327    1332#
DLYTBL    013140            2041#
DONFLG    001425             534#   1838*   1849    1888*
DSWR    = 177570              88#    459
DTR0    = 000400             282#
DTR1    = 001000             283#
DTR2    = 002000             284#
DTR3    = 004000             285#
DT1       013042            1908    2003#
DT2       013054            1912    2009#
DT3       013072            1916    1920    1932    1936    1940    1944    1948    1952    1960    1964    1968    1980    1984
                            1988    1992    1996    2000    2017#
DT4       013100            1924    1928    1976    2021#
DT5       013116            1972    2029#
DVALID=   100000             225#
DZCR0     001500             574#
DZCR1     001512             580#
DZCR10    001620             622#
DZCR11    001632             628#
DZCR12    001644             634#
DZCR13    001656             640#
DZCR14    001670             646#
DZCR15    001702             652#
DZCR16    001714             658#
DZCR17    001726             664#
DZCR2     001524             586#
DZCR3     001536             592#
DZCR4     001550             598#
DZCR5     001562             604#
DZCR6     001574             610#
DZCR7     001606             616#
DZVACT    001406             519#
DZVCSR    002010             722#    913    1314*   1315    1322*   1707*   1739*   1742    1758*   1768    1793    1837*   1857
                            1893    1899*
DZVC0     001502             575#
DZVC1     001514             581#
DZVC10    001622             623#
DZVC11    001634             629#
DZVC12    001646             635#
DZVC13    001660             641#
DZVC14    001672             647#
DZVC15    001704             653#
DZVC16    001716             659#
DZVC17    001730             665#
DZVC2     001526             587#
DZVC3     001540             593#
DZVC4     001552             599#
DZVC5     001564             605#
```

```
DZVC6    001576            611#
DZVC7    001610            617#
DZVLEV   007660            821     1696#
DZVLPR   002020            726#    1357*   1712*   1738*   1832*
DZVMSR   002030            730#    1721*   1819
DZVNUM   001414            522#
DZVRBU   002014            724#    1711*   1771    1860
DZVRIS   002042            738#    1698*   1756*   1834*
DZVRIV   002040            737#     810     916    1696    1755*   1833*
DZVTCR   002024            728#    1717*   1735*   1841*
DZVTDR   002034            732#    1722*   1749*   1795*   1896*
DZVTIS   002046            740#    1702*   1836*
DZVTIV   002044            739#    1700*   1835*
DZV.EN   001740            670#
DZV.MA   001500            526      572#
EIGHT =  000030            246#
EIGHTS=  000070            250#
EMTVEC=  000030            177#
EM1      011312           1906     2002#
EM10     011730           1942     2002#
EM13     011767           1946     2002#
EM14     012020           1950     2002#
EM15     012052           1954     2002#
EM16     012114           1958     2002#
EM17     012166           1962     2002#
EM2      011365           1910     2002#
EM20     012224           1966     2002#
EM21     012265           1970     2002#
EM22     012315           1974     2002#
EM23     012357           1978     2002#
EM24     012407           1982     2002#
EM25     012435           1986     2002#
EM26     012465           1990     2002#
EM27     012514           1994     2002#
EM3      011413           1914     2002#
EM30     012562           1998     2002#
EM4      011452           1918     2002#
EM5      011501           1922     2002#
EM6      011530           1926     2002#
EM7      011567           1930     2002#
EM8      011630           1934     2002#
EM9      011672           1938     2002#
ERRMSG   005142           1401*    1421    1424#
ERRVEC=  000004            170#
ERTABO   005316           1416     1457#
EVEPAR=  000000            256#
EXITER   005246           1444     1447#
FIVE  =  000000            243#
FIVES =  000040            247#
FRMERR=  020000            223#
HALTS    005172           1386     1435#
HDRFLG   001423            532#
HDZVCS   002012            723#    1709*
HDZVLP   002022            727#    1715*
HDZVMS   002032            731#    1724*
HDZVRB   002016            725#    1714*
```

```
HDZVTC  002026              729#    1719*   1813*
HDZVTD  002036              733#    1725*
HILIM   004214             1168*    1196    1215#
HT    = 000011               80#    1030    1071
INBUF   007512              933     1138    1174     1567*   1568    1573    1578    1596*   1597    1601    1688#   1779*   1780*
                           1788
INIFLG  001422              531#     792     797*
INSTER= 104404              686#     952    1190     1584    1606
INSTR = 104403              684#     805     813      822     826     836
INSTR2  004014             1145     1157#
INTRAN  011034             1835     1893#
INTREC  010644             1833     1856#
INTSVC  010226             1755     1768#
IOTVEC= 000020              175#
LAST    007030              791*    1671#
LF    = 000012               81#    1065    1071
LIMITS  004142             1184     1196#
LINE    001366              511#
LINESP  007040              831     1622    1675#
LINEX   002366              834#    1764    1848
LINE0   001504              576#
LINE1   001516              582#
LINE10  001624              624#
LINE11  001636              630#
LINE12  001650              636#
LINE13  001662              642#
LINE14  001674              648#
LINE15  001706              654#
LINE16  001720              660#
LINE17  001732              666#
LINE2   001530              588#
LINE3   001542              594#
LINE4   001554              600#
LINE5   001566              606#
LINE6   001600              612#
LINE7   001612              618#
LOBITS  004220             1170*    1200    1217#
LOCK    001364              506#     983     985     1338*   1410    1812*   1828*
LOCKUP  007026              848*    1670#   1851*    1856*
LOLIM   004212             1167*    1198    1214#
LPRSET= 104421              712#
LP0   = 000000              238#
LP1   = 000001              239#
LP2   = 000002              240#
LP3   = 000003              241#
MAINT = 000010              201#    1576
MANT0   001510              578#
MANT1   001522              584#
MANT10  001630              626#
MANT11  001642              632#
MANT12  001654              638#
MANT13  001666              644#
MANT14  001700              650#
MANT15  001712              656#
MANT16  001724              662#
MANT17  001736              668#
```

```
MD-11-DVDZC-A   MACY11 30(1046)  26-JUL-77  08:34  PAGE 53
DVDZCA.P11    25-JUL-77 11:21          CROSS REFERENCE TABLE -- USER SYMBOLS

MANT2   001534             590#
MANT3   001546             596#
MANT4   001560             602#
MANT5   001572             608#
MANT6   001604             614#
MANT7   001616             620#
MASK  = 000200             193#    777*    846*   1756    1762*   1804*   1834    1836
MASTEK  006245            1412    1555#
MBADLN  006356            1555#
MCABLE  007360             855    1685#
MCHAR   007224            1685#   1759
MCSRX   006175             881    1417    1555#
MDATA   007616            1271    1281    1692#
MEPASS  006013             880    1555#
MERRPC  006323            1415    1555#
MERRX   006222             889    1555#
MERR2   006053            1555#
MERR3   006122            1555#
MINVAL  007150            1626    1685#
MLINE   007176             837    1685#
MLOCK   006146            1555#
MNEW    006250            1555#
MNTFLG  001424             533#   1322    1571*   1576*   1581*
MODE    001372             513#
MPASS   007134            1685#
MPASSX  006211             886    1555#
MPFAIL  005750            1548    1555#
MQUICK  007375            1685#   1741
MR      006037            1555#
MREGAD  007100             814    1685#
MSENAB= 000040             203#   1739    1758    1837
MSPEED  007206             827    1685#
MTERM   007333             861    1685#
MTITLE  001000             362#    796
MTSTN   006233            1413    1555#
MVECTO  007056             806    1685#
MVECX   006203             883    1555#
MWHICH  007265             823    1685#
NEXT    001362             505#   1337    1453    1754*   1806*
NOLIST= ###### U              1
NUMLIN  007044            1612*   1678#   1736    1787    1830    1885
NUMTCR  007046            1614*   1621*   1680#   1735    1813    1815    1817    1841
ODDPAR= 000200             253#
ONESTO= 000000             254#
OVRRUN= 040000             224#
PAR     001370             512#   1355    1736*   1737*   1738    1829*   1830*   1831*   1832
PARAM = 104405             688#    807     815     838
PARAM1  004062            1173#   1191
PARER = 010000             222#
PARERR  004136            1176    1178    1180    1190#   1197    1199    1201
PARITY= 000100             252#
PARMD = 104415             704#    828
PAR0    001506             577#
PAR1    001520             583#
PAR10   001626             625#
PAR11   001640             631#
```

```
PAR12     001652             637#
PAR13     001664             643#
PAR14     001676             649#
PAR15     001710             655#
PAR16     001722             661#
PAR17     001734             667#
PAR2      001532             589#
PAR3      001544             595#
PAR4      001556             601#
PAR5      001570             607#
PAR6      001602             613#
PAR7      001614             619#
PAWCH  =  104416             706#     824
PIRQ   =  177772              87#
PIRQVE=   000240             181#
POPRO  =  012600             190#
POP1SP=   005726             188#
POP2SP=   022626             192#    1347
PRIO      007050            1681#
PR0    =  000000             104#
PR1    =  000040             105#
PR2    =  000100             106#
PR3    =  000140             107#
PR4    =  000200             108#
PR5    =  000240             109#
PR6    =  000300             110#
PR7    =  000340             111#
PS     =  177776              84#      85
PSW    =  177776              85#
PUSHRO=   010046             189#
PUSH1S=   005746             187#
PUSH2S=   024646             191#
PWRVEC=   000024             176#    1517*    1518*    1527*    1533*    1545*    1546*
QUITS     010636            1850     1854#
RCVON  =  010000             257#    1831
RDATA     007034            1673#
RDONE  =  000200             205#
RECDAT    007052            1683#    1771*    1774     1778     1779     1781*    1782*    1783     1785     1860*    1863     1866     1869
                            1873     1879*    1880*    1881     1883
REGIST    001402             517#    2037
RESREG    005170            1431     1434#
RESTAR    002512             863#    1550
RESVEC=   000010             171#
RESOS  =  104410             694#    1434
RIE    =  000100             204#    1758     1837
RINGO  =  000001             289#
RING1  =  000002             290#
RING2  =  000004             291#
RING3  =  000010             292#
RL0    =  000000             230#
RL1    =  000400             231#
RL2    =  001000             232#
RL3    =  001400             233#
RUN       001412             521#
SAVACT    001410             520#
SAVLIN    001374             514#     841     1612     1613     1783     1785*    1787*    1881     1883*    1885*    2019     2027     2031
```

D06

```
MD-11-DVDZC-A   MACY11 30(1046)  26-JUL-77  08:34  PAGE 55                                                          PAGE:  0068
DVDZCA.P11    25-JUL-77 11:21          CROSS REFERENCE TABLE -- USER SYMBOLS

SAVNO   001416          524#
SAVNUM  001415          523#
SAVPC   001404          518#   1223*   1459
SAV05 = 104407          692#   1391
SCOP1 = 104401          680#   1826
SERV.G  005332         1379    1448    1463#   1468    1751    1763    1847
SET     006626          844    1612#
SETFLG= 104406          690#
SET1    006660         1617    1620#
SEVEN = 000020          245#
SEVENS= 000060          249#
SHIFT = 104420          710#
SILOAL= 020000          207#
SILOEN= 010000          206#
SIX   = 000010          244#
SIXS  = 000050          248#
SPACNT  004473         1292#
SPEED   007042         1629*   1676#   1737    1829
SPIN    010602         1845#   1852
STACK = 001120           75#    776     847    1454
STFLG   007024          786*    835*    852     854*    858     860*   1669#
STKLMT= 177774           86#
STOP    001446          545#   1371
SV05    004230         1227#
SWR     001304          459#    782*    784*    981     988    1380    1385    1443    1449    1451    1469    1474*   1486*
                       1487*   1488*   1495*   1499*   1525    1538*   1752
SWREG   000176          356#    784
SW0   = 000001          139#
SW00  = 000001          129#    139
SW01  = 000002          128#    138
SW02  = 000004          127#    137
SW03  = 000010          126#    136
SW04  = 000020          125#    135
SW05  = 000040          124#    134
SW06  = 000100          123#    133
SW07  = 000200          122#    132
SW08  = 000400          121#    131    1449
SW09  = 001000          120#    130     981
SW1   = 000002          138#
SW10  = 002000          119#   1451
SW11  = 004000          118#
SW12  = 010000          117#    988    1380
SW13  = 020000          116#   1385
SW14  = 040000          115#
SW15  = 100000          114#
SW2   = 000004          137#
SW3   = 000010          136#
SW4   = 000020          135#
SW5   = 000040          134#
SW6   = 000100          133#
SW7   = 000200          132#
SW8   = 000400          131#
SW9   = 001000          130#
S110  = 001000          261#
S1200 = 003400          266#
S134  = 001400          262#
```

```
S150   = 002000              263#
S1800  = 004000              267#
S19200= 007400               274#
S2000  = 004400              268#
S2400  = 005000              269#
S300   = 002400              264#
S3600  = 005400              270#
S4800  = 006000              271#
S50    = 000000              259#
S600   = 003000              265#
S7200  = 006400              272#
S75    = 000400              260#
S9600  = 007000              273#
TABLE2   006722             1620     1635#
TBITVE= 000014               172#
TBUF     007054             1684#    1778*    1795
TCR0   = 000001              278#
TCR1   = 000002              279#
TCR2   = 000004              280#
TCR3   = 000010              281#
TDATA    007032             1672#
TD0      001426              537#    1369
TD1      001430              538#
TD2      001432              539#
TD3      001434              540#
TEIGHT   002106              763#
TEMP     007554             1690#
TFIVE    002114              766#
TIE    = 040000              208#    1837     1899
TKVEC  = 000060              179#
TL0    = 000000              213#
TL1    = 000400              214#
TL2    = 001000              215#
TL3    = 001400              216#
TMTBL    002050              747#
TPVEC  = 000064              180#
TRAPVE= 000034               178#
TRDY   = 100000              209#
TRTVEC= 000014               173#
TR0      001436              541#
TR1      001440              542#
TR2      001442              543#
TR3      001444              544#
TSEVEN   002110              764#
TSIX     002112              765#
TST1     010020              857     1733#
TST2     010374              851     1804#
TWOSTO= 000040               255#
TYPDAT   005156             1407     1427     1430#
TYPE   = 104402              682#     796      855      861      880      881      883      886      889     1035     1136     1155     1248
                            1281     1408     1409     1412     1413     1415     1417     1419     1423     1428     1470     1472     1500
                            1547     1626     1759
TYPMSG   005046             1405     1408#
T110     002054              750#
T1200    002066              755#
T134     002056              751#
```

```
T150      002060           752#
T1800     002070           756#
T2000     002072           757#
T2400     002074           758#
T300      002062           753#
T3600     002076           759#
T4800     002100           760#
T50       002050           748#
T600      002064           754#
T7200     002102           761#
T75       002052           749#
T9600     002104           762#
VEC1      002254           793      798#
WCHFLG    007022           825      849      1668#
WRDCNT    004470          1256*    1282*     1290#
WTBS.F    005144          1422     1425#
XBEGIN    002416           846#     909
XBX       004734          1381     1383      1385#
XCSR      002660           882      911#     1418
XERR      002702           890      920#
XHEAD     006330          1555#
XMTCNT    001400           516#
XMTLIN    001376           515#
XPASS     002674           887      917#
XSTATQ    006420          1555#
XTCR1     006650          1616#    1619
XTSTN     005324          1414     1460#
XVEC      002666           884      914#
XX      = 160210           573#     579#      585#     591#     597#     603#     609#     615#     621#     627#     633#     639#     645#
                           651#     657#      663#     669#
YY      = 000500           573#     579#      585#     591#     597#     603#     609#     615#     621#     627#     633#     639#     645#
                           651#     657#      663#     669#
ZZ      = 000020           573#     579#      585#     591#     597#     603#     609#     615#     621#     627#     633#     639#     645#
                           651#     657#      663#     669#
$APTHD    001446           555      561#
$ASTAT= ****** U          1105     1120
$ATYC     003454          1076     1078#
$ATY1     003430          1074#
$ATY3     003436          1020     1075#
$ATY4     003446          1077#    1438
$AUTOB    001300           456#
$BASE     001174           409#     818      1706
$BDADR    001266           451#
$BDDAT    001272           453#
$CDW1     001200           411#
$CDW2     001202           412#
$CHARC    003424          1037*    1047*     1054     1063*    1068#
$CMTAG    001244           439#
$CM1    = 000006           471#     472#      473#     474#     475#     476#     477#
$CM2    = 000014           471#     472#      473#     474#     475#     476#     477#
$CM3    = 000006           469#     471
$CM4    = 000005           477#     478#      479#     480#     481#     482#
$CPUOP    001146           383#
$CRLF     001357           484#    1036      1071     1248     1408     1409     1419     1500
$DDW0     001204           413#
$DDW1     001206           414#
```

```
$DDW10   001230          423#
$DDW11   001232          424#
$DDW12   001234          425#
$DDW13   001236          426#
$DDW14   001240          427#
$DDW15   001242          428#
$DDW2    001210          415#
$DDW3    001212          416#
$DDW4    001214          417#
$DDW5    001216          418#
$DDW6    001220          419#
$DDW7    001222          420#
$DDW8    001224          421#
$DDW9    001226          422#
$DEVCT   001130          374#
$DEVM    001176          410#
$DOAGN   002654          896      901      907#
$E     = 000002            1#
$ENDAD   002644          349      794      903#     1441
$ENDCT   002630          898#
$ENV     001140          379#    1015     1083     1107     1435
$ENVM    001141          380#     780     1017     1022     1085
$EOP     002516          877#    1754     1791     1806     1855
$EOPCT   002622          895#     899
$ERFLG   001247          442#     789*     879*    1390*    1404     1420*
$ERMAX   001261          448#
$ERROR   004704          339     1379#
$ERRPC   001262          449#     878*    1387     1389*
$ERRTB   001362          500#
$ERTTL   001256          446#     788*     922     1447*
$ETABL   001140          378#
$ETEND   001244          431#     567
$FATAL   001122          371#    1111*
$FFLG    003674         1074*    1077*    1105     1114*    1122#
$FILLC   001322          467#    1040     1071
$FILLS   001321          466#    1071
$FLIP  = 177777            1#
$GDADR   001264          450#
$GDDAT   001270          452#
$GET42   002634          900#
$HD    = 000001           10       11
$HIBTS   001446          562#
$ICNT    001250          443#
$ILLUP   005742         1517     1533     1552#
$INTAG   001301          457#
$ITEMB   001260          447#    1395*    1437
$LF      001360          485#    1071
$LFLG    003673         1115*    1121#
$LPADR   001252          444#     779*     851*     857*     863     1453*    1455
$LPERR   001254          445#
$MADR1   001152          396#
$MADR2   001156          400#
$MADR3   001162          403#
$MADR4   001166          406#
$MAIL    001120          369#     563      567     1015
$MAMS1   001150          390#
```

```
$MAMS2  001154          398#
$MAMS3  001160          401#
$MAMS4  001164          404#
$MBADR  001450          563#
$MFLG   003672         1075*   1081     1116*    1120#
$MSGAD  001134          376#   1091*    1094
$MSGLG  001136          377#   1096*
$MSGTY  001120          370#   1089     1097*    1109     1113*
$MTYP1  001151          391#
$MTYP2  001155          399#
$MTYP3  001161          402#
$MTYP4  001165          405#
$N    = 000001            1#
$NULL   001320          465#   1042     1071
$PASS   001126          373#    787*     885*     888*     892*     893*     910      919
$PASTM  001454          565#
$PWRAD  005736         1550#
$PWRDN  005576          337     778     1517#    1545
$PWRMG  005732         1548#
$PWRUP  005650         1527    1533#
$QUES   001356          483#   1071     1155
$REGAD  001324          469#
$REG0   001326          471#   1232*    1237     2007     2015
$REG1   001330          472#   1231*    1238     2005
$REG2   001332          473#   1230*    1239
$REG3   001334          474#   1229*    1240
$REG4   001336          475#   1228*    1241     2013     2025
$REG5   001340          476#   1227*    1242     2011     2023     2033
$RTNAD  002656          909#
$SAVR6  005746         1526*   1534     1535*    1536*    1554#
$SETUP= 000000          891
$SVPC = 000040          347#    352
$SWR  = 164000            1#     10      482      483      874      891      902      908      910      1551
$SWREG  001142          381#    782
$TESTN  001124          372#
$TIMES  001354          482#    891*
$TKB    001312          462#   1142     1148     1463     1477
$TKS    001310          461#   1140     1467     1475     1760     1845
$TMP0   001342          477#
$TMP1   001344          478#   1557     2035
$TMP2   001346          479#   1559
$TMP3   001350          480#
$TMP4   001352          481#
$TN   = 000001           10#
$TPB    001316          464#   1060*    1071     1148*    1384*    1484*
$TPFLG  001323          468#   1009     1071
$TPS    001314          463#   1058     1071     1146     1382     1481
$TSTM   001452          564#
$TSTNM  001246          441#    790*    1462     1734*    1805*
$TYPE   003146          989    1009#    1102
$TYPEC  003360         1039    1046     1053     1058#    1059
$TYPEX  003426         1064    1066     1069#
$UNIT   001132          375#
$UNITM  001456          566#
$USWR   001144          382#
$VECT1  001170          407#
```

```
$VECT2  001172            408#
$Y     = 000023           671#    678     680#    682#    684#    686#    688#    690#    692#    694#    696#    698#    700#
                          702#    704#    706#    708#    710#    712#    714#    716#
$$GET4= 000000            902#
.      = 013200           332#    333     336#    347     348#    350#    352#    354#    357#    361#    363#    486     519#
                          520#    522#    524#    525#    551     552#    554#    556#    571#    574#    575#    576#    577#
                          578#    580#    581#    582#    583#    584#    586#    587#    588#    589#    590#    592#    593#
                          594#    595#    596#    598#    599#    600#    601#    602#    604#    605#    606#    607#    608#
                          610#    611#    612#    613#    614#    616#    617#    618#    619#    620#    622#    623#    624#
                          625#    626#    628#    629#    630#    631#    632#    634#    635#    636#    637#    638#    640#
                          641#    642#    643#    644#    646#    647#    648#    649#    650#    652#    653#    654#    655#
                          656#    658#    659#    660#    661#    662#    664#    665#    666#    667#    668#    910     1071
                         1123#   1512#   1529    1553    1689#   1691#   1693#   1769    1772    1775    1858    1861    1864
                         1867    1870    1877    1894    2002#
.ADVAN  004572            679    1337#
.BUFSE  004662            715    1368#
.CNVRT  004320            699    1249#
.CONVR  004314            697    1248#
.DCLAS  004540            709    1321#
.DELAY  004552            703    1325#
.DEVIC  004520            701    1313#
.ERRTA  011064           1400    1902#
.INSTE  004002            687    1153#
.INSTR  003676            685    1132#
.INST1  003716           1136#   1156
.LPRSE  004622            713    1353#
.MSG    003720           1134*   1137#
.PARAM  004022            689    1164#
.PARMD  002710            705     925#
.PAWCH  006552            707    1595#   1607
.RESOS  004262            695    1237#
.SAVOS  004222            693    1223#
.SCOP1  003104            681     981#
.SETFL  006432            691    1566#   1585
.SHIFT  004604            711    1344#
.START  002116            358     775#    779
.TRPSR  004476            341    1302#
.TRPTA  001742            677#   1307
.TYPE   003130            683     988#
.$ASTA= ****** U         1075    1078
.$X    = 001446           551#    556
```

```
COMMEN    182#
ENDCOM    182#
ERROR      76#   1747    1770    1773    1776    1786    1822    1853    1859    1862    1865    1868    1871    1878    1884
         1895
ESCAPE    182#
GETPRI    182#
GETSWR    182#
MULT      182#
NEWTST    182#
PASEND      1#    878
POP       182#   1117    1118    1538    1539
PRGEND      1#    865
PRGFRT      1#
PUSH      182#   1078    1080    1101    1519    1525
REPORT      1#    182#
SCOPE      77#
SETPRI    182#
SETUP     182#
SKIP      182#
SLASH     182#
SPACE     182#
STARS     182#    345     364     367     435     548     550     557     872     994    1073    1515    1531
SWRSU     182#
TYPBIN    182#
TYPDEC    182#
TYPNAM    182#
TYPNUM    182#
TYPOCS    182#
TYPOCT    182#
TYPTXT    182#
$BUFFE      1#   1685
$EOP        1#    865
$GETFL      1#
$GETPA      1#
$HEADE      1#     11
$INTSE      1#
$JUNK       1#    573     579     585     591     597     603     609     615     621     627     633     639     645     651
          657     663
$MRESE      1#
$MSG        1#   1555
$SETFL      1#   1561
$STAG       1#
$STAGF      1#
$TRPDE      1#    678     680     682     684     686     688     690     692     694     696     698     700     702     704
          706     708     710     712     714
$TSTN       1#
$VARIA      1#    360
$XZ         1#
$$CMRE    363#    471     472     473     474     475     476
$$CMTM    363#    477     478     479     480     481
$$ESCA    182#
$$NEWT    182#
$$SKIP    182#
.EQUAT      1#     72
.HEADE      1#
.SETUP      1#
```

```
MD-11-DVDZC-A   MACY11 30(1046)  26-JUL-77  08:34  PAGE 63
DVDZCA.P11     25-JUL-77 11:21          CROSS REFERENCE TABLE -- MACRO NAMES

.$ACT1        1#     343
.$APTB        1#     365#
.$APTH        1#     546
.$APTY        1#    1071
.$CATC        1#
.$CMTA     363#
.$EOP         1#     870
.$ERRO        1#
.$POWE        1#    1513
.$TRAP        1#
.$TYPE        1#     992


. ABS.  013200      000

 ERRORS DETECTED:  0

 DVDZCA,DVDZCA.SEQ=DVDZCA.P11
 RUN-TIME: 17 8 1 SECONDS
 RUN-TIME RATIO: 86/27=3.2
 CORE USED:  28K  (55 PAGES)
```