DZV-11

(4) LINE ASYNCHRONOUS MUX

MD-11-DVDZB-A

TESTS, PART 2 OF 2

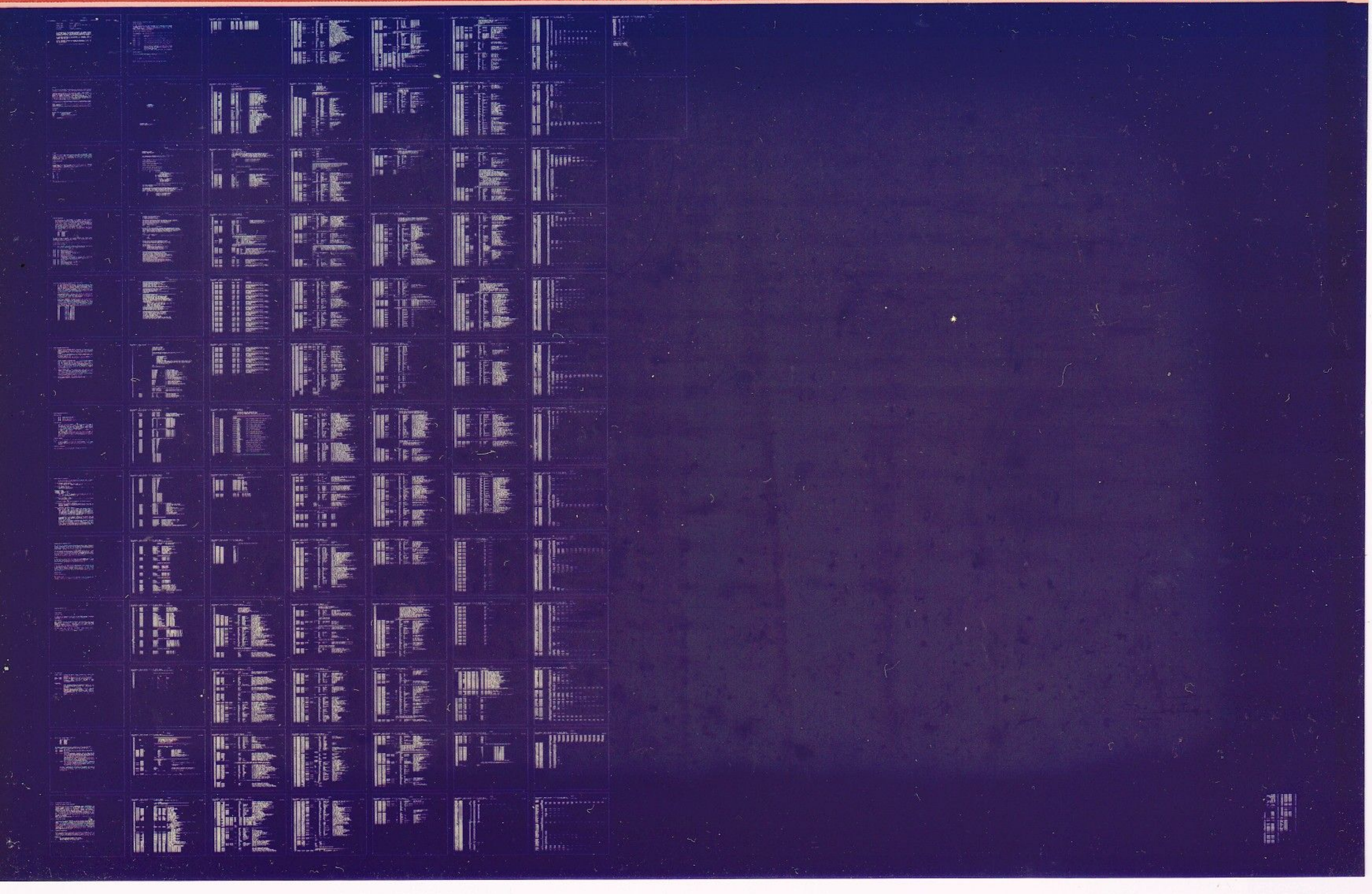
EP-DVDZB-A-DL-A

FICHE 1 OF 1

OCT 1977

digital

MADE IN USA



EOF1DVDZBASB0411 00010000 770920 IDENTIFICATION 2HDR1DVDZBASE0 00010000 770920
SEQ 0001

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

PRODUCT NAME: DZV11 4 LINE ASYNC MUX TESTS PART 2 OF 2

DATE RELEASED: APRIL 1977

MAINTAINER: DIAGNOSTIC ENGINEERING

1. ABSTRACT

The function of the DZV11 diagnostics is to verify the option operates according to specifications. The diagnostics also verify that the DZV11 operates in its environment such as the system in which it is installed.

Parameters may be supplied to the program by either 'AUTO SIZING' or input from the user on the console by having SW00=1 at start time. Auto sizing will be done only the first time the program is started and SW07=0 and SW00=0 and SW03=0. The AUTOSIZER is designed to detect DZV11 device addresses and vectors only. All remaining parameters will default to certain values (see Sec.8.5). Console input may be controlled at any start time through the use of SW00,SW03, SW04, and SW06 (see Sec. 4.1.1 for a detailed description of these switches).

Currently there are three standalone diagnostics (DVDZA,DVDZB,and DVDZC) one system module for DEC X/11 (DZBA), and an overlay for ITEP (DVDZD).

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

2. REQUIREMENTS

2.1 EQUIPMENT

An LSI11 CPU with minimum 4K of memory.
ASR 33 (or equivalent for console)
DZV11 INTERFACE MODULE
H329 Staggered turnaround connector.
H325 Cable turnaround connector.

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

2.2 STORAGE

Program will use all 4K of memory except where ABL and BOOTSTRAP LOADER reside. Location 1500 thru 1740 are especially to be noted and to be untouched by operator after parameters have been input from console (SWOO=1); or after the 'AUTO SIZING' has been done. These locations may be changed if the user understands their meaning and different parameters are required.

3. LOADING PROCEEDURE

3.1 METHOD

All programs are in absolute format and are loaded using the ABSOLUTE LOADER. NOTE: if the diagnostics are on a media such as DISK ,MAGTAPE,DECTAPE, or CASSETTE; follow instructions for the monitor which has been provided on that specific media.

ABSOLUTE LOADER starting address *500

MEMORY	* SIZE
4k	17
8k	37
12k	57
16k	77
20k	117
24k	137
28k	157

3.1.1 Starting the processor at the Absolute Loader starting address will load the diagnostic into memory.

4. STARTING PROCEEDURE

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

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

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

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

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

4.1 CONTROL SWITCH SETTINGS

NOTE: This program utilizes a Software Switch Register which may be modified by changing Loc. 176 or by typing Control "G" (↑G) on the console terminal while the program is running.

```
SW 15   Set:  Halt on error
SW 14   Set:  Loop on current test
SW 13   Set:  Inhibit error print out
SW 12   Set:  Inhibit **ALL** type out/bell on error.
SW 11   Set:  Inhibit iterations.  (quick pass)
SW 10   Set:  Escape to next test
SW 09   Set:  Loop with current data
SW 08   Set:  Catch error and loop on it
SW 07   Set:  NO AUTO SIZE.  If 1st start of program after  loading and
              if SW00=0 then the program will assume that the status map
              has been built from a previous DZV11 diagnostic run.
SW 06   Set:  Reselect DZV11's desired active
SW 05   Set:  Reserved
SW 04   Set:  Select delay parameter (see SEC. 4.1.1)
SW 03   Set:  Extra parameter input  (see SEC. 4.1.1)
SW 02   Set:  Lock on selected test
SW 01   Set:  Restart program at selected test
SW 00   Set:  Get users parameters from console
```

4.1.1 SWITCH REGISTER CONTROL OF PARAMETER INPUT FROM CONSOLE

SW 00 GET USERS PARAMETERS FROM CONSOLE. Setting this switch at start up time allows the user to input at the Console terminal the following parameters: base device address, base vector address, mode of operation (EXTERNAL, INTERNAL, OR STAGGERED), and the number of DZV11's that are running. Using this switch alone will default the following parameters: all 4 lines are set to be tested on each DZV11, the default baud rate is set at 19.2 Kbaud and the character length for the majority of testing is set at eight bits per character with two stop bits.

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

SW 04 SELECT DELAY PARAMETER. The DELAY parameter this switch controls determines the length of time the program stalls waiting for a character to be completely transmitted or received. This delay count is automatically set to provide enough delay time for the default baud rate specified when running the program on an LSI11 with MOS memory. When running this program on a processor with a faster memory speed this delay count should be adjusted proportionately higher than the following defaulted values:

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

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 06 RESELECT DZV11'S DESIRED ACTIVE. A message is typed out on the console terminal asking the operator to type a bit map of the DZV's desired active. Using this switch allows location DZVACTV to be altered (see Sec. 8.3 for a description of this location).
EXAMPLE:
If the devices corresponding to the DZV11's numbered zero, two, and four in the DZV11 Status Map (Loc. 1500 through 1740) are to be tested, type in: 25
This will set bits zero, two, and four in location DZVACTV. All remaining devices in the status map will then not be tested.

SW 01 RESTART PROGRAM AT SELECTED TEST it is strongly suggested that at least one pass has been made before trying to select a test that is not in the order of sequence the reason being is that the program has to clear areas and set up parameters.
Note: if running multiple DZV11's; the DZV11 you desire to be under test must be selected by the use of SW06 before locking on the test. In other words; each time the program is started; the first DZV11 will be selected to be under test unless SW06 is used to select only one.

SW 09 LOOP ON CURRENT DATA: this switch will only work if call 'SCOP1' is in that test. The reason being that most tests deal with blocks of different data to be sent or received all at once thus in block data, one pattern can't be singled out.
This switch is designed to provide an aid for a trained trouble-shooter to sample various signals on the module and is not meant to be used as a general user control switch.

SW 04 SELECT DELAY PARAMETER: THIS SWITCH SHOULD BE USED WITH CARE AS TOO SHORT A DELAY WILL CAUSE VALID TESTS TO FAIL.
(see Sec. 4.1.1)

4.1.3 SWITCH REGISTER PRIORITIES

ERROR SWITCHES

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

SCOPE SWITCHES

1. SW 09 (if enabled by 'SCOP1'). If an '*' is printed in front of the test no. on an error report (ex. *TEST NO. 10) SW09 is incorporated in that test and therefore SW09 is *usually* the best switch for the scope loop (SW14=0, SW10=0, SW09=1, SW08=0) if the program user is technically trained to electronically isolate signal problems on the DZV11 module.
   If SW09 is not enabeled; and there is a *HARD* error (constant); SW08 is best.
2. For intermittent errors either start the program with SW01 and SW02 set which will allow the user to lock on a selected test, or else set SW14 as an error is being typed out on the terminal. SW14 will continue to loop on that test regardless of whether an error occurs.
3. SW 14 Loop on current test.

4.2 STARTING ADDRESS

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

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

5. OPERATING PROCEEDURE

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

5.1 NORMAL START OF DIAGNOSTIC

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

"1ST CSR ADDRESS (160000:163770): "
You must type in the first DZV11 CSR in the system you wish testing to begin at. RANGE: 160000:163770

"1ST VECTOR ADDRESS (300:770): "
You must type in the vector of the first DZV11 in the system under test. RANGE 300:770

"Maintenance Mode
[EXTERNAL <H325> (E)]
[INTERNAL <DZCSR03=1>(I)]
[STAGGERED <H329> (S)] :
Type "E" or "I" or "S" depending on which mode you wish to run in. If running "EXTERNAL"; all selected lines must be terminated by an H325 test connector.

"# OF DZV11'S <IN OCTAL> (1:20): "
Type total number of DZV11's to be tested in the system. RANGE is 1 thru 20 in octal.

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

"LINES ACTIVE BY BIT <IN OCTAL> (001:017):"
Each bit represents a line and any combination of lines may be selected (HOWEVER IN STAGGERED MODE TWO ADJACENT LINES MUST BE SELECTED (0-1, 2-3).

"DEFAULT BAUD RATE <IN OCTAL> (00:17): "
This gives the user a chance to change the default baud rate used in APP. 90% of the test. Baud rate choices are:
"00"(50 baud),"01"(75 baud),"02"(110 baud),"03"(134 baud),
"04"(150 baud),"05"(300 baud),"06"(600 baud),"07"(1200 baud),
"10"(1800 baud),"11"(2000 baud),"12"(2400 baud),"13"(3600 baud),
"14"(4800 baud),"15"(7200 baud),"16"(9600 baud),"17"(19.2 kbaud)
Low default baud rates are not suggested since they lengthen the time to complete a program pass dramatically.

It is important to note that all DZV11's in the system must be CONTIGIOUS for both ADDRESS and VECTORS. Also all the EXTRA PARAMETERS other than CSR and VECTORS are given to the EXISTING DZV11's in the system.
If the mode of operation is different for each DZV11 THIS MUST BE PATCHED INTO THE CORRECT STATUS MAP ENTRY which is printed at start time. An alternative is to put SW00=1 at start time; answer questions about DZV11 under test and INDICATE ONE DZV11 in the system. IF THE STATUS MAP IS TO BE "PATCHED" IT MUST BE DONE AFTER THE QUESTIONS ARE ANSWERED OR AFTER THE AUTO SIZE.

5.2 PROGRAM AND/OR OPERATOR ACTION

The variety of program Control Switches provided in this Diagnostic Package is designed to provide the user with a wide range of 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 operator to regain control of cpu. If this should happen; look in location 'STSTNM' (address 1246) for the number of the test that was running at the time of the catastrophic error. In this way the operator will have an idea as to what the DZV11 was doing at the time of the error.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

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

7.2 OPERATING RESTRICTIONS

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

8. MISCELLANEOUS

8.1 EXECUTION TIME

All DZV11 device diagnostics will give an 'END PASS' message (providing no errors and SW12=0) within 2 min. This is assuming SW11=1 (INHIBIT ITERATIONS) is set to give the fastest possible execution.

8.2 PASS COMPLETE

NOTE: *EVERY* time the program is started; the tests will run as if SW11 (delete iterations) was up (=1). This is to 'VERIFY NO *HARD* ERRORS' as soon as possible. Therefore the first pass -EACH TIME PROGRAM IS STARTED- will be a 'QUICK PASS' until all DZV11's in system are tested. When the diagnostic has completed a pass the following is an example of the print out to be expected.

END PASS DVDZB-A CSR: 160100 VEC: 300 PASSES: 000001 ERRORS: 000000

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

8.3 KEY LOCATIONS

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

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

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

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

1500	160100	This is the system control register for the 1st DZV11 in the system.
1502	000300	This is vector 'A' for the first DZV11 in the system.
1504	000017	This is the binary representation of what lines are to be tested.
1506	017470	This is the parameter location used in most of the tests. It indicates parameters of: RX ON, SPEED SELECT 17 (19.2K BAUD) EIGHT BITS PER CHAR, AND TWO STOP BITS. The user may alter the stop bits and the speed, but the remaining parameters should be left alone. This location is used to load the DZV11 Line Parameter Register for each line. The meaning of the bits set in this location is the same as the function of the related bits in the device Line Parameter Register.
1510	000000	This location will contain either all zeros indicating that internal loop was selected as mode of operation or it will contain 100000 indicating that "staggered mode" was selected or it will contain 000200 indicating that "external" was the mode selected.

The above is repeated for each DZV11 in the system. The table is filled by AUTO SIZING or by the manual parameter input program as described previously. Also if desired by user; the locations may be altered by hand to suit the specific configuration.

8.5 *** METHOD OF AUTO SIZING ***

8.5.1 FINDING THE CONTROL STATUS REGISTER.

The program will start at address 160000 and start 'REFERENCING' the address in the pointer. If a NON-EX MEMORY TRAP occures, the pointer (holding 160000) is updated by 10 and the above is repeated until address 163770 is reached. If a 'BUS REPLY' response was issued by the DZV11 (or any other device) (no nxm trap), "MASTER SCAN ENABLE" is attempted to be set and the TCR bits for all four lines are set. "TRDY" is then tested to be set and "MASTER SCAN ENABLE" is tested to be still set. The diagnostic will then check that at least one TCR bit is still set. If all of the above worked, this device is assumed to be a DZV11. If any of the above failed, updating of the pointer is done and the sequence is repeated.
NOTE: If the program does not find your DZV11, something is wrong and AUTO SIZING should not be done.

8.5.2 FINDING THE VECTOR

The vector area (address 300-776) is filled with the instruction IOT and '.+2' (next address). Bit14 and Bit5 (TX INTERUPT ENABLE AND MSTSCAN ENABLE) are set into the DZVCSR. All TCR bits are set, a delay occurs, and if no interupt occures (because of a bad DZV11) the program assumes vector address 300 and the problem should be fixed in the diagnostic. Once the problem is fixed, the program should be setup again to set the correct vector. If an interupt occurred, the address to which the DZV11 interupted to is picked up and reported as the vector. NOTE: if the vector reported is not the vector set up by you, there is a problem and AUTO SIZING should not be done.

8.5.3 PARAMETER ASSUMPTIONS.

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

1) ALL FOUR LINES ARE ASSUMED TO BE TESTED.
2) DEFAULT BAUD RATE IS SET TO 17 (19.2 KBAUD).
3) MODE OF OPERATION IS "INTERNAL MODE".

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

9.0 RUNNING THE DZV11 DIAGNOSTIC UNDER APT

9.1.1 THE APT INTERFACE

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

9.1.2 SETTING UP THE DIAGNOSTIC USING APT

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

$SWREG	-(1142)	used as the software switch register while running under APT.
$VECT1	-(1170)	used to specify the first vector address
$BASE	-(1174)	used to indicate bottom address of DZV11 under test
$DEVM	-(1176)	a bit map representing which DZV11's will be tested
$CDW1	-(1200)	used to indicate which lines to run on all DZV11's
$CDW2	-(1202)	used to indicate the default test mode. Set to 0 for internal testing, 200 for external loop back (H325 installed), or set to 100000 for staggered loop back testing (H329 installed).
$DDW0	-(1204)	each of the $DDW words describes the parameters (LPR) for a particular DZV11, going up to 16 DZV11's

9.1.3 RUNNING UNDER APT

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

NOTE

Be sure $BASE points to the first DZV11 before running

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

DOCUMENT
****
DVDZBA 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.

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

           51       MISCELLANEOUS DEFINITIONS

           63       GENERAL PURPOSE REGISTER DEFINITIONS

           75       PRIORITY LEVEL DEFINITIONS

           85       "SWITCH REGISTER" SWITCH DEFINITIONS

          113       DATA BIT DEFINITIONS (BIT00 TO BIT15)

          141       BASIC "CPU" TRAP VECTOR ADDRESSES

          358                               BITS 15-11=CPU TYPE
                                                 11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                                 11/70=06,PDQ=07,Q=10
                                            BIT 10=REAL TIME CLOCK
                                            BIT  9=FLOATING POINT PROCESSOR
                                            BIT  8=MEMORY MANAGEMENT

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

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

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

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

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

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

1072    THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
        AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
        AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
        THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
        SW14=1  LOOP ON TEST
        SW11=1  INHIBIT ITERATIONS
        CALL
                SCOPE           ;;SCOPE=IOT

1147    ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
        THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
        NOTE1:  $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
        NOTE2:  $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
        NOTE3:  $FILLC CONTAINS THE CHARACTER TO FILL AFTER.

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


1931    ROUTINE USED TO SET UP THE DIAGNOSTIC VIA APT.
        IF BIT7 IN THE ENVIRONMENT MODE ($ENVM) BYTE IS SET,
        THE PROGRAM WILL LOAD ITS PARAMETERS FROM THE ETABLE.

1963    ROUTINE USED TO "AUTO SIZE" THE DZV11
        CSR AND VECTOR.
        NOTE:   THE CSR MAY BE ANY WHERE IN THE FLOATING
                ADDRESS RANGE (160000:163770)
                AND THE VECTOR MAY BE ANY WHERE IN THE
                FLOATING VECTOR RANGE (300:770)


2071    ********** TEST 1 **********
        THIS TEST VERIFIES OVERRUN AND SILO ALARM
        ONE LINE AT A TIME  - BASED UPON VALID LINES
        AS EACH OF THE FIRST 16 CHARS ARE SENT; SILO ALARM IS
        TESTED TO BE CLEARED. ON THE 16TH CHAR THE PROGRAM THEN
        EXPECTS SILO ALARM TO SET. THEN THE ENTIRE
        SILO IS FILLED AND AN OVERRUN IS EXPECTED ON THE 65TH
        CHAR PULLED OUT OF THE SILO.
        ERROR PRINTOUTS WILL REPORT TRANSMITTING LINE NO.
        USING SWITCH NINE FOR THIS TEST SENDS 20. CHARACTERS
        ON DZV LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
        USED TO SCOPE SILO ALARM PULSES, ETC.
```

```
2192    ********************** TEST 2 ********************************
        THIS TEST THAT "SILO ENABLE" WILL INHIBIT
        RECEIVER INTERRUPTS AND THAT ON THE
        16TH CHAR THAT "SILO ALARM" WILL CAUSE AN
        INTERRUPT WITH "RIE" SET.
        THIS WILL DO ALL SELECTED LINES ONE AT A TIME.
        ERROR PRINTOUTS WILL REPORT TRANSMITTING LINE NO.

2264    ********************** TEST 3 ********************************
        THIS TEST RUNS ALL LINES FULL BORE
        BASED UPON QUALIFIED LINES
        ..THIS IS AN INTERRUPT TEST ON THE RECEIVER AND
        TRANSMITTER

2397    ********************** TEST 4 ********************************
        DZV11 RELATIVE TIMING TEST.
        EACH SELECTED LINE WILL IN TURN RUN 16. CHARS
        AT ALL BAUD RATES AND THEN THE HIGHEST BAUD
        WITH ALL CHAR LENGTHS. EACH NEW PARAMETER SHOULD
        DECREASE IN TIME FROM THE PREVIOUS PARAMETERS SELECTED.
        THE TIME IS CHECKED AGAINST THE LAST PARAMETER USED
         AND A LOWER TIME IS EXPECTED ON THE CURRENT PARAMETER.
        PARAMETERS ARE:
          EIGHT BITS/PER/CHAR - TWO STOP BITS AT
                50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000
                2400, 3600, 4800, 7200, 9600 BAUD.
          19.2 K BAUD - TWO STOP BITS AT
                SEVEN, SIX, FIVE BITS/PER/CHAR.
        AFTER EACH LINE HAS FINISHED ALL THE ABOVE PARAMETERS
        THE NEXT SELECTED LINE IS THEN TESTED.
        WHEN RUNNING UNDER THE APT MANUFACTURING SYSTEM
        THIS TEST IS ONLY RUN THE FIRST PASS

2491    ********************** TEST 5 ********************************
        THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
        THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
        THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
        YOU ARE IN "STAGGERED" MODE.
        40(8) CHARS ARE USED FOR THIS TEST.
        ALL SELECTED LINES WILL BE ENABLED AT THE SAME TIME.
        THIS TEST FIRST CHECKS EVEN PARITY FOR ODD LINES AND
        ODD PARITY FOR EVEN LINES, THEN IT CHECKS THE REVERSE.
```

```
                        .TITLE  MD-11-DVDZB-A
                        ;*COPYRIGHT (C) 1977
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*
000001                  $TN=1
                                ;STARTING PROCEDURE
                                ;LOAD PROGRAM
                                ;LOAD ADDRESS 000200
                                ;PRESS START
                                ;PROGRAM WILL TYPE
                                ;"MAINDEC-11-DVDZBA/<200>/FOUR LINE ASYNC MUX TESTS, PART 2 OF 2"
                                ;PROGRAM WILL TYPE "RUNNING" TO INDICATE THAT TESTING HAS STARTED
                                ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
                                ;AND THEN RESUME TESTING

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


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

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1120 ***
001120                  STACK=  1120
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
```

```
                        .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
100000                  DVALID=BIT15    ;DATA VALID

                                ;DZVRBUF WORD DEFINITIONS
                                ;----------------------

000000                  RL0=0           ;RECEIVER LINE 0
000400                  RL1=BIT8        ;RECEIVER LINE 1
001000                  RL2=BIT9        ;RECEIVER LINE 2
001400                  RL3=BIT9!BIT8   ;RECEIVER LINE 3

                                ;DZVLPR WORD DEFINITIONS
                                ;----------------------

000000                  LP0=0           ;LINE PARAMETER 0
000001                  LP1=BIT0        ;LINE PARAMETER 1
000002                  LP2=BIT1        ;LINE PARAMETER 2
000003                  LP3=BIT1!BIT0   ;LINE PARAMETER 3

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

000100                  PARITY=BIT6             ;PARITY ENABLED
```

```
                        ODDPAR=BIT7             ;ODD PARITY ENABLED
                        ONESTOP=0               ;ONE STOP BIT ENABLED
                        TWOSTOP=BIT5            ;TWO STOP BITS ENABLED
                        EVEPAR=0                ;EVEN PARITY ENABLED
                        RCVON=BIT12             ;ENABLE RECEIVER (RECEIVER ON)

                        S50=0                   ;SPEED 50 BAUD
                        S75=BIT8                ;SPEED 75 BAUD
                        S110=BIT9               ;SPEED 110 BAUD
                        S134=BIT9!BIT8          ;SPEED 134.5 BAUD
                        S150=BIT10              ;SPEED 150 BAUD
                        S300=BIT10!BIT8         ;SPEED 300 BAUD
                        S600=BIT10!BIT9         ;SPEED 600 BAUD
                        S1200=BIT10!BIT9!BIT8   ;SPEED 1200 BAUD
                        S1800=BIT11             ;SPEED 1800 BAUD
                        S2000=BIT11!BIT8        ;SPEED 2000 BAUD
                        S2400=BIT11!BIT9        ;SPEED 2400 BAUD
                        S3600=BIT11!BIT9!BIT8   ;SPEED 3600 BAUD
                        S4800=BIT11!BIT10       ;SPEED 4800 BAUD
                        S7200=BIT11!BIT10!BIT8  ;SPEED 7200 BAUD
                        S9600=BIT11!BIT10!BIT9  ;SPEED 9600 BAUD
                        S19200=BIT11!BIT10!BIT9!BIT8    ;SPEED 19200 BAUD

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

                ;DZVMSR BIT DEFINITIONS
                ;----------------------
000001          RING0=BIT0              ;RING INDICATED ON LINE 0
000002          RING1=BIT1              ;RING INDICATED ON LINE 1
000004          RING2=BIT2              ;RING INDICATED ON LINE 2
000010          RING3=BIT3              ;RING INDICATED ON LINE 3
000400          CO0=BIT8                ;CARRIER PRESENT ON LINE 0
001000          CO1=BIT9                ;CARRIER PRESENT ON LINE 1
002000          CO2=BIT10               ;CARRIER PRESENT ON LINE 2
004000          CO3=BIT11               ;CARRIER PRESENT ON LINE 3

                ;DZVTDR BIT DEFINITIONS
                ;----------------------

000400          BRK0=BIT8               ;BREAK FOR LINE 0
001000          BRK1=BIT9               ;BREAK FOR LINE 1
002000          BRK2=BIT10              ;BREAK FOR LINE 2
004000          BRK3=BIT11              ;BREAK FOR LINE 3
```

```
277
278                             ;TABLE OF LOOP AROUND FUNCTIONS (H325)
279                             ;
280                             ;       -----------------
281                             ;       I               ↑
282                             ;       V               ↑
283                             ;       REC             TRANS
284                             ;       DATA            DATA
285                             ;
286                             ;       -----------------
287                             ;       I               ↑
288                             ;       V               ↑
289                             ;       CO              RTS
290                             ;
291                             ;       -----------------
292                             ;       I               ↑
293                             ;       V               ↑
294                             ;       RING            DTR
```

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

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

            000020      .=20
000020  004300                  .SCOPE                  ;SCOPE LOOP HANDLER
000022  000200                  MASK                    ;HANDLE AT PRIORITY 7
000024  007236                  $PWRDN                  ;POWER FAIL HANDLER
000026  000340                  340                     ;SERVICE AT PRIORITY LEVEL 7
000030  006344                  $ERROR                  ;ERROR HANDLER
000032  000340                  340                     ;SERVICE AT PRIORITY LEVEL 7
000034  006136                  .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                  340                     ;SERVICE AT PRIORITY LEVEL 7
                        .SBTTL  ACT11 HOOKS

                        ;;*****************************************************************************
                        ;HOOKS REQUIRED BY ACT11
            000040              $SVPC=.                 ;SAVE PC
            000046              .=46
000046  004234                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
            000052              .=52
000052  000000                  .WORD   0               ;;2)SET LOC.52 TO ZERO
            000040              .=$SVPC                 ;; RESTORE PC

            000174      .=174
000174  000000          DISPREG:0                       ;SOFTWARE DISPLAY REGISTER FOR SWITCHLESS 11S
000176  000000          SWREG:  0                       ;SOFTARE SWITCH REGISTER FOR SWITCHLESS 11S
            000200      .=200
000200  000137  002116          JMP     .START          ;GO TO START OF PROGRAM


            001000      .=1000
001000  005200  040515  047111  MTITLE: .ASCIZ  <200><12>/MAINDEC-11-DVDZBA/<200>/FOUR LINE ASYNC MUX TESTS, PART 2 OF 2
(2)
```

```
                        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  000300                                  $VECT1: .WORD   AVECT1          ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001172  000000                                  $VECT2: .WORD   AVECT2          ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001174  160010                                  $BASE:  .WORD   ABASE           ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001176  000001                                  $DEVM:  .WORD   ADEVM           ;;DEVICE MAP
001200  000017                                  $CDW1:  .WORD   ACDW1           ;;CONTROLLER DESCRIPTION WORD#1
001202  000000                                  $CDW2:  .WORD   ACDW2           ;;CONTROLLER DESCRIPTION WORD#2
001204  017470                                  $DDW0:  .WORD   ADDW0           ;;DEVICE DESCRIPTOR WORD#0
001206  017470                                  $DDW1:  .WORD   ADDW1           ;;DEVICE DESCRIPTOR WORD#1
001210  017470                                  $DDW2:  .WORD   ADDW2           ;;DEVICE DESCRIPTOR WORD#2
001212  017470                                  $DDW3:  .WORD   ADDW3           ;;DEVICE DESCRIPTOR WORD#3
001214  017470                                  $DDW4:  .WORD   ADDW4           ;;DEVICE DESCRIPTOR WORD#4
001216  017470                                  $DDW5:  .WORD   ADDW5           ;;DEVICE DESCRIPTOR WORD#5
```

```
  393  001220  017470                SDDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
  394  001222  017470                SDDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
  395  001224  017470                SDDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
  396  001226  017470                SDDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
  397  001230  017470                SDDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
  398  001232  017470                SDDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
  399  001234  017470                SDDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
  400  001236  017470                SDDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
  401  001240  017470                SDDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
  402  001242  017470                SDDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
  403
  404
  405  001244                        SETEND:
  406
```

```
                                .SBTTL  COMMON TAGS

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

001244                  SCMTAG:                         ;;START OF COMMON TAGS
001244  000000                  .WORD   0
001246  000             STSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001247  000             SERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001250  000000          SICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001252  000000          SLPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001254  000000          SLPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001256  000000          SERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001260  000             SITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001261  001             SERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001262  000000          SERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001264  000000          SGDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001266  000000          SBDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001270  000000          SGDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001272  000000          SBDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001274  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001276  000000                  .WORD   0
001300  000             SAUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001301  000             SINTAG: .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          STKS:   177560                  ;;TTY KBD STATUS
001312  177562          STKB:   177562                  ;;TTY KBD BUFFER
001314  177564          STPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001316  177566          STPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001320  000             SNULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001321  002             SFILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001322  012             SFILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001323  000             STPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001324  000000          SREGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001326  000000          SREG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001330  000000          SREG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001332  000000          SREG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001334  000000          SREG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001336  000000          SREG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001340  000000          SREG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001342  000000          STMP0:  .WORD   0               ;;USER DEFINED
001344  000000          STMP1:  .WORD   0               ;;USER DEFINED
001346  000000          STMP2:  .WORD   0               ;;USER DEFINED
001350  000000          STMP3:  .WORD   0               ;;USER DEFINED
001352  000000          STMP4:  .WORD   0               ;;USER DEFINED
001354  000000          STIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001356  077             SQUES:  .ASCII  /?/             ;;QUESTION MARK
001357  015             SCRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001360  000012          SLF:    .ASCIZ  <12>            ;;LINE FEED
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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


001362                          $ERRTB:

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

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

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

001366  000017                  LINE:   17                      ;DEFAULT ALL FOUR LINES RUNNING
001370  017470                  PAR:    17470                   ;PARAMETERS: 8 BITS/CHAR,2 STOP BITS,19200 BAUD,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  000132       $TSTM:  .WORD   90.        ;;RUN TIM OF LONGEST TEST
001454  000137       $PASTM: .WORD   95.        ;;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
```

```
                        PAR1:   .BLKW   1       ;PARAMETERS
                        MANT1:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

                        DZCR2:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 2
                        DZVC2:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 2
                        LINE2:  .BLKW   1       ;ALL LINES SELECTED
                        PAR2:   .BLKW   1       ;PARAMETERS
                        MANT2:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

                        DZCR3:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 3
                        DZVC3:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 3
                        LINE3:  .BLKW   1       ;ALL LINES SELECTED
                        PAR3:   .BLKW   1       ;PARAMETERS
                        MANT3:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

                        DZCR4:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 4
                        DZVC4:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 4
                        LINE4:  .BLKW   1       ;ALL LINES SELECTED
                        PAR4:   .BLKW   1       ;PARAMETERS
                        MANT4:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

                        DZCR5:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 5
                        DZVC5:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 5
                        LINE5:  .BLKW   1       ;ALL LINES SELECTED
                        PAR5:   .BLKW   1       ;PARAMETERS
                        MANT5:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

                        DZCR6:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 6
                        DZVC6:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 6
                        LINE6:  .BLKW   1       ;ALL LINES SELECTED
                        PAR6:   .BLKW   1       ;PARAMETERS
                        MANT6:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

                        DZCR7:  .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 7
                        DZVC7:  .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 7
                        LINE7:  .BLKW   1       ;ALL LINES SELECTED
                        PAR7:   .BLKW   1       ;PARAMETERS
                        MANT7:  .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

                        DZCR10: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 10
                        DZVC10: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 10
                        LINE10: .BLKW   1       ;ALL LINES SELECTED
                        PAR10:  .BLKW   1       ;PARAMETERS
                        MANT10: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

                        DZCR11: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 11
                        DZVC11: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 11
                        LINE11: .BLKW   1       ;ALL LINES SELECTED
                        PAR11:  .BLKW   1       ;PARAMETERS
                        MANT11: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE

                        DZCR12: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 12
                        DZVC12: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 12
                        LINE12: .BLKW   1       ;ALL LINES SELECTED
                        PAR12:  .BLKW   1       ;PARAMETERS
                        MANT12: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
```

```
613
614  001656  000001          DZCR13: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 13
615  001660  000001          DZVC13: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 13
616  001662  000001          LINE13: .BLKW   1       ;ALL LINES SELECTED
617  001664  000001          PAR13:  .BLKW   1       ;PARAMETERS
618  001666  000001          MANT13: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
619
620  001670  000001          DZCR14: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 14
621  001672  000001          DZVC14: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 14
622  001674  000001          LINE14: .BLKW   1       ;ALL LINES SELECTED
623  001676  000001          PAR14:  .BLKW   1       ;PARAMETERS
624  001700  000001          MANT14: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
625
626  001702  000001          DZCR15: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 15
627  001704  000001          DZVC15: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 15
628  001706  000001          LINE15: .BLKW   1       ;ALL LINES SELECTED
629  001710  000001          PAR15:  .BLKW   1       ;PARAMETERS
630  001712  000001          MANT15: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
631
632  001714  000001          DZCR16: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 16
633  001716  000001          DZVC16: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 16
634  001720  000001          LINE16: .BLKW   1       ;ALL LINES SELECTED
635  001722  000001          PAR16:  .BLKW   1       ;PARAMETERS
636  001724  000001          MANT16: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
637
638  001726  000001          DZCR17: .BLKW   1       ;CONTROL STATUS REGISTER FOR DZV11 NUMBER 17
639  001730  000001          DZVC17: .BLKW   1       ;RECEIVER AND BASE VECTOR  FOR DZV11 NUMBER 17
640  001732  000001          LINE17: .BLKW   1       ;ALL LINES SELECTED
641  001734  000001          PAR17:  .BLKW   1       ;PARAMETERS
642  001736  000001          MANT17: .BLKW   1       ;MAINTENANCE MODE FOR THIS DEVICE
643
644  001740  177777          DZV.END:        177777
```

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

                        ;:*****************************************************************
                        ;-----------------------------------------------------------------
001742                  .TRPTAB:
104400                  ADVANCE=TRAP+0          ;CALL TO ADVANCE TO NEXT TEST( OR SCOPE THIS ONE)
001742  006232                  .ADVANCE
104401                  SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
001744  004544                  .SCOP1
104402                  TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
001746  004570                  .TYPE
104403                  INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
001750  005336                  .INSTR
104404                  INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
001752  005442                  .INSTER
104405                  PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
001754  005462                  .PARAM
104406                  SETFLG=TRAP+6           ;CALL TO  SET FLAG ROUTINE
001756  010074                  .SETFLG
104407                  SAVO5=TRAP+7            ;CALL TO REGISTER SAVE ROUTINE
001760  005662                  .SAVO5
104410                  RESO5=TRAP+10           ;CALL TO REGISTER RESTORE ROUTINE
001762  005722                  .RESO5
104411                  CONVRT=TRAP+11          ;CALL TO DATA OUTPUT ROUTINE
001764  005754                  .CONVRT
104412                  CNVRT=TRAP+12           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
001766  005760                  .CNVRT
104413                  DEVICE.CLR=TRAP+13              ;CALL TO ISSUE A DEVICE CLEAR
001770  006160                  .DEVICE.CLR
104414                  DELAY=TRAP+14           ;CALL TO DELAY FOR FAST CPU'S
001772  006212                  .DELAY
104415                  PARMD=TRAP+15           ;CONVERT DECIMAL STRING TO OCTAL
001774  011142                  .PARMD
104416                  PAWCH=TRAP+16           ;SET FLAG    ECHO OR  CABLE
001776  010214                  .PAWCH
104417                  DCLASM=TRAP+17          ;CLEAR DEVICE, SET MAINT. BIT IF I MODE
002000  006200                  .DCLASM
104420                  SHIFT=TRAP+20           ;CALL TO ROTATE LINE POINTER
002002  006244                  .SHIFT
104421                  LPRSET=TRAP+21          ;CALL TO SET UP LPR DEVICE REGISTER
002004  006262                  .LPRSET
104422                  BUFSET=TRAP+22          ;CALL TO ZERO BUFFER AREA
002006  006322                  .BUFSET

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

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

002010  160040                 DZVCSR:  160040  ;R/W
002012  160041                 HDZVCSR: 160041  ;R/W
002014  160042                 DZVRBUF: 160042  ;READ ONLY
002016  160043                 HDZVRBUF:160043  ;READ ONLY
002020  160042                 DZVLPR:  160042  ;WRITE ONLY
002022  160043                 HDZVLPR: 160043  ;WRITE 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
```

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

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

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

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

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

                                        ;GET THE BASE ADDRESS OF THE DZV11'S

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

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

                                        ;GET THE BASE VECTOR ADDRESS

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

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

                                        ;GET THE NUMBER OF DZV11'S RUNNING

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

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

```
                                50$:    MOV     (R0)+,(R1)      ;ADDRESS
                                        ADD     #10,(R1)+       ;POINT TO THE NEXT DZV11 ADDRESS VALUE
                                        MOV     (R0)+,(R1)      ;VECTOR
                                        ADD     #10,(R1)+       ;POINT TO THE NEXT VECTOR VALUE
                                        MOV     (R0)+,(R1)+     ;LINES
                                        MOV     (R0)+,(R1)+     ;PARAMETERS
                                        MOV     (R0)+,(R1)+     ;MAINTENANCE MODE
                                        BR      45$
                                55$:    BIT     #SW03,@SWR      ;ASK PARAMETERS ?
                                        BNE     60$             ;IF NO, GO DO AUTO SIZING
                                        JMP     100$            ;GO SET UP FOR AUTO SIZING
                                60$:    JSR     PC,65$          ;GO ASK PARAMETERS
                                        DECB    INIFLG          ;INSURE NO AUTO SIZE IF QUESTIONS ANSWERED
                                        JMP     105$            ;GO TO THE NEXT BLOCK

                                ;GET THE ACTIVE LINES PARAMETER

                                65$:
                                        INSTR                   ;CALL THE STRING INPUT ROUTINE
                                        93$                     ;POINTER TO MESSAGE TO BE PRINTED
                                        PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
                                        1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                                        17                      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                                        LINE0                   ;POINTER TO MAP LOCATION TO BE FILLED
                                        .BYTE   360             ;MASK OF INVALID BITS FOR THIS PARAMETER
                                        .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
                                        CLRB    HDRFLG          ;MAKE SURE THE CHANGES ARE PRINTED

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

                                        TST     MANTO           ;IS STAGGERED THE MODE OF OPERATION?
                                        BPL     85$             ;IF NOT, SKIP THIS SEGMENT
                                        MOV     LINE0,R3        ;GET A SCRATCH COPY OF THE ACTIVE LINES
                                70$:    ROR     R3              ;GET A LINE SELECTION BIT(EVEN NUMBER LINE)
                                        BCS     80$             ;IF IT IS SELECTED, CHECK TO SEE IF THE NEXT IS TOO
                                        BEQ     85$             ;IF ALL HAVE BEEN CHECKED, CONTINUE PROCESSING
                                        ASR     R3              ;IF IT IS 0,CHECK TO SEE IF THE NEXT IS TOO
                                        BCC     70$             ;IF THIS ONE'S 0 TOO, GO CHECK THE NEXT PAIR
                                75$:    TYPE    ,$QUES          ;THIS IS AN INCORRECT PARAMETER
                                        TYPE    ,MBADLN         ;LET THE USER KNOW ABOUT IT
                                        BR      65$             ;GO GET THE CORRECT PARAMETER
                                80$:    BEQ     75$             ;IF ANOTHER FLAG ISN'T SET, THERE'S AN ERROR
                                        ASR     R3              ;GET THE NEXT FLAG
                                        BCC     75$             ;IF IT ISN'T SET, THERE'S AN ERROR
                                        CLC                     ;INITIALIZE THE "C" BIT FOR TESTING OF THE NEXT PAIR
                                        BR      70$             ;GO TEST THE NEXT PAIR OF FLAGS

                                ;GET THE LINE PARAMETER REGISTER ARGUMENT

                                85$:
                                        INSTR                   ;CALL THE STRING INPUT ROUTINE
                                        94$                     ;POINTER TO MESSAGE TO BE PRINTED
                                        PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
                                        0                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                                        17                      ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
```

```
                                        PAR0                    ;POINTER TO MAP LOCATION TO BE FILLED
                                        .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
                                        .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
002732 001506                           MOV     #LINE0,R2       ;POINT TO THE LINE SELECTION PARAMETER
002734 000                              MOV     #PAR0,R3        ;POINT TO THE CHOSEN PARAMETERS
002735 001                              MOV     (R3),R4         ;USE BAUD RATE AS AN INDEX IN DELAY TABLE
002736 012702 001504                    ASL     R4              ;ALIGN INDEX ON WORD BOUNDARY
002742 012703 001506                    MOV     DLYTBL(R4),DLYCNT ;SET THE DELAY COUNT FOR THIS BAUD RATE
002746 011304                           SWAB    (R3)            ;PLACE IN HIGH BYTE
002750 006304                           BIS     #10070,(R3)     ;PLACE EXTRA PARAMETERS INTO LOC
002752 016437 017166 006230             MOV     (R2),12(R2)     ;LOAD THE LINES
002760 000313                           MOV     (R3),12(R3)     ;LOAD THE PARAMETERS
002762 052713 010070                    ADD     #12,R2          ;POINT TO THE NEXT SET
002766 011262 000012            90$:    ADD     #12,R3          ;.. OF BOTH PARAMETERS
002772 011363 000012                    CMP     R3,#PAR17       ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
002776 062702 000012                    BNE     90$             ;IF NOT, GO LOAD SOME MORE PARAMETERS
003002 062703 000012                    RTS     PC              ;RETURN TO CALLING BLOCK
003006 020327 001734
003012 001365
003014 000207
003016 030600 052123 041440     91$:    .ASCIZ  <200>/1ST CSR ADDRESS (160000:163770):  /
(1) 003062 030600 052123 053040 92$:    .ASCIZ  <200>/1ST VECTOR ADDRESS (300:770):  /
(1) 003123 000200 044514 042516 93$:    .ASCIZ  <200>/LINES ACTIVE BY BIT <IN OCTAL>(001:17):  /
(1) 003176 042200 043105 052501 94$:    .ASCIZ  <200>/DEFAULT BAUD RATE <IN OCTAL>(00:17):  /
(1) 003246 021600 047440 020106 95$:    .ASCIZ  <200>/# OF DZV11'S <IN OCTAL> (1:20):  /
(1) 003311 000200 040515 047111 96$:    .ASCII  <200>/MAINTENANCE MODE/
(1) 003332 020200 042533 052130         .ASCII  <200>/ [EXTERNAL  <H325>      (E)]/
(1) 003366 020200 044533 052116         .ASCII  <200>/ [INTERNAL  <DZVCSR03=1>(I)]/
(1) 003423 000200 055440 052123         .ASCIZ  <200>/ [STAGGERED <H329>      (S)]: /
(1) 003462 042600 052116 051105 97$:    .ASCIZ  <200>/ENTER DELAY PARAMETER: /
(1) 003514                              .EVEN
(1) 003514                      100$:
003514 122737 000377 001422             CMPB    #377,INIFLG     ;ONLY DO AUTO SIZE ON 1ST START
003522 001006                           BNE     105$            ;
003524 032777 000200 175552             BIT     #BIT7,@SWR      ;BIT7=1??
003532 001002                           BNE     105$            ;BR IF NO AUTO SIZE
003534 004737 011464                    JSR     PC,AUTO.SIZE    ;GO DO THE AUTO SIZE
003540 105737 001423            105$:   TSTB    HDRFLG          ;HAS THE TABLE BEEN TYPED YET?
003544 001021                           BNE     120$            ;IF SO, DON'T TYPE IT AGAIN
003546 105337 001423                    DECB    HDRFLG          ;INDICATE THAT THE TABLE WILL BE TYPED
003552 104402 007772                    TYPE    ,XHEAD          ;TYPE MAP HEADER
003556 012700 001500                    MOV     #DZV.MAP,R0     ;SET POINTER
003562 010037 001344            110$:   MOV     R0,$TMP1        ;POINT TO THE MAP LOCATION
003566 012037 001346                    MOV     (R0)+,$TMP2     ;SET DATA
003572 022737 177777 001346             CMP     #-1,$TMP2       ;END OF LIST?
003600 001403                           BEQ     120$            ;BR IF YES
003602 104411                   115$:   CONVRT                  ;CALL THE OCTAL TO ASCII CONVERSION ROUTINE
003604 010062                           XSTAT0                  ;CONVERT THE DATA AT THIS ADDRESS
003606 000765                           BR      110$            ;GO PRINT THE NEXT PARAMETER
003610 013737 001410 001406     120$:   MOV     SAVACTV,DZVACTV ;COPY BIT MAP OF ACTIVE DEVICES
003616 113737 001414 001416             MOVB    DZVNUM,SAVNO    ;COPY NO. OF DEVICES IN THE SYSTEM
003624 032777 000100 175452             BIT     #SW06,@SWR      ;DESELECT SPECIFIC DEVICES??
003632 001431                           BEQ     135$            ;BR IF NO.
003634                          121$:
003634 104403                           INSTR                   ;CALL THE STRING INPUT ROUTINE
003636 007710                           MNEW                    ;POINTER TO MESSAGE TO BE PRINTED
003640 104405                           PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
003642 000001                           1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
003644 177777                           177777                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
```

```
 954  003646  001406                          DZVACTV                 ;POINTER TO MAP LOCATION TO BE FILLED
 955  003650     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
 956  003651     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
 957  003652  023737  001406  001410          CMP     DZVACTV,SAVACTV ;IS VALUE VALID?
 958  003660  101403                          BLOS    122$            ;IF YES BRANCH
 959  003662  104402  007562                  TYPE    ,MERR3          ;IF NOT TYPE ERROR
 960  003666  000762                          BR      121$            ;THEN REASK QUESTION
 961  003670  105037  001416          122$:   CLRB    SAVNO           ;INITIALIZE NO. OF ACTIVE DEVICES
 962  003674  013737  001406  001344          MOV     DZVACTV,$TMP1   ;COPY BIT MAP OF ACTIVE DEVICES
 963  003702  006237  001344          126$:   ASR     $TMP1           ;ROTATE OUT AN ACTIVE BIT
 964  003706  103002                          BCC     127$            ;IF NOT ACTIVE SKIP RECORDING IT
 965  003710  105237  001416                  INCB    SAVNO           ;INCREMENT NO. OF ACTIVE DEVICES
 966  003714  001372                  127$:   BNE     126$            ;IF NOT DONE GO CONTINUE
 967  003716  032777  000020  175360  135$:   BIT     #SW04,@SWR      ;CHECK TO SEE IF DELAY COUNT CHANGES
 968  003724  001407                          BEQ     140$            ;IF NOT, GO CLEAR VECTOR AREA
 969  003726  1044J3                          INSTR                   ;CALL THE STRING INPUT ROUTINE
 970  003730  003462                          97$                     ;POINTER TO MESSAGE TO BE PRINTED
 971  003732  104405                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
 972  003734  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
 973  003736  177777                          177777                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
 974  003740  006230                          DLYCNT                  ;POINTER TO MAP LOCATION TO BE FILLED
 975  003742     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
 976  003743     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
 977  003744  012700  000300          140$:   MOV     #300,R0         ;PREPARE TO CLEAR THE FLOATING
 978  003750  012701  000302                  MOV     #302,R1         ;VECTOR AREA. 300-776
 979  003754  010120                  145$:   MOV     R1,(R0)+        ;START PUTTING "PC+2 - HALT"
 980  003756  005021                          CLR     (R1)+           ;IN VECTOR AREA.
 981  003760  022021                          CMP     (R0)+,(R1)+     ;POP POINTERS
 982  003762  022700  001000                  CMP     #1000,R0        ;ALL DONE??
 983  003766  001372                          BNE     145$            ;BR IF NO.
 984
 985                                  ;TEST START AND RESTART
 986                                  ;----------------------
 987
 988  003770  012706  001120          .BEGIN: MOV     #STACK,SP       ;SET UP STACK
 989  003774  106427  000200                  MTPS    #MASK           ;LOCK OUT INTERRUPTS
 990  004000  005737  000042                  TST     @#42            ;IS PROGRAM UNDER MONITOR CONTROL
 991  004004  001015                          BNE     2$              ;BR IF YES
 992  004006  032777  000004  175270          BIT     #BIT2,@SWR      ;CHECK FOR LOCK ON TEST
 993  004014  001406                          BEQ     1$              ;BR IF NO LOCK DESIRED.
 994  004016  104402  007606                  TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
 995  004022  012737  000240  004312          MOV     #NOP,TTST       ;ADJUST SCOPE ROUTINE.
 996  004030  000403                          BR      2$              ;CONTINUE ALONG.
 997  004032  013737  004540  004312  1$:     MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
 998  004040  012737  010436  001252  2$:     MOV     #CYCLE,$LPADR   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
 999  004046  113737  001416  001415          MOVB    SAVNO,SAVNUM    ;COPY NO. OF ACTIVE DEVICES
1000  004054  104402  007477                  TYPE    ,MR             ;TYPE "RUNNING"
1001  004060  000177  175166                  JMP     @$LPADR         ;START TESTING
```

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

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

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

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

```
                                        DZVRIV
                                XPASS:  1
                                        .BYTE   6,2
                                        $PASS
                                XERR:   1
                                        .BYTE   6,2
                                        $ERTTL

                                        ;SCOPE LOOP AND ITERATION HANDLER
                                        ;-------------------------------

                                .SBTTL  SCOPE HANDLER ROUTINE

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

001300                          $SCOPE:
001300  005037  001262          .SCOPE: CLR     $ERRPC          ;CLEAR LAST ERROR PC.
001304  022716  012172                  CMP     #TST1+2,(SP)    ;IS THIS THE SCOPE AT THE BEGINNING OF TST1?
001310  001413                          BEQ     $XTSTR          ;IF SO, DON'T LOOP ON IT
001312  000406                  TTST:   BR      1$              ;GOTO 1$    (IF LOCK SW02=1; THIS LOC =240)
001314  105777  174770                  TSTB    @$TKS           ;KEYBOARD DONE?
001320  100067                          BPL     $OVER           ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
001322  017766  174764  177776          MOV     @$TKB,-2(SP)    ;CLEAR DONE BIT
001330  032777  040000  174746  1$:     BIT     #BIT14,@$WR     ;;LOOP ON PRESENT TEST?
001336  001060                          BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
001340  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
001342  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
001346  012737  004366  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
001354  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
001360  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
001364  000436                          BR      $SVLAD          ;;GO TO THE NEXT TEST
001366  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
001370  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
001374  000441                          BR      $OVER           ;;LOOP ON THE PRESENT TEST
001376                          6$:;#####END OF CODE FOR THE XOR TESTER#####
001376  105737  001247          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
001402  001404                          BEQ     3$              ;;BR IF NO
001404  105037  001247          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
001410  005037  001354                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
001414  032777  004000  174662  3$:     BIT     #BIT11,@$WR     ;;INHIBIT ITERATIONS?
001422  001011                          BNE     1$              ;;BR IF YES
001424  005737  001126                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
001430  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
001432  005237  001250                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
001436  023737  001354  001250          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
001444  002015                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
```

```
1114  004446  012737  000001  001250  1$:     MOV     #1,$ICNT          ;;REINITIALIZE THE ITERATION COUNTER
1115  004454  013737  004542  001354          MOV     $MXCNT,$TIMES     ;;SET NUMBER OF ITERATIONS TO DO
1116  004462  105237  001246          $SVLAD: INCB    $TSTNM            ;;COUNT TEST NUMBERS
1117  004466  113737  001246  001124          MOVB    $TSTNM,$TESTN     ;;SET TEST NUMBER IN APT MAILBOX
1118  004474  011637  001252                  MOV     (SP),$LPADR       ;;SAVE SCOPE LOOP ADDRESS
1119  004500  013777  001246  174600  $OVER:  MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER
1120  004506  013716  001252                  MOV     $LPADR,(SP)       ;;FUDGE RETURN ADDRESS
1121  004512  004737  006772                  JSR     PC,SERV.G         ;FIND OUT IF ↑G WAS TYPED
1122  004516  105037  001424                  CLRB    MNTFLG            ;CLEAR THE MAINTENANCE BIT SETTER AFTER EACH TEST
1123  004522  005737  001372                  TST     MODE              ;HAS THE MODE BEEN CHANGED?
1124  004526  001003                          BNE     4$                ;IF NOT INTERNAL , GO DO A TEST
1125  004530  112737  000010  001424          MOVB    #MAINT,MNTFLG     ;IF INTERNAL MODE NOW, SET THE MAINTENANCE BIT
1126  004536  000002          4$:     RTI                               ;GO DO THE TEST
1127  004540  000406          BRW:    406
1128  004542  000005          $MXCNT: 5                                 ;;MAX. NUMBER OF ITERATIONS
1129
1130                                  ;CHECK FOR FREEZE ON CURRENT DATA
1131                                  ;-------------------------------
1132
1133  004544  032777  001000  174532  .SCOP1: BIT     #SW09,@SWR        ;IS SW09=1(SET)?
1134  004552  001405                          BEQ     1$                ;BR IF NOT SET.
1135  004554  005737  001364                  TST     LOCK              ;IS THERE A TIGHT LOOP SPECIFIED?
1136  004560  001402                          BEQ     1$                ;IF NO, RETURN
1137  004562  013716  001364                  MOV     LOCK,(SP)         ;IF YES, GOTO THE ADDRESS IN LOCK.
1138  004566  000002          1$:     RTI                               ;GO BACK.
1139
1140  004570  032777  010000  174506  .TYPE:  BIT     #SW12,@SWR        ;INHIBIT ALL PRINTOUT??
1141  004576  001403                          BEQ     $TYPE             ;IF NOT, GO TYPE
1142  004600  062716  000002                  ADD     #2,(SP)           ;SKIP OVER MESSAGE POINTER
1143  004604  000002                          RTI                       ;RETURN TO WHERE PROCEDURE WAS INVOKED
1144                                  .SBTTL  TYPE ROUTINE
1145
1146                                  ;;*****************************************************************
1147                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
1148                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
1149                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
1150                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
1151                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
1152                                  ;*
1153                                  ;*CALL:
1154                                  ;*1) USING A TRAP INSTRUCTION
1155                                  ;*      TYPE    ,MESADR           ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
1156                                  ;*OR
1157                                  ;*      TYPE
1158                                  ;*      MESADR
1159                                  ;*
1160
1161  004606  105737  001323          $TYPE:  TSTB    $TPFLG            ;;IS THERE A TERMINAL?
1162  004612  100002                          BPL     1$                ;;BR IF YES
1163  004614  000000                          HALT                      ;;HALT HERE IF NO TERMINAL
1164  004616  000430                          BR      3$                ;;LEAVE
1165  004620  010046          1$:     MOV     R0,-(SP)                  ;;SAVE R0
1166  004622  017600  000002                  MOV     @2(SP),R0         ;;GET ADDRESS OF ASCIZ STRING
1167  004626  122737  000001  001140          CMPB    #APTENV,$ENV      ;;RUNNING IN APT MODE
1168  004634  001011                          BNE     62$               ;;NO,GO CHECK FOR APT CONSOLE
1169  004636  132737  000100  001141          BITB    #APTSPOOL,$ENVM   ;;SPOOL MESSAGE TO APT
```

```
1170  004644  001405                        BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
1171  004646  010037  004656                MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
1172  004652  004737  005076                JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
1173  004656  000000              61$:      .WORD   0               ;;MESSAGE ADDRESS
1174  004660  132737  000040  001141  62$:  BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
1175  004666  001003                        BNE     60$             ;;YES,SKIP TYPE OUT
1176  004670  112046              2$:       MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
1177  004672  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
1178  004674  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
1179  004676  012600              60$:      MOV     (SP)+,R0        ;;RESTORE R0
1180  004700  062716  000002        3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
1181  004704  000002                        RTI                     ;;RETURN
1182  004706  122716  000011        4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
1183  004712  001430                        BEQ     8$
1184  004714  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
1185  004720  001006                        BNE     5$
1186  004722  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
1187  004724  104402                        TYPE                    ;;TYPE A CR AND LF
1188  004726  001357                        $CRLF
1189  004730  105037  005064                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
1190  004734  000755                        BR      2$              ;;GET NEXT CHARACTER
1191  004736  004737  005020        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
1192  004742  123726  001322        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
1193  004746  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
1194  004750  013746  001320                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
1195                                                                ;;AND THE NULL CHAR.
1196  004754  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
1197  004760  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
1198  004762  004737  005020                JSR     PC,$TYPEC       ;;GO TYPE A NULL
1199  004766  105337  005064                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
1200  004772  000770                        BR      7$              ;;LOOP
1201
1202                                        ;HORIZONTAL TAB PROCESSOR
1203
1204  004774  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
1205  005000  004737  005020        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
1206  005004  132737  000007  005064        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
1207  005012  001372                        BNE     9$              ;;TAB STOP
1208  005014  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
1209  005016  000724                        BR      2$              ;;GET NEXT CHARACTER
1210  005020  105777  174270      $TYPEC:   TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
1211  005024  100375                        BPL     $TYPEC
1212  005026  116677  000002  174262        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
1213  005034  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
1214  005042  001003                        BNE     1$              ;;BRANCH IF NO
1215  005044  105037  005064                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
1216  005050  000406                        BR      $TYPEX          ;;EXIT
1217  005052  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
1218  005060  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
1219  005062  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
1220  005064  000000              $CHARCNT: .WORD   0               ;;CHARACTER COUNT STORAGE
1221  005066  000207              $TYPEX:   RTS     PC
1222
1223                                        .SBTTL  APT COMMUNICATIONS ROUTINE
1224
1225                                        ;;**************************************************************
```

```
1226  005070  112737  000001  005334  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
1227  005076  112737  000001  005332  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
1228  005104  000403                  BR      $ATYC
1229  005106  112737  000001  005334  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
1230  005114                          $ATYC:
1231  005114  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
1232  005116  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
1233  005120  105737  005332                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
1234  005124  001450                          BEQ     5$              ;;IF NOT:  BR
1235  005126  122737  000001  001140          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
1236  005134  001031                          BNE     3$              ;;IF NOT:  BR
1237  005136  132737  000100  001141          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
1238  005144  001425                          BEQ     3$              ;;IF NOT:  BR
1239  005146  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
1240  005152  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
1241  005160  005737  001120          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
1242  005164  001375                          BNE     1$              ;;IF NOT:  WAIT
1243  005166  010037  001134                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
1244  005172  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
1245  005174  001376                          BNE     2$
1246  005176  163700  001134                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
1247  005202  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
1248  005204  010037  001136                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
1249  005210  012737  000004  001120          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
1250  005216  000413                          BR      5$
1251  005220  017637  000004  005244  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
1252  005226  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
1253  005234  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
1254  005240  004737  004606                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
1255  005244  000000                  4$:     .WORD   0
1256  005246                          5$:
1257  005246  105737  005334          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
1258  005252  001416                          BEQ     12$             ;;IF NOT:  BR
1259  005254  005737  001140                  TST     $ENV            ;;RUNNING UNDER APT?
1260  005260  001413                          BEQ     12$             ;;IF NOT:  BR
1261  005262  005737  001120          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
1262  005266  001375                          BNE     11$             ;;IF NOT:  WAIT
1263  005270  017637  000004  001122          MOV     @4(SP),$FATAL   ;;GET ERROR #
1264  005276  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
1265  005304  005237  001120                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
1266  005310  105037  005334          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
1267  005314  105037  005333                  CLRB    $LFLG           ;;CLEAR LOG FLAG
1268  005320  105037  005332                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
1269  005324  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
1270  005326  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
1271  005330  000207                          RTS     PC              ;;RETURN
1272  005332     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
1273  005333     000          $LFLG:  .BYTE   0               ;;LOG FLAG
1274  005334     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
1275        005336                          .EVEN
1276        000200          APTSIZE=200
1277        000001          APTENV=001
1278        000100          APTSPOOL=100
1279        000040          APTCSUP=040
1280
1281                                ;STRING INPUT ROUTINE
```

```
1282                                        ;------------------------
1283
1284  005336  010346                .INSTR: MOV     R3,-(SP)        ;SAVE R3 ON STACK
1285  005340  010446                        MOV     R4,-(SP)        ;SAVE R4 ON STACK
1286  005342  017637  000004  005360        MOV     @4(SP),.MSG     ;GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
1287  005350  062766  000002  000004        ADD     #2,4(SP)        ;POINT TO INSTRUCTION AFTER ADDRESS POINTER
1288  005356  104402                .INST1: TYPE                    ;PRINT THE MESSAGE
1289  005360  000000                .MSG:   0                       ;MESSAGE IS POINTED TO FROM HERE
1290  005362  012704  010270                MOV     #INBUF,R4       ;POINT R4 TO THE INPUT BUFFER
1291  005366  012703  000007                MOV     #7,R3           ;SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
1292  005372  105777  173712        1$:     TSTB    @$TKS           ;HAS A CHARACTER BEEN RECEIVED?
1293  005376  100375                        BPL     1$              ;IF NO, KEEP WAITING FOR IT
1294  005400  117714  173706                MOVB    @$TKB,(R4)      ;IF YES, SAVE IT IN THE INPUT BUFFER
1295  005404  142714  000200                BICB    #200,(R4)       ;KEEP ONLY THE 7-BIT ASCII INFORMATION
1296  005410  122427  000015                CMPB    (R4)+,#15       ;IS THIS CHARACTER A LINE FEED?
1297  005414  001417                        BEQ     INSTR2          ;IF SO, TERMINATE THE INPUT SEQUENCE
1298  005416  105777  173672        2$:     TSTB    @$TPS           ;IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
1299  005422  100375                        BPL     2$              ;IF WE CAN'T, WAIT UNTIL WE CAN
1300  005424  017777  173662  173664        MOV     @$TKB,@$TPB     ;ECHO THE CHARACTER BACK
1301  005432  005303                        DEC     R3              ;REDUCE THE NUMBER OF CHARACTERS RECEIVED
1302  005434  001356                        BNE     1$              ;IF WE DON'T HAVE 7, GO GET SOME MORE
1303  005436  012604                        MOV     (SP)+,R4        ;IF WE HAVE 7, RESTORE R4
1304  005440  012603                        MOV     (SP)+,R3        ;RESTORE R3
1305  005442  010346                .INSTE: MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
1306  005444  010446                        MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
1307  005446  104402  001356                TYPE    ,$QUES          ;PRINT A QUESTION MARK... WHAT'S GOING ON?
1308  005452  000741                        BR      .INST1          ;GO PRINT THE MESSAGE AGAIN
1309  005454  012604                INSTR2: MOV     (SP)+,R4        ;RESTORE R4
1310  005456  012603                        MOV     (SP)+,R3        ;RESTORE R3
1311  005460  000002                        RTI                     ;RETURN TO THE MAIN PROCEDURE
1312
1313                                        ;CONVERT ASCII STRING TO OCTAL
1314                                        ;----------------------------
1315
1316  005462  010546                .PARAM: MOV     R5,-(SP)        ;SAVE R5 ON THE STACK
1317  005464  010446                        MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
1318  005466  016605  000004                MOV     4(SP),R5        ;GET THE SETUP INFORMATION POINTER
1319  005472  012537  005652                MOV     (R5)+,LOLIM     ;SET THE LOW LIMIT FOR THE INPUT
1320  005476  012537  005654                MOV     (R5)+,HILIM     ;SET THE HIGH LIMIT FOR THE INPUT
1321  005502  012537  005656                MOV     (R5)+,DEVADR    ;SAVE THE ADDRESS WHERE THE RESULT WILL BE STORED
1322  005506  112537  005660                MOVB    (R5)+,LOBITS    ;GET THE MASK OF THE INCORRECT BITS
1323  005512  112537  005661                MOVB    (R5)+,ADRCNT    ;GET THE COUNT OF ITEMS TO BE STORED
1324  005516  010566  000004                MOV     R5,4(SP)        ;POINT TO WHERE MAIN LINE PROGRAM WILL RESUME
1325  005522  005005                PARAM1: CLR     R5              ;INITIALIZE THE ASCII TO OCTAL RESULT WORD
1326  005524  012704  010270                MOV     #INBUF,R4       ;POINT TO THE INPUT BUFFER
1327  005530  122714  000015                CMPB    #15,(R4)        ;IS THIS CHARACTER A CARRIAGE RETURN?
1328  005534  001420                        BEQ     PARERR          ;IF SO, PRINT THE MESSAGE AGAIN
1329  005536  121427  000060        1$:     CMPB    (R4),#60        ;IS THIS CHARACTER BELOW THE NUMERIC RANGE?
1330  005542  002415                        BLT     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
1331  005544  121427  000067                CMPB    (R4),#67        ;IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
1332  005550  003012                        BGT     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
1333  005552  142714  000060                BICB    #60,(R4)        ;ISOLATE THE NUMBER THE CHARACTER REPRESENTS
1334  005556  152405                        BISB    (R4)+,R5        ;CONCATENATE THESE BITS TO THE ALREADY EXISTING STRING
1335  005560  122714  000015                CMPB    #15,(R4)        ;IS THE NEXT CHARACTER A CARRIAGE RETURN?
1336  005564  001406                        BEQ     LIMITS          ;IF SO, GO SEE IF NUMBER IS WITHIN LIMITS
1337  005566  006305                        ASL     R5              ;CLEAR BIT POSITION 0, MOVE EXISTING STRING TO LEFT
```

```
1338  005570  006305                  ASL     R5              ;CLEAR POSITION 1, MOVE STRING TO LEFT AGAIN
1339  005572  006305                  ASL     R5              ;MOVE THE STRING ONE MORE TIME TO MAKE ROOM FOR
1340                                                          ;NEXT THREE BITS
1341  005574  000760                  BR      1$              ;GO GET THE NEXT CHARACTER
1342  005576  104404          PARERR: INSTER                  ;THERE WAS AN ERROR... GO PRINT MESSAGE AGAIN
1343  005600  000750                  BR      PARAM1          ;TRY GETTING THE PARAMETERS AGAIN
1344
1345                                  ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
1346                                  ;--------------------------------------
1347
1348  005602  020537  005654  LIMITS: CMP     R5,HILIM        ;DOES RESULT EXCEED ITS MAXIMUM CORRECT VALUE?
1349  005606  101373                  BHI     PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
1350  005610  020537  005652          CMP     R5,LOLIM        ;IS THE RESULT LOWER THAN ALLOWED?
1351  005614  103770                  BLO     PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
1352  005616  133705  005660          BITB    LOBITS,R5       ;ARE ANY INCORRECT BITS SET IN THE RESULT?
1353  005622  001365                  BNE     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
1354
1355                                  ;STORE NUMBER AT SPECIFIED ADDRESS
1356
1357  005624  013704  005656          MOV     DEVADR,R4       ;POINT TO THE LOCATION WHERE THE RESULT WILL BE STORED
1358  005630  010524          1$:     MOV     R5,(R4)+        ;STORE THE RESULT
1359  005632  062705  000002          ADD     #2,R5           ;CALCULATE THE NEXT DATUM
1360  005636  105337  005661          DECB    ADRCNT          ;REDUCE COUNT OF STORED RESULTS. IS IT EXCEEDED?
1361  005642  001372                  BNE     1$              ;IF NOT, GO STORE THE NEXT DATUM
1362  005644  012604                  MOV     (SP)+,R4        ;RESTORE R4
1363  005646  012605                  MOV     (SP)+,R5        ;RESTORE R5
1364  005650  000002                  RTI                     ;RETURN TO THE MAIN PROGRAM
1365
1366  005652  000000          LOLIM:  0                       ;LOWEST ACCEPTABLE VALUE
1367  005654  000000          HILIM:  0                       ;HIGHEST ACCEPTABLE
1368  005656  000000          DEVADR: 0                       ;LOCATION WHERE RESULT WILL BE STORED
1369  005660     000          LOBITS: .BYTE   0               ;INCORRECT BITS MASK
1370  005661     000          ADRCNT: .BYTE   0               ;COUNT OF ITEMS TO BE STORED
1371
1372                                  ;SAVE PC OF TEST THAT FAILED AND R0-R5
1373                                  ;-------------------------------------
1374
1375  005662  016637  000004  001404 .SAV05: MOV     4(SP),SAVPC     ;SAVE R7 (PC)
1376
1377                                  ;SAVE R0-R5
1378
1379  005670  010537  001340          SV05:   MOV     R5,$REG5        ;SAVE R5
1380  005674  010437  001336                  MOV     R4,$REG4        ;SAVE R4
1381  005700  010337  001334                  MOV     R3,$REG3        ;SAVE R3
1382  005704  010237  001332                  MOV     R2,$REG2        ;SAVE R2
1383  005710  010137  001330                  MOV     R1,$REG1        ;SAVE R1
1384  005714  010037  001326                  MOV     R0,$REG0        ;SAVE R0
1385  005720  000002                  RTI                     ;LEAVE.
1386
1387                                  ;RESTORE R0-R5
1388
1389  005722  013700  001326          .RES05: MOV     $REG0,R0        ;RESTORE R0
1390  005726  013701  001330                  MOV     $REG1,R1        ;RESTORE R1
1391  005732  013702  001332                  MOV     $REG2,R2        ;RESTORE R2
1392  005736  013703  001334                  MOV     $REG3,R3        ;RESTORE R3
1393  005742  013704  001336                  MOV     $REG4,R4        ;RESTORE R4
```

```
1394  005746  013705  001340                  MOV     $REGS,R5        ;RESTORE R5
1395  005752  000002                          RTI                     ;LEAVE
1396
1397                                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
1398                                          ;-----------------------------------------------------
1399
1400  005754  104402  001357          .CONVR: TYPE    ,$CRLF          ;PRINT A CARRIAGE RETURN
1401  005760  010046                  .CNVRT: MOV     R0,-(SP)        ;SAVE R0
1402  005762  010146                          MOV     R1,-(SP)        ;SAVE R1
1403  005764  010346                          MOV     R3,-(SP)        ;SAVE R3
1404  005766  010446                          MOV     R4,-(SP)        ;SAVE R4
1405  005770  010546                          MOV     R5,-(SP)        ;SAVE R5
1406  005772  017601  000012                  MOV     @12(SP),R1      ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
1407  005776  062766  000002  000012          ADD     #2,12(SP)       ;POINT TO WHERE MAIN PROGRAM WILL RESUME
1408  006004  012137  006130                  MOV     (R1)+,WRDCNT    ;GET NUMBER OF WORDS TO BE PRINTED
1409  006010  112105                  1$:     MOVB    (R1)+,R5        ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
1410  006012  112100                          MOVB    (R1)+,R0        ;GET THE NUMBER OF SPACES TO PRINT
1411  006014  013104                          MOV     @(R1)+,R4       ;COPY THE WORD TO BE CONVERTED
1412  006016  110537  006132                  MOVB    R5,CHRCNT       ;COPY THE CHARACTER COUNT
1413  006022  010403                  3$:     MOV     R4,R3           ;COPY THE ARGUMENT WORD AGAIN
1414  006024  042703  177770                  BIC     #^C<7>,R3       ;ISOLATE THREE BITS TO BE TREATED AS A CHARACTER
1415  006030  062703  000060                  ADD     #060,R3         ;MAKE AN ASCII CHARACTER OUT OF THEM
1416  006034  110346                          MOVB    R3,-(SP)        ;SAVE THAT CHARACTER
1417  006036  006004                          ROR     R4              ;MOVE THE NEXT THREE BITS INTO PLACE
1418  006040  006204                          ASR     R4              ;MOVE THEM AGAIN
1419  006042  006204                          ASR     R4              ;AND FINALLY A THIRD TIME
1420  006044  005305                          DEC     R5              ;REDUCE CHARACTER COUNT.ARE ALL CHARACTERS
1421                                                                  ;BUILT?
1422  006046  001365                          BNE     3$              ;IF NO, GO BUILD THE NEXT ONE.
1423  006050  012703  010374                  MOV     #MDATA,R3       ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
1424  006054  112623                  4$:     MOVB    (SP)+,(R3)+     ;STORE THE CHARACTER, STARTING WITH THE MOST
1425  006056  105337  006132                  DECB    CHRCNT          ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
1426  006062  001374                          BNE     4$              ;IF NO, GO TRANSFER ANOTHER
1427  006064  105700                          TSTB    R0              ;ARE ANY SPACES TO BE PRINTED?
1428  006066  001404                          BEQ     6$              ;IF NO, DON'T SET UP ANY
1429  006070  112723  000040          5$:     MOVB    #040,(R3)+      ;ADD A SPACE TO THE OUTPUT BUFFER
1430  006074  105300                          DECB    R0              ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
1431  006076  001374                          BNE     5$              ;IF YES, GO ADD ANOTHER SPACE
1432  006100  105013                  6$:     CLRB    (R3)            ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
1433  006102  104402  010374                  TYPE    ,MDATA          ;PRINT THE STRING WE JUST BUILT
1434  006106  005337  006130                  DEC     WRDCNT          ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
1435  006112  001336                          BNE     1$              ;IF YES, GO CONVERT THEM
1436  006114  012605                          MOV     (SP)+,R5        ;RESTORE R5
1437  006116  012604                          MOV     (SP)+,R4        ;RESTORE R4
1438  006120  012603                          MOV     (SP)+,R3        ;RESTORE R3
1439  006122  012601                          MOV     (SP)+,R1        ;RESTORE R1
1440  006124  012600                          MOV     (SP)+,R0        ;RESTORE R0
1441  006126  000002                          RTI                     ;RETURN TO THE MAIN PROGRAM
1442  006130  000000          WRDCNT: 0
1443  006132     000          CHRCNT: .BYTE                           ;NUMBER OF CHARACTERS TO PRINT
1444  006133     000          SPACNT: .BYTE   0                       ;NUMBER OF SPACES TO PRINT
1445
1446  006134  000000          BINWRD: 0
1447
1448
1449                                          ;TRAP DISPATCH SERVICE
```

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

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

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

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

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

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

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

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

```
                                        ;LINE PARAMETER REGISTER SETUP ROUTINE

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

                                        ;ROUTINE TO ZERO DATA BUFFER

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

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

006344  004737  006772        $ERROR: JSR     PC,SERV.G       ;FIND OUT IF <^G> WAS HIT
006350  032777  010000  172726        BIT     #SW12,@SWR      ;BELL ON ERROR?
006356  001406                        BEQ     XBX             ;BR IF NO BELL
006360  105777  172730                TSTB    @$TPS           ;TTY READY.
006364  100003                        BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
006366  112777  000207  172722        MOVB    #207,@$TPB      ;PUSH A BELL AT THE TTY.
006374  032777  020000  172702 XBX:   BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
006402  001113                        BNE     HALT$           ;BR IF NO PRINT OUT WANTED.
006404  021637  001262                CMP     (SP),$ERRPC     ;WAS THIS ERROR FOUND LAST TIME?
006410  001404                        BEQ     1$              ;BR IF YES
006412  011637  001262                MOV     (SP),$ERRPC     ;RECORD BEING HERE
006416  105037  001247                CLRB    $ERFLG          ;PREPARE HEADER
006422  104407                1$:     SAV05                   ;SAVE ALL PROC REGISTERS
006424  011605                        MOV     (SP),R5         ;GET THE PC OF ERROR
006426  162705  000002                SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
006432  011504                        MOV     (R5),R4         ;GET ERROR INSTRUCTION
006434  110437  001260                MOVB    R4,$ITEMB       ;COPY TEST NUMBER FOR APT HANDLING
006440  006304                        ASL     R4              ;MULT BY TWO
006442  061504                        ADD     (R5),R4         ;DOUBLE IT
006444  006304                        ASL     R4              ;MULT AGAIN
006446  042704  177001                BIC     #177001,R4      ;CLEAR JUNK
006452  062704  015316                ADD     #.ERRTAB,R4     ;GET POINTER
006456  012437  006602                MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
006462  012437  006614                MOV     (R4)+,DATAHD    ;GET DATA HEADRER
006466  011437  006626                MOV     (R4),DATABP     ;GET DATA TABLE
006472  105737  001247                TSTB    $ERFLG          ;TYPE HEADER
006476  001403                        BEQ     TYPMSG          ;BR IF YES
006500  005737  006626                TST     DATABP          ;DOES DATA TABLE EXIST?
```

```
1559  006504  001044                    BNE     TYPDAT          ;BR IF YES.
1560  006506  104402  001357    TYPMSG: TYPE    ,$CRLF          ;TYPE A CARRIAGE RETURN
1561  006512  104402  001357            TYPE    ,$CRLF          ;AND TYPE ANOTHER
1562  006516  005737  001364            TST     LOCK
1563  006522  001402                    BEQ     1$
1564  006524  104402  007705            TYPE    ,MASTEK
1565  006530  104402  007673    1$:     TYPE    ,MTSTN
1566  006534  104412  006764            CNVRT   ,XTSTN          ;SHOW IT
1567  006540  104402  007765            TYPE    ,MERRPC         ;TYPE PC.
1568  006544  104412  006756            CNVRT   ,ERTABO         ;SHOW IT
1569  006550  104402  007635            TYPE    ,MCSRX
1570  006554  104412  004250            CNVRT   ,XCSR
1571  006560  104402  001357            TYPE    ,$CRLF          ;GIVE A CR/LF
1572  006564  112737  177777  001247    MOVB    #-1,$ERFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
1573  006572  005737  006602            TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
1574  006576  001402                    BEQ     WTBS.FM         ;BR IF NO.
1575  006600  104402                    TYPE                    ;TYPE
1576  006602  000000            ERRMSG: 0                       ;    ERROR MESSAGE
1577  006604                    WTBS.FM:
1578  006604  005737  006614            TST     DATAHD          ;DATA HEADER?
1579  006610  001402                    BEQ     TYPDAT          ;BR IF NO
1580  006612  104402                    TYPE                    ;TYPE
1581  006614  000000            DATAHD: 0                       ;    DATA HEADER
1582  006616  005737  006626    TYPDAT: TST     DATABP          ;DATA TABLE?
1583  006622  001402                    BEQ     RESREG          ;BR IF NO.
1584  006624  104411                    CONVRT                  ;SHOW
1585  006626  000000            DATABP: 0                       ;    DATA TABLE
1586  006630  104410            RESREG: RESO5                   ;RESTORE PROC REGISTERS
1587  006632  122737  000001  001140  HALTS:  CMPB    #APTENV,$ENV    ;IS APT RUNNING?
1588  006640  001007                    BNE     15$             ;SKIP APT CALL IF NOT
1589  006642  113737  001260  006654    MOVB    $ITEMB,5$       ;COPY ERROR NUMBER
1590  006650  004737  005106            JSR     PC,$ATY4        ;CALL APT SERVICE
1591  006654  000000            5$:     .WORD   0               ;ERROR NUMBER STUCK HERE
1592  006656  000777            10$:    BR      10$             ;LOCK UP HERE
1593  006660  022737  004234  000042  15$:    CMP     #$ENDAD,@#42    ;CHECK TO SEE IF IN ACT-11 MODE
1594  006666  001403                    BEQ     20$             ;IF SO, HANDLE ACCORDINGLY
1595  006670  005777  172410            TST     @SWR            ;HALT ON ERROR?
1596  006674  100004                    BPL     EXITER          ;BR IF NO HALT ON ERROR
1597  006676  016677  000002  172402  20$:    MOV     2(SP),@DISPLAY  ;SHOW ERROR PC IN DATA DISPLAY
1598  006704  000000                    HALT                    ;HALT
1599  006706  005237  001256    EXITER: INC     $ERTTL          ;UPDATE ERROR COUNT
1600  006712  004737  006772            JSR     PC,SERV.G       ;FIND OUT IF ↑G WAS TYPED
1601  006716  032777  000400  172360    BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
1602  006724  001007                    BNE     1$              ;BR IF YES
1603  006726  032777  002000  172350    BIT     #SW10,@SWR      ;GOTO NEXT TEST?
1604  006734  001407                    BEQ     2$              ;BR IF NO
1605  006736  013737  001362  001252    MOV     NEXT,$LPADR     ;SET FOR NEXT TEST
1606  006744  012706  001120    1$:     MOV     #STACK,SP       ;RESET SP
1607  006750  000177  172276            JMP     @$LPADR         ;GOTO SPECIFIED TEST
1608  006754  000002            2$:     RTI                     ;RETURN
1609  006756  000001            ERTABO: 1
1610  006760     006     002            .BYTE   6,2
1611  006762  001404                    $AVPC
1612  006764  000001            XTSTN:  1
1613  006766     002     002            .BYTE   2,2
1614  006770  001246                    $TSTNM
```

```
1615  006772  017746  172314          SERV.G: MOV     @STKB,-(SP)     ;OTHERWISE, GET THE LAST CHARACTER TYPED
1616  006776  042716  000200                  BIC     #BIT7,(SP)      ;STRIP PARITY(EIGHTH) BIT
1617  007002  122726  000007                  CMPB    #7,(SP)+        ;IS IT ↑G?
1618  007006  001076                          BNE     6$              ;IF NOT, IGNORE INPUT
1619  007010  032777  004000  172272          BIT     #4000,@STKS     ;RX BUSY?
1620  007016  001365                          BNE     SERV.G          ;BR IF YES
1621  007020  017737  172260  007226          MOV     @SWR,90$        ;SAVE (SWR).
1622  007026  104402  007206          1$:     TYPE    ,89$            ;TYPE HEADER FOR OLD SWITCH REGISTER
1623  007032  104412  007220                  CNVRT   ,88$            ;TYPE THE NUMBER ITSELF
1624  007036  104402  007230                  TYPE    ,91$            ;AFTER HAVING CONVERTED IT TO ASCII
1625  007042  105037  007234                  CLRB    92$             ;CLEAR SWR CHANGE FLAG
1626  007046  005077  172232                  CLR     @SWR            ;CLEAR THE SOFTWARE SWITCH REGISTER
1627  007052  105777  172232          3$:     TSTB    @STKS           ;WAIT FOR DONE
1628  007056  100375                          BPL     3$              ;CONTINUE WAITING FOR IT
1629  007060  017746  172226                  MOV     @STKB,-(SP)     ;PUT THE CHARACTER ON THE STACK
1630  007064  042716  000200                  BIC     #BIT7,(SP)      ;STRIP PARITY BIT
1631  007070  122726  000015                  CMPB    #15,(SP)+       ;IS IT THE CARRIAGE RETURN CHAR?
1632  007074  001433                          BEQ     4$              ;IF SO, GO PRINT CRLF
1633  007076  105777  172212          2$:     TSTB    @STPS           ;IS THE OUTPUT BUFFER AVAILABLE
1634  007102  100375                          BPL     2$              ;IF NOT, WAIT FOR IT TO BE READY
1635  007104  105237  007234                  INCB    92$             ;INDICATE THAT THE SWR WAS CHANGED
1636  007110  014677  172202                  MOV     -(SP),@STPB     ;PLACE THE CHARACTER THERE(ECHO BACK)
1637  007114  000241                          CLC                     ;GET READY TO ROTATE
1638  007116  006177  172162                  ROL     @SWR            ;MOVE THE EXISTING BITS OVER
1639  007122  006177  172156                  ROL     @SWR            ;TO MAKE ROOM FOR THE INCOMING
1640  007126  006177  172152                  ROL     @SWR            ;THREE BITS FROM THIS CHARACTER
1641  007132  103735                          BCS     1$              ;ERROR
1642  007134  022627  000060                  CMP     (SP)+,#60       ;IS IT LOWER THAN 0?
1643  007140  002732                          BLT     1$              ;IF SO, GO ASK AGAIN
1644  007142  026627  177776  000067          CMP     -2(SP),#67      ;IS IT HIGHER THAN 7?
1645  007150  003326                          BGT     1$              ;IF SO, GO ASK AGAIN
1646  007152  042746  177770                  BIC     #↑C<7>,-(SP)    ;ISOLATE INFORMATION BITS
1647  007156  052677  172122                  BIS     (SP)+,@SWR      ;ADD THEM TO THE SWITCH REGISTER
1648  007162  000733                          BR      3$              ;GO CHECK FOR THE NEXT CHARACTER
1649  007164  105737  007234          4$:     TSTB    92$             ;HAS THE SWR BEEN CHANGED?
1650  007170  001003                          BNE     5$              ;IF YES GO TYPE CRLF
1651  007172  013777  007226  172104          MOV     90$,@SWR        ;IF NOT RESTORE SWR
1652  007200  104402  001357          5$:     TYPE    ,$CRLF          ;TYPE A CARRIAGE RETURN AND LINE FEED
1653  007204  000207                  6$:     RTS     PC              ;RETURN TO CALLING PROCEDURE
1654
1655  007206  020200  051450  051127  89$:     .ASCIZ  <200>? (SWR)=/?
1656  007214  036451  000057
1657                                  .EVEN
1658  007220  000001                  88$:    1
1659  007222     006     000                  .BYTE   6,0
1660  007224  007226                          90$
1661  007226  000000                  90$:    .WORD   0
1662  007230  036457  000057          91$:    .ASCIZ  ?/=/?
1663  007234     000                  92$:    .BYTE   0
1664          007236                  .EVEN
1665                                  .SBTTL  POWER DOWN AND UP ROUTINES
1666
1667                                  ;;*****************************************************************
1668                                  ;POWER DOWN ROUTINE
1669  007236  012737  007402  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
1670  007244  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
```

```
 1671  007252  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 1672  007254  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 1673  007256  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 1674  007260  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 1675  007262  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 1676  007264  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 1677  007266  017746  172012                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
 1678  007272  010637  007406                MOV     SP,$SAVR6       ;;SAVE SP
 1679  007276  012737  007310  000024        MOV     #$PWRUP,@#PWRVEC;;SET UP VECTOR
 1680  007304  000000                        HALT
 1681  007306  000776                        BR      .-2             ;;HANG UP
 1682
 1683                                  ;;*****************************************************************
 1684                                  ;POWER UP ROUTINE
 1685  007310  012737  007402  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
 1686  007316  013706  007406                MOV     $SAVR6,SP       ;;GET SP
 1687  007322  005037  007406                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
 1688  007326  005237  007406          1$:   INC     $SAVR6          ;;WAIT FOR THE INC
 1689  007332  001375                        BNE     1$              ;;OF  WORD
 1690  007334  012677  171744                MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
 1691  007340  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
 1692  007342  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
 1693  007344  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 1694  007346  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 1695  007350  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 1696  007352  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 1697  007354  012737  007236  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
 1698  007362  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
 1699  007370  104402                        TYPE                    ;;REPORT THE POWER FAILURE
 1700  007372  007410                  $PWRMG: .WORD   MPFAIL          ;;POWER FAIL MESSAGE POINTER
 1701  007374  012716                        MOV     (PC)+,(SP)      ;;RESTART AT RESTART
 1702  007376  010776                  $PWRAD: .WORD   RESTART         ;;RESTART ADDRESS
 1703  007400  000002                        RTI
 1704  007402  000000                  $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
 1705  007404  000776                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
 1706  007406  000000                  $SAVR6: 0                       ;;PUT THE SP HERE
 1707  007410  050200  051127  043040  MPFAIL: .ASCIZ  <200>/PWR FAILED. RESTART AT LAST TEST /
   (2)  007453     200  047105  020104  MEPASS: .ASCIZ  <200>/END PASS DVDZB-A  /
   (2)  007477     200  052522  047116  MR:     .ASCIZ  <200>/RUNNING    /
   (2)  007513     200  051120  043517  MERR2:  .ASCIZ  <200>/PROGRAM INDICATES NO DEVICES PRESENT./
   (2)  007562  044600  051516  043125  MERR3:  .ASCIZ  <200>/INSUFFICIENT DATA!/
   (2)  007606  046200  041517  020113  MLOCK:  .ASCIZ  <200>/LOCK ON SELECTED TEST/
   (2)  007635     103  051123  020072  MCSRX:  .ASCIZ   /CSR: /
   (2)  007643     126  041505  020072  MVECX:  .ASCIZ  /VEC: /
   (2)  007651     120  051501  042523  MPASSX: .ASCIZ  /PASSES: /
   (2)  007662  051105  047522  051522  MERRX:  .ASCIZ  /ERRORS: /
   (2)  007673     124  051505  020124  MTSTN:  .ASCIZ  /TEST NO: /
   (2)  007705      52  000040          MASTEK: .ASCIZ  /* /
   (2)  007710  052200  050131  020105  MNEW:   .ASCIZ  <200>/TYPE A BIT MAP OF DZV11'S DESIRED ACTIVE:  /
   (2)  007765     120  035103  000040  MERRPC: .ASCIZ  /PC: /
   (2)  007772  046600  050101  047440  XHEAD:  .ASCIZ  <200>/MAP OF DZV11 STATUS/<200>
   (2)  010020  044600  046114  043505  MBADLN: .ASCIZ  <200>/ILLEGAL ENTRY IN STAGGERED MODE/<200>
   (2)                                  .EVEN
   (2)  010062  000002                  XSTATQ: 2
 1708  010064     006       003                .BYTE   6,3
 1709  010066  001344                        $TMP1
```

```
1710  010070  006     002                     .BYTE   6,2
1711  010072  001346                         $TMP2
1712                                  .EVEN
1713                                  ;THIS ROUTINE ESTABLISHES WHICH MAINTENANCE MODE THE DEVICE IS IN
1714                                  ;-------------------------------------------------------------
1715                                  ;E=EXTERNAL LOOP BACK
1716                                  ;I=INTERNAL LOOP BACK
1717                                  ;S=STAGGERED LOOP BACK
1718  010074  017605  000000          .SETFLG:MOV     @(SP),R5        ;PICK UP ADDRESS OF TAG
1719  010100  042737  000040  010270          BIC     #40,INBUF       ;STRIP LOWER CASE
1720  010106  122737  000105  010270          CMPB    #'E,INBUF       ;IS IT EXTERNAL LOOP BACK ?
1721  010114  001005                         BNE     4$              ;NO
1722  010116  013715  010206                 MOV     1$,(R5)         ;YES STORE INFO
1723  010122  105037  001424                 CLRB    MNTFLG          ;SET MAINT BIT =0
1724  010126  000422                         BR      7$              ;GET OUT
1725  010130  122737  000111  010270  4$:     CMPB    #'I,INBUF       ;IS IT INTERNAL LOOP BACK ?
1726  010136  001006                         BNE     5$              ;NO
1727  010140  013715  010210                 MOV     2$,(R5)         ;YES STORE INFO
1728  010144  112737  000010  001424         MOVB    #MAINT,MNTFLG   ;SET UP THE MAINTENANCE FLAG LOADER
1729  010152  000410                         BR      7$              ;GET OUT
1730  010154  122737  000123  010270  5$:     CMPB    #'S,INBUF       ;IS IT STAGGERED LOOP BACK ?
1731  010162  001007                         BNE     6$              ;WHAT ?
1732  010164  013715  010212                 MOV     3$,(R5)         ;YES STORE INFO
1733  010170  105037  001424                 CLRB    MNTFLG          ;ZERO BITS
1734  010174  062716  000002          7$:     ADD     #2,(SP)         ;POP AROUND
1735  010200  000002                         RTI
1736  010202  104404                  6$:     INSTER                  ;RETRY
1737  010204  000733                         BR      .SETFLG         ;DITTO
1738  010206  000200                  1$:     .WORD   200             ;EXTERNAL =  E
1739  010210  000000                  2$:     .WORD   0               ;INTERNAL = I
1740  010212  100000                  3$:     .WORD   100000          ;STAGGERED = S
1741
```

```
1742                                            ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1743                                            ;BUFFER TO THE CHARACTERS "E" AND "C".
1744                                            ;IF THE CHARACTER IS "E" CLEAR THE FLAG
1745                                            ;IF THE CHARACTER IS "C" SET THE FLAG
1746
1747  010214  017605  000000          .PAWCH: MOV     @(SP),R5
1748  010220  142737  000040  010270          BICB    #40,INBUF       ;SET FOR LOWER CASE INPUT
1749  010226  122737  000105  010270          CMPB    #'E,INBUF       ;IS IT "E" ?
1750  010234  001002                          BNE     1$
1751  010236  105015                          CLRB    (R5)            ;000
1752  010240  000406                          BR      2$
1753  010242  122737  000103  010270  1$:     CMPB    #'C,INBUF       ;IS IT "C" ?
1754  010250  001005                          BNE     3$
1755  010252  112715  177777                  MOVB    #-1,(R5)        ;3177
1756  010256  062716  000002          2$:     ADD     #2,(SP)
1757  010262  000002                          RTI
1758  010264  104404                  3$:     INSTER                  ;RETRY
1759  010266  000752                          BR      .PAWCH
1760
1761                                            ;BUFFERS FOR INPUT-OUTPUT
1762
1763  010270  000000                  INBUF:  0
1764          010332                  .=.+40
1765  010332  000000                  TEMP:   0
1766          010374                  .=.+40
1767  010374  000000                  MDATA:  0
1768          010436                  .=.+40
1769
```

```
1770
1771                                    ;
1772                                    ;ROUTINE USED TO "CYCLE" THROUGH UP TO SIXTEEN DZV11'S
1773                                    ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
1774                                    ;AND RUNS THE SPECIFIED DZV11'S.  THIS ROUTINE *MUST*
1775                                    ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
1776                                    ;SETUP NECESSARY.
1777                                    ;
1778
1779  010436  005737  001406    CYCLE:  TST     DZVACTV         ;ARE ANY DZV11'S TO BE TESTED?
1780  010442  001004                    BNE     1$              ;BR IF OK.
1781  010444  104402  007513            TYPE    ,MERR2          ;NO DZV11'S SELECTED!!
1782  010450  000000                    HALT                    ;STOP THE SHOW.
1783  010452  000776                    BR      .-2             ;DISQUALIFY CONT. SW.
1784  010454  013737  004542  001354 1$: MOV     $MXCNT,$TIMES   ;RESTORE THE NUMBER OF ITERATIONS TO MAKE
1785  010462  033737  001412  001406    BIT     RUN,DZVACTV     ;IS THIS ONE "ACTIVE"
1786  010470  001017                    BNE     2$              ;BR IF GOOD ONE FOUND.
1787  010472  006137  001412            ROL     RUN             ;UPDATE POINTER
1788  010476  005537  001412            ADC     RUN             ;CATCH CARRY FROM RUN
1789  010502  062737  000012  001420    ADD     #12,ACTIVE      ;UPDATE ADDRESS POINTER.
1790  010510  022737  001740  001420    CMP     #DZV.END,ACTIVE ;HAVE WE PASSED THE END OF THE MAP?
1791  010516  001356                    BNE     1$              ;IF NO, KEEP GOING; NOT ALL TESTED FOR.
1792  010520  012737  001500  001420    MOV     #DZV.MAP,ACTIVE ;RESET ADDRESS POINTER.
1793  010526  000752                    BR      1$              ;KEEP LOOKING FOR ACTIVE DZV11
1794  010530  006137  001412        2$: ROL     RUN             ;UPDATE POINTER.
1795  010534  005537  001412            ADC     RUN             ;CATCH CARRY.
1796  010540  013700  001420            MOV     ACTIVE,R0       ;GET ADDRESS POINTER.
1797  010544  062737  000012  001420    ADD     #12,ACTIVE      ;UPDATE.
1798  010552  022737  001740  001420    CMP     #DZV.END,ACTIVE
1799                                                            ;ALL DONE?
1800  010560  001003                    BNE     3$              ;BR IF NO.
1801  010562  012737  001500  001420    MOV     #DZV.MAP,ACTIVE ;RESTORE POINTER.
1802  010570  012037  001174        3$: MOV     (R0)+,$BASE     ;LOAD SYSTEM CTRL. REG
1803  010574  012037  002040            MOV     (R0)+,DZVRIV    ;LOAD VECTOR
1804  010600  012037  001366            MOV     (R0)+,LINE      ;SET UP DZV LINES ACTIVE
1805  010604  012037  001370            MOV     (R0)+,PAR       ;SET UP PARAMETERIZATION
1806  010610  012037  001372            MOV     (R0)+,MODE      ;SET UP MAINTENANCE MODE
1807  010614  105037  001424            CLRB    MNTFLG  ;RESET MAINT. FLAG IF
1808  010620  005737  001372            TST     MODE            ;RUNNING TESTS
1809  010624  001003                    BNE     9$              ;IN
1810  010626  112737  000010  001424    MOVB    #MAINT,MNTFLG   ;INTERNAL MAINT. MODE
1811  010634  004737  011002        9$: JSR     PC,DZVLEV       ;SET UP
1812  010640  005737  000042            TST     @#42            ;ARE WE UNDER MONITOR CONTROL?
1813  010644  001051                    BNE     7$              ;IF YES, SKIP THIS SETUP
1814  010646  032777  000002  170430    BIT     #SW01,@SWR      ;IF SW01=1, GET STARTING TEST #
1815  010654  001445                    BEQ     7$              ;BR IF NO TEST IS TO BE INPUTTED
1816  010656  104402  001357        4$: TYPE    ,$CRLF
1817  010662  104403                    INSTR                   ;CALL THE STRING INPUT ROUTINE
1818  010664  007673                    MTSTN                   ;POINTER TO MESSAGE TO BE PRINTED
1819  010666  104405                    PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1820  010670  000001                    1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1821  010672  001000                    1000                    ;HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1822  010674  001246                    $TSTNM                  ;POINTER TO MAP LOCATION TO BE FILLED
1823  010676     000                    .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
1824  010677     001                    .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
1825  010700  012700  012170            MOV     #TST1,R0
```

```
1826  010704  022710  000004          5$:     CMP     #4,(R0)
1827  010710  001020                          BNE     6$
1828  010712  022760  012737  000002          CMP     #12737,2(R0)
1829  010720  001014                          BNE     6$
1830  010722  023760  001246  000004          CMP     $TSTNM,4(R0)    ;IS THIS THE TEST ?
1831  010730  001010                          BNE     6$              ;IF NOT, DON'T PROCESS NUMBER
1832  010732  010037  001252                  MOV     R0,$LPADR       ;SAVE PC
1833  010736  062737  000002  001252          ADD     #2,$LPADR       ;POP OVER PREVIOUS SCOPE
1834  010744  104402  001357                  TYPE    ,$CRLF
1835  010750  000412                          BR      8$
1836  010752  005720                  6$:     TST     (R0)+
1837  010754  020027  014322                  CMP     R0,#TLAST+10
1838  010760  001351                          BNE     5$
1839  010762  104402  001356                  TYPE    ,$QUES
1840  010766  000733                          BR      4$
1841  010770  012737  012170  001252  7$:     MOV     #TST1,$LPADR    ;PREPARE TEST ADDRESS
1842  010776                          8$:
1843  010776  000177  170250          RESTART:JMP     @$LPADR         ;GO START TESTING.***WARNING!****
1844                                                                  ;THIS JUMP IS USED BY POWER UP ROUTINE!!!!
1845
1846                                  ;THIS UTILITY SETS UP CSR'S,SETS UP VECTORS.
1847  011002  013700  002040          DZVLEV: MOV     DZVRIV,R0       ;PLACE THE BASE VECTOR ADDRESS IN R0
1848  011006  062700  000002                  ADD     #2,R0           ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
1849  011012  010037  002042                  MOV     R0,DZVRIS       ;STORE IT HERE
1850  011016  062700  000002                  ADD     #2,R0           ;CALCULATE THE TRANSMITTER INTERRUPT VECTOR
1851  011022  010037  002044                  MOV     R0,DZVTIV       ;STORE IT HERE
1852  011026  062700  000002                  ADD     #2,R0           ;CALCULATE THE TRANSMITTER VECTOR STATUS ADDRESS
1853  011032  010037  002046                  MOV     R0,DZVTIS       ;STORE IT HERE
1854
1855                                  ;THIS SEGMENT SETS UP POINTERS FOR THE GIVEN DZV11. $BASE IS THE BASE ADDRESS
1856                                  ;OF THE DEVICE
1857  011036  013700  001174                  MOV     $BASE,R0        ;COPY THE ADDRESS BEING LOADED
1858  011042  010037  002010                  MOV     R0,DZVCSR       ;XXX0
1859  011046  005200                          INC     R0
1860  011050  010037  002012                  MOV     R0,HDZVCSR      ;XXX1
1861  011054  005200                          INC     R0
1862  011056  010037  002014                  MOV     R0,DZVRBUF      ;XXX2
1863  011062  010037  002020                  MOV     R0,DZVLPR       ;XXX2
1864  011066  005200                          INC     R0
1865  011070  010037  002016                  MOV     R0,HDZVRBUF     ;XXX3
1866  011074  010037  002022                  MOV     R0,HDZVLPR      ;XXX3
1867  011100  005200                          INC     R0
1868  011102  010037  002024                  MOV     R0,DZVTCR       ;XXX4
1869  011106  005200                          INC     R0
1870  011110  010037  002026                  MOV     R0,HDZVTCR      ;XXX5
1871  011114  005200                          INC     R0
1872  011116  010037  002030                  MOV     R0,DZVMSR       ;XXX6
1873  011122  010037  002034                  MOV     R0,DZVTDR       ;XXX6
1874  011126  005200                          INC     R0
1875  011130  010037  002032                  MOV     R0,HDZVMSR      ;XXX7
1876  011134  010037  002036                  MOV     R0,HDZVTDR      ;XXX7
1877  011140  000207                          RTS     PC
```

```
1878                                            ;CONVERT DECIMAL ASCII STRING TO OCTAL
1879  011142  011605                  .PARMD: MOV     (SP),R5
1880  011144  012537  011326                  MOV     (R5)+,6$
1881  011150  012537  011330                  MOV     (R5)+,7$
1882  011154  012537  011332                  MOV     (R5)+,8$
1883  011160  112537  011334                  MOVB    (R5)+,9$
1884  011164  112537  011335                  MOVB    (R5)+,10$
1885  011170  010516                          MOV     R5,(SP)
1886  011172  005005                  2$:     CLR     R5
1887  011174  012704  010270                  MOV     #INBUF,R4
1888  011200  122714  000015                  CMPB    #15,(R4)
1889  011204  001424                          BEQ     3$
1890  011206  121427  000060          1$:     CMPB    (R4),#'0
1891  011212  002421                          BLT     3$
1892  011214  121427  000071                  CMPB    (R4),#'9
1893  011220  003016                          BGT     3$
1894  011222  142714  000060                  BICB    #'0,(R4)
1895  011226  005002                          CLR     R2
1896  011230  152402                          BISB    (R4)+,R2
1897  011232  060205                          ADD     R2,R5
1898  011234  122714  000015                  CMPB    #15,(R4)
1899  011240  001410                          BEQ     4$
1900  011242  006305                          ASL     R5              ;X2
1901  011244  010502                          MOV     R5,R2           ;SAVE X2
1902  011246  006305                          ASL     R5              ;X4
1903  011250  006305                          ASL     R5              ;X8
1904  011252  060205                          ADD     R2,R5           ;TIMES 10
1905  011254  000754                          BR      1$
1906  011256  104404                  3$:     INSTER
1907  011260  000744                          BR      2$
1908
1909                                            ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
1910
1911  011262  020537  011330          4$:     CMP     R5,7$
1912  011266  101373                          BHI     3$
1913  011270  020537  011326                  CMP     R5,6$
1914  011274  103770                          BLO     3$
1915  011276  133705  011334                  BITB    9$,R5
1916  011302  001365                          BNE     3$
1917
1918                                            ;STORE NUMBER AT SPECIFIED ADDRESS
1919
1920  011304  013704  011332                  MOV     8$,R4
1921  011310  010524                  5$:     MOV     R5,(R4)+
1922  011312  062705  000002                  ADD     #2,R5
1923  011316  105337  011335                  DECB    10$
1924  011322  001372                          BNE     5$
1925  011324  000002                          RTI
1926  011326  000000                  6$:     0
1927  011330  000000                  7$:     0
1928  011332  000000                  8$:     0
1929  011334     000                  9$:     .BYTE 0
1930  011335     000                  10$:    .BYTE 0
```

```
1931                                        ;*ROUTINE USED TO SET UP THE DIAGNOSTIC VIA APT.
1932                                        ;*IF BIT7 IN THE ENVIRONMENT MODE ($ENVM) BYTE IS SET,
1933                                        ;*THE PROGRAM WILL LOAD ITS PARAMETERS FROM THE ETABLE.
1934
1935  011336  012700  001500        SETAPT: MOV     #DZV.MAP,R0     ;POINT TO THE DEVICE MAP TABLE
1936  011342  013701  001174                MOV     $BASE,R1        ;BUILD DEVICE ADDRESSES IN R1
1937  011346  013702  001170                MOV     $VECT1,R2       ;BUILD DEVICE VECTORS IN R2
1938  011352  042702  177007                BIC     #^C<770>,R2     ;STRIP AWAY OTHER INFORMATION
1939  011356  012704  001204                MOV     #$DDWD,R4       ;POINT TO THE BEGINNING OF DEVICE PARAMETERS
1940  011362  013705  001176                MOV     $DEVM,R5        ;GET THE MAP OF ACTIVE DEVICES
1941  011366  105037  001414                CLRB    DZVNUM          ;INITIALIZE THE NO. OF ACTIVE DEVICES
1942  011372  005037  001410                CLR     SAVACTV         ;CLEAR THE ACTIVE BIT MAP
1943  011376  006005                1$:     ROR     R5              ;GET A DEVICE SELECTION BIT
1944  011400  103407                        BCS     3$              ;IF IT IS SELECTED, GO SET UP A MAP
1945  011402  001422                        BEQ     5$              ;IF NO MORE ARE SELECTED, GET OUT OF SETUP
1946  011404  005724                        TST     (R4)+           ;POINT TO NEXT DEVICE DESCRIPTOR
1947  011406  062701  000010        2$:     ADD     #10,R1          ;SET UP THE NEXT ADDRESS
1948  011412  062702  000010                ADD     #10,R2          ;SET UP THE NEXT VECTOR GROUP
1949  011416  000767                        BR      1$              ;GO SEE IF MORE DEVICES REMAIN
1950  011420  006137  001410        3$:     ROL     SAVACTV         ;SET BIT IN ACTIVE DEVICE MAP
1951  011424  105237  001414                INCB    DZVNUM          ;INCREMENT NO. OF ACTIVE DEVICES
1952  011430  010120                        MOV     R1,(R0)+        ;LOAD DEVICE ADDRESS
1953  011432  010220                        MOV     R2,(R0)+        ;LOAD THE VECTOR ADDRESS
1954  011434  013720  001200                MOV     $CDW1,(R0)+     ;GET THE NUMBER OF LINES IN OPERATION
1955  011440  012420                        MOV     (R4)+,(R0)+     ;LOAD DEVICE PARAMETERS
1956  011442  013720  001202                MOV     $CDW2,(R0)+     ;LOAD DEFAULT TESTING MODE
1957  011446  000757                        BR      2$              ;GO BUILD THE NEXT ADDRESS
1958  011450  012710  177777        5$:     MOV     #-1,(R0)        ;TERMINATE THE DEVICE MAP
1959  011454  012737  001142  001304        MOV     #$SAREG,SWR     ;SET TO SOFTWARE APT SWITCH REGISTER
1960  011462  000207                        RTS     PC              ;RETURN TO PRINT STATUS TABLE
1961
1962
1963                                        ;*ROUTINE USED TO "AUTO SIZE" THE DZV11
1964                                        ;*CSR AND VECTOR.
1965                                        ;*NOTE:  THE CSR MAY BE ANY WHERE IN THE FLOATING
1966                                        ;*       ADDRESS RANGE (160000:163770)
1967                                        ;*       AND THE VECTOR MAY BE ANY WHERE IN THE
1968                                        ;*       FLOATING VECTOR RANGE (300:770)
1969                                        ;*
1970
1971  011464                        AUTO.SIZE:
1972  011464  000005                        RESET                   ;INSURE A BUS INIT.
1973  011466  105337  001422                DECB    INIFLG          ;SHOW THAT I WAS HERE
1974  011472  012702  001500        CSRMAP: MOV     #DZV.MAP,R2     ;LOAD MAP POINTER.
1975  011476  012703  001204                MOV     #$DDWD,R3       ;POINT TO ETABLE DEVICE DESCRIPTOR WORDS
1976  011502  005022                1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
1977  011504  022702  001740                CMP     #DZV.END,R2     ;ALL DONE?
1978  011510  001374                        BNE     1$              ;BR IF NO
1979  011512  105037  001414                CLRB    DZVNUM          ;SET OCTAL NUMBER OF DZV11'S TO 0
1980  011516  012702  001500                MOV     #DZV.MAP,R2
1981  011522  012701  160000                MOV     #160000,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
1982  011526  012737  011772  000004        MOV     #6$,@#4         ;SET FOR NON-EXISTENT DEVICE TIME OUT
1983  011534  052711  000040        2$:     BIS     #BIT5,(R1)      ;TRY TO SET MASTER SCAN ENABLE
1984  011540  052761  000017  000004        BIS     #17,4(R1)       ;TRY TO TRANSMIT ON ANY LINE
1985  011546  005000                        CLR     R0              ;USE R0 AS A COUNTER
1986  011550  005711                7$:     TST     (R1)            ;HAS TRANSMITTER READY COME UP?
```

```
1987  011552  100403                        BMI     8$              ;IF SO, GO GET A FINAL CHECK
1988  011554  005300                        DEC     R0              ;REDUCE COUNT. TIME UP?
1989  011556  001374                        BNE     7$              ;IF NOT, KEEP WAITING
1990  011560  000437                        BR      3$              ;ASSUME IT'S NOT A DZV11
1991  011562  032761  000017  000004  8$:   BIT     #17,4(R1)       ;ARE ANY TCR BITS  STILL SET? THEY SHOULD BE
1992  011570  001433                        BEQ     3$              ;IF IT'S NOT, ASSUME IT'S NOT A DZV11
1993  011572  032711  000040                BIT     #BIT5,(R1)      ;IS MASTER SCAN ENABLE STILL SET?
1994  011576  001430                        BEQ     3$              ;IF NOT, ASSUME IT'S NOT A DZV11
1995  011600  052711  000020                BIS     #20,(R1)        ;SET DEVICE CLEAR
1996  011604  000240                        NOP
1997  011606  032711  000040                BIT     #40,(R1)        ;DID SCANNER CLEAR
1998  011612  001022                        BNE     3$              ;IF NOT ASSUME IT IS NOT DZV
1999  011614  005061  000004                CLR     4(R1)           ;GET RID OF TCR BITS
2000                          ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DZV11 CSR ADDRESS.
2001  011620  010122                        MOV     R1,(R2)+        ;STORE CSR IN CORE TABLE.
2002  011622  005722                        TST     (R2)+           ;POP OVER VECTOR STORE AREA
2003  011624  012722  000017                MOV     #17,(R2)+       ;SET THE DEFAULT LINE SELECTION PARAMETER
2004  011630  012712  017470                MOV     #17470,(R2)     ;SET THE DEFAULT PARAMETERS
2005  011634  012223                        MOV     (R2)+,(R3)+     ;COPY PARAMETERS INTO ETABLE DESCRIPTOR
2006  011636  005022                        CLR     (R2)+           ;SET THE DEFAULT MODE OF OPERATION
2007  011640  012712  177777                MOV     #-1,(R2)        ;TERMINATE LIST
2008  011644  105237  001414                INCB    DZVNUM          ;UPDATE DEVICE COUNTER
2009  011650  122737  000020  001414        CMPB    #20,DZVNUM      ;ARE MAX. NO. OF DEV FOUND?
2010  011656  001405                        BEQ     10$             ;YES DON'T LOOK FOR ANY MORE.
2011  011660  062701  000010          3$:   ADD     #10,R1          ;UPDATE CSR POINTER ADDRESS
2012  011664  022701  164000                CMP     #164000,R1
2013  011670  001321                        BNE     2$              ;BR IF MORE ADDRESS TO CHECK.
2014  011672                          10$:
2015  011672  105737  001414                TSTB    DZVNUM          ;WERE ANY DZV11'S FOUND AT ALL?
2016  011676  001430                        BEQ     5$              ;ERROR AUTO SIZER FOUND NO DZV11'S IN THIS SYS.
2017  011700  113701  001414                MOVB    DZVNUM,R1
2018  011704  012737  000001  001410        MOV     #1,SAVACTV      ;CREATE A BIT MAP OF
2019  011712  005301                  4$:   DEC     R1              ;THE DEVICES IN THE SYSTEM
2020  011714  001404                        BEQ     9$
2021  011716  000261                        SEC
2022  011720  006137  001410                ROL     SAVACTV
2023  011724  000772                        BR      4$
2024  011726  013737  001500  001174  9$:   MOV     DZCR0,$BASE     ;POINT TO THE ADDRESS OF FIRST DEVICE
2025  011734  013737  001510  001202        MOV     MAN70,$CDW2     ;INDICATE TO ETABLE WHAT MODE IS BEING USED
2026  011742  012737  000006  000004  99$:   MOV     #6,@#4          ;RESTORE TRAP VECTOR
2027  011750  013737  001410  001176        MOV     SAVACTV,$DEVM   ;SAVE ACTIVE REGISTER
2028  011756  000410                        BR      VECMAP          ;GO FIND THE VECTOR NOW.
2029  011760  104402  007513          5$:   TYPE    ,MERR2          ;NOTIFY OPR THAT NO DZV11'S FOUND.
2030  011764  005000                        CLR     R0              ;MAKE DATA DISPLAY ZERO
2031  011766  000000                        HALT                    ;STOP THE SHOW
2032  011770  000776                        BR      .-2             ;DISABLE CONT. SW.
2033  011772  012716  011660          6$:   MOV     #3$,(SP)        ;ENTERED BY NON-EXISTENT TIME-OUT
2034  011776  000002                        RTI                     ;RETURN TO MAINSTREAM
2035
2036  012000  012737  000200  000022  VECMAP: MOV     #MASK,@#22      ;SET IOT TRAP PRIORITY
2037  012006  012737  012122  000020        MOV     #4$,@#20        ;SET IOT TRAP VECTOR
2038  012014  012702  001500                MOV     #DZV.MAP,R2     ;SET SOFTWARE POINTER
2039  012020  012700  000300                MOV     #300,R0         ;FLOATING VECTORS START HERE.
2040  012024  012701  000302                MOV     #302,R1         ;PC OF IOT INSTR.
2041  012030  010120                  1$:   MOV     R1,(R0)+        ;START FILLING VECTOR AREA
2042  012032  012721  000004                MOV     #4,(R1)+        ;WITH .+2; IOT
```

```
2043  012036  022021                    CMP     (R0)+,(R1)+     ;ADD 2 TO R0 +R1
2044  012040  020127  001000            CMP     R1,#1000        ;HAS THE VECTOR AREA BEEN EXCEEDED?
2045  012044  101771                    BLOS    1$              ;BR IF MORE TO FILL
2046  012046  013704  001410            MOV     SAVACTV,R4      ;STORE TEMPORARILY
2047  012052  006004            2$:     ROR     R4              ;BRING OUT A BIT
2048  012054  103036                    BCC     5$              ;BR IF ALL DONE
2049  012056  106427  000000            MTPS    #0              ;ZERO CPU PRIO
2050  012062  012772  040040  000000    MOV     #BIT14+BIT5,2(R2)       ;SET TIE AND MAS SCAN
2051  012070  011201                    MOV     (R2),R1         ;GET CSR
2052  012072  112761  000017  000004    MOVB    #17,4(R1)       ;SET THE TCR BITS FOR ALL LINES
2053                                    ;ATTEMPT TO FORCE AN INTERRUPT
2054  012100  005200                    INC     R0              ;STALL
2055  012102  001376                    BNE     .-2             ;       FOR TIME TO INTERRUPT
2056  012104  012762  000300  000002    MOV     #300,2(R2)      ;NO INTERRUPT ASSUME 300 AND FIX DZV11 LATER
2057  012112  000005                    RESET                   ;INIT
2058  012114  062702  000012    3$:     ADD     #12,R2          ;POP SOFTWARE POINTER
2059  012120  000754                    BR      2$              ;KEEP GOING
2060  012122  011662  000002    4$:     MOV     (SP),2(R2)      ;GET VECTOR ADDRESS
2061  012126  162762  000010  000002    SUB     #10,2(R2)       ;POINT BACK TO THE CORRECT VECTOR
2062  012134  042762  000007  000002    BIC     #7,2(R2)        ;CLEAR JUNK
2063  012142  022626                    POP2SP                  ;POP IOT JUNK OFF STACK
2064  012144  012716  012114            MOV     #3$,(SP)        ;SET FOR RETURN
2065  012150  000002                    RTI
2066  012152  013737  001502  001170  5$:     MOV     DZVC0,$VECT1    ;COPY VECTOR OF FIRST DEVICE INTO ETABLE
2067  012160  012737  004300  000020    MOV     #.SCOPE,IOTVEC  ;RESTORE THE SCOPE TRAP
2068  012166  000207                    RTS     PC              ;ALL DONE WITH "AUTO SIZING"
2069
```

```
                                        ;************************ TEST 1 *******************************
                                        ;#THIS TEST VERIFIES OVERRUN AND SILO ALARM
                                        ;#ONE LINE AT A TIME - BASED UPON VALID LINES
                                        ;#AS EACH OF THE FIRST 16 CHARS ARE SENT; SILO ALARM IS
                                        ;#TESTED TO BE CLEARED. ON THE 16TH CHAR THE PROGRAM THEN
                                        ;#EXPECTS SILO ALARM TO SET. THEN THE ENTIRE
                                        ;#SILO IS FILLED AND AN OVERRUN IS EXPECTED ON THE 65TH
                                        ;#CHAR PULLED OUT OF THE SILO.
                                        ;#ERROR PRINTOUTS WILL REPORT TRANSMITTING LINE NO.
                                        ;#USING SWITCH NINE FOR THIS TEST SENDS 20. CHARACTERS
                                        ;#ON DZV LINE PREVIOUSLY SELECTED CONTINUOUSLY WHILE SW09=1.
                                        ;#USED TO SCOPE SILO ALARM PULSES, ETC.
                                ;:*  TEST 1
                                ;:*************************************************************************
012170  000004                TST1:   SCOPE
012172  012737  000001  001246        MOV     #1,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
012200  012737  012632  001362        MOV     #TST2,NEXT      ;POINT TO THE START OF THE NEXT TEST
012206  012737  012546  001364        MOV     #18$,LOCK       ;SET FOR LOOP
012214  104417                      DCLASM                  ;SET DCLR IN CSR AND SET MNTFLG
012216  104421                      LPRSET                  ;LOAD LINE PARAMETERS
012220  005037  001374              CLR     SAVLIN          ;INIT LINE INDICATOR
012224  104422                      BUFSET                  ;ZERO DATA BUFFER
012226  012702  000001              MOV     #1,R2           ;LINE POINTER
012232  052777  010040  167550      BIS     #MSENAB!SILOEN,@DZVCSR  ;START SCANNER & SET SILO ENABLE
012240  030237  001366        3$:   BIT     R2,LINE         ;VALID LINE?
012244  001533                      BEQ     21$             ;IF NOT GO TO NEXT LINE
012246  013700  001374              MOV     SAVLIN,R0       ;MAKE OFFSET
012252  006300                      ASL     R0              ;MAKE POWER OF TWO
012254  010277  167544              MOV     R2,@DZVTCR      ;SET TCR BIT
012260  105777  167524        4$:   TSTB    @DZVCSR         ;REC DONE = 1 ?
012264  100001                      BPL     .+4
012266  104020                      ERROR   20              ;REC DONE SHOULD NOT = 1
012270  005003                      CLR     R3              ;SET CHARACTER COUNT
012272  005004                5$:   CLR     R4
012274  005777  167510        6$:   TST     @DZVCSR         ;IS TRDY SET?
012300  100404                      BMI     7$              ;IF YES, LOAD CHAR.
012302  104414                      DELAY                   ;WAIT FOR TRDY TO SET
012304  005204                      INC     R4              ;INC DELAY COUNTER
012306  001372                      BNE     6$
012310  104003                      ERROR   3               ;*TRDY FAILED TO SET
012312  116077  001426  167514  7$: MOVB    TDO(R0),@DZVTDR ;LOAD A CHARACTER
012320  005260  001426              INC     TDO(R0)         ;SET UP NEXT CHARACTER
012324  020327  000017              CMP     R3,#15.         ;16 CHARACTERS ?
012330  103006                      BHIS    8$
012332  032777  020000  167450      BIT     #SILOAL,@DZVCSR ;SILO ALARM = 0 ?
012340  001413                      BEQ     10$             ;YES
012342  104013                      ERROR   13              ;*SILO ALARM SHOULD NOT = 1
                                                            ;UNTIL 16. DATA CHARACTERS
012344  000411                      BR      10$
012346  005004                8$:   CLR     R4
012350  032777  020000  167432  9$: BIT     #SILOAL,@DZVCSR
012356  001004                      BNE     10$
012360  104414                      DELAY
012362  005204                      INC     R4
012364  001371                      BNE     9$
```

```
                                        ERROR   14              ;*SILO ALARM FAILED TO SET!
                                                                ;SILO ALARM SHOULD =1 AFTER 16.
                                                                ;DATA CHARACTERS
012370  005203                  10$:   INC     R3              ;INC CHAR COUNT
012372  022703  000102                CMP     #66.,R3         ;FINISHED SENDING CHARACTERS ?
012376  001335                        BNE     5$              ;NO
012400  005004                        CLR     R4
012402  104414                        DELAY
012404  105204                        INCB    R4
012406  001375                        BNE     .-4
                                      ;NOW LETS READ THE SILO
012410  013705  001374                MOV     SAVLIN,R5       ;MAKE EXPECTED LINE #
012414  005737  001372                TST     MODE            ;IS THIS TEST IN STAGGERED MODE?
012420  100006                        BPL     13$             ;IF NOT, SKIP STAGGERED SETUP

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

012422  006205                        ASR     R5              ;GET THE LAST BIT INTO THE CARRY BIT
012424  103402                        BCS     11$             ;IF IT IS SET, GO CLEAR IT
012426  000261                        SEC                     ;IF IT IS CLEAR SET IT HERE
012430  000401                        BR      12$             ;SKIP THE CLEARING
012432  000241                 11$:   CLC                     ;CLEAR THE CARRY BIT (INVERSION OF LINE PARITY)
012434  006105                 12$:   ROL     R5              ;GET THE NEW BIT BACK INTO R5
012436  000305                 13$:   SWAB    R5              ;PUT IN UPPER BYTE
012440  052705  100000                BIS     #DVALID,R5      ;ADD DATA VALID
012444  017704  167344         14$:   MOV     @DZVRBUF,R4     ;ACTUAL
012450  020405                        CMP     R4,R5           ;ACTUAL VS. EXPECTED
012452  001401                        BEQ     15$             ;YES
012454  104006                        ERROR   6               ;*DATA/CONTENTS DID NOT COMPARE
012456  032777  020000  167324  15$:  BIT     #SILOAL,@DZVCSR ;SILO ALARM= 0 ?
012464  001401                        BEQ     16$             ;YES
012466  104016                        ERROR   16              ;READING DZVRBUF DID NOT CLEAR SILO ALARM
012470  005205                 16$:   INC     R5              ;UP CHARACTER
012472  120527  000077                CMPB    R5,#63.         ;LAST SILO CHAR ?....64TH CHAR
012476  101762                        BLOS    14$
012500  005205                        INC     R5              ;ADD 1 MORE FOR THE CLOBBERED CHAR
012502  052705  040000                BIS     #OVRRUN,R5      ;ADD OVERRUN TO EXPECTED
012506  120527  000101                CMPB    R5,#65.         ;LAST CHARACTER ?
012512  001754                        BEQ     14$
012514  017704  167274                MOV     @DZVRBUF,R4     ;FOR GOOD MEASURE
012520  005704                        TST     R4              ;DATA VALID SHOULD = 0
012522  100001                        BPL     17$             ;YES
012524  104017                        ERROR   17              ;DATA VALID SHOULD = 0
012526  040277  167272         17$:   BIC     R2,@DZVTCR      ;CLR TCR BIT
012532  104401                        SCOP1                   ;LOOP?
012534  005237  001374         21$:   INC     SAVLIN          ;INC EXPECTED LINE
012540  104420                        SHIFT                   ;NEXT LINE
012542  000137  012240                JMP     3$              ;YES

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

012546  052777  010040  167234  18$:  BIS     #MSENAB!SILOEN,@DZVCSR  ;SETUP DEVICE
012554  012777  012622  167262        MOV     #20$,@DZVTIV    ;SETUP TRANSMITTER VECTOR
012562  012701  000024                MOV     #20.,R1         ;TEMPORARY COUNT OF CHARACTER BURST
```

```
2182  012566  050277  167232                BIS     R2,@DZVTCR      ;ENABLE LINE
2183  012572  052777  040000  167210        BIS     #TIE,@DZVCSR    ;ENABLE INTERRUPTS
2184  012600  106427  000000                MTPS    #0              ;LOWER PRIORITY
2185  012604  000001                19$:    WAIT                    ;ALLOW INTERRUPTS
2186  012606  077102                        SOB     R1,19$          ;REDUCE COUNT. ALL CHARACTERS SENT?
2187  012610  042777  050040  167172        BIC     #SILOEN!MSENAB!TIE,@DZVCSR ;RESET SILO COUNTER, CLEAR STROBE
2188  012616  104401                        SCOP1                   ;LOOP AGAIN?
2189  012620  000742                        BR      17$             ;IF NOT, RETURN TO WHERE YOU LEFT OFF
2190  012622  112777  000252  167204  20$:  MOVB    #252,@DZVTDR    ;SEND A CHARACTER
2191  012630  000002                        RTI                     ;ALLOW MORE CHARACTERS TO COME
2192                                        ;********************** TEST 2 ******************************
2193                                        ;*THIS TEST THAT "SILO ENABLE" WILL INHIBIT
2194                                        ;*RECEIVER INTERRUPTS AND THAT ON THE
2195                                        ;*16TH CHAR THAT "SILO ALARM" WILL CAUSE AN
2196                                        ;*INTERRUPT WITH "RIE" SET.
2197                                        ;*THIS WILL DO ALL SELECTED LINES ONE AT A TIME.
2198                                        ;*ERROR PRINTOUTS WILL REPORT TRANSMITTING LINE NO.
2199                                ;:*  TEST 2
2200                                ;:*************************************************************
2201  012632  000004                TST2:   SCOPE
2202  012634  012737  000002  001246        MOV     #2,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
2203  012642  012737  013132  001362        MOV     #TST3,NEXT      ;POINT TO THE START OF THE NEXT TEST
2204  012650  012737  012674  001364        MOV     #3$,LOCK        ;SET FOR LOOP
2205  012656  104417                        DCLASM                  ;SET DCLR IN CSR AND SET MNTFLG
2206  012660  104421                        LPRSET                  ;LOAD LINE PARAMETERS
2207  012662  005037  001374                CLR     SAVLIN          ;INIT LINE INDICATOR
2208  012666  104422                        BUFSET                  ;ZERO DATA BUFFER
2209  012670  012702  000001                MOV     #1,R2           ;LINE POINTER
2210  012674  012777  013104  167136  3$:   MOV     #11$,@DZVRIV    ;SET FOR UNEXPECTED INTER.
2211  012702  012777  000200  167132        MOV     #MASK,@DZVRIS   ;SET PRIO.
2212  012710  052777  010140  167072        BIS     #MSENAB!SILOEN!RIE,@DZVCSR
2213                                                                ;START SCANNER & SET SILO ENABLE
2214  012716  030237  001366                BIT     R2,LINE         ;VALID LINE?
2215  012722  001477                        BEQ     18$             ;IF NOT GO TO NEXT LINE
2216  012724  005777  167064                TST     @DZVRBUF        ;EMPTY THE SILO
2217  012730  100775                        BMI     .-4             ;BR IF DATA VALID IS SET!
2218  012732  106427  000000                MTPS    #0              ;SET PROCESSOR PRIORITY TO 0
2219  012736  013700  001374                MOV     SAVLIN,R0       ;MAKE OFFSET
2220  012742  006300                        ASL     R0              ;MAKE POWER OF TWO
2221  012744  010277  167054                MOV     R2,@DZVTCR      ;SET TCR BIT
2222  012750  005004                4$:     CLR     R4
2223  012752  005777  167032        6$:     TST     @DZVCSR
2224  012756  100404                        BMI     7$
2225  012760  104414                        DELAY
2226  012762  005204                        INC     R4
2227  012764  001372                        BNE     6$
2228  012766  104003                        ERROR   3               ;*TRDY FAILED TO SET
2229  012770  116077  001426  167036  7$:   MOVB    TDO(R0),@DZVTDR ;LOAD A CHARACTER
2230  012776  005260  001426                INC     TDO(R0)         ;SET UP NEXT CHARACTER
2231  013002  022760  000017  001426        CMP     #15.,TDO(R0)    ;15 CHARS YET?
2232  013010  001406                        BEQ     8$
2233  013012  032777  020000  166770        BIT     #SILOAL,@DZVCSR ;SILO ALARM = 0 ?
2234  013020  001401                        BEQ     .+4             ;YES
2235  013022  104013                        ERROR   13              ;*SILO ALARM SHOULD NOT = 1
2236                                                                ;UNTIL 16. DATA CHARACTERS
2237  013024  000752                        BR      6$
```

```
2238  013026  012777  013112  167004  8$:     MOV     #12$,@DZVRIV    ;SET NEW VECTOR
2239  013034  005777  166750          1$:     TST     @DZVCSR         ;READY FOR 16TH CHAR
2240  013040  100375                          BPL     .-4
2241  013042  016077  001426  166764          MOV     TDO(R0),@DZVTDR ;LOAD THE 16TH CHAR.
2242  013050  005004                          CLR     R4
2243  013052  032777  020000  166730  9$:     BIT     #SILOAL,@DZVCSR
2244  013060  001005                          BNE     10$
2245  013062  104414                          DELAY
2246  013064  005204                          INC     R4
2247  013066  001371                          BNE     9$
2248  013070  104014                          ERROR   14              ;*SILO ALARM FAILED TO SET!
2249  013072  000410                          BR      17$             ;SILO ALARM SHOULD =1 AFTER 16.
2250                                                                  ;DATA CHARACTERS
2251  013074  000240          10$:    NOP                             ;STALL
2252  013076  000240                  NOP
2253  013100  104027                          ERROR   27              ;SILO ALARM NOT INTERRUPTING.
2254  013102  000404                          BR      17$             ;CONTINUE TEST.
2255  013104  022626          11$:    POP2SP                          ;FAKE RTI
2256  013106  104012                          ERROR   12              ;RX SHOULD NOT INTERRUPT
2257  013110  000401                          BR      17$             ;CONTINUE
2258  013112  022626          12$:    POP2SP                          ;GOOD INTERRUPT TO HERE.
2259  013114  040277  166704  17$:    BIC     R2,@DZVTCR              ;CLR TCR BIT
2260  013120  104401                          SCOP1                   ;LOOP?
2261  013122  005237  001374  18$:    INC     SAVLIN                  ;INC EXPECTED LINE
2262  013126  104420                          SHIFT                   ;NEXT LINE
2263  013130  000661                          BR      3$              ;YES
```

```
                                        ;************************ TEST 3 ********************************
                                        ;*THIS TEST RUNS ALL LINES FULL BORE
                                        ;*BASED UPON QUALIFIED LINES
                                        ;*..THIS IS AN INTERRUPT TEST ON THE RECEIVER AND
                                        ;*TRANSMITTER
                                ;:*  TEST 3
                                ;:*************************************************************************
013132  000004                  TST3:   SCOPE
013134  012737  000003  001246          MOV     #3,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
013142  012737  013674  001362          MOV     #TST4,NEXT      ;POINT TO THE START OF THE NEXT TEST
013150  104417                          DCLASM                  ;SET DCLR IN CSR AND SET MNTFLG
013152  013737  001366  013672          MOV     LINE,RXTCR      ;SET IMAGE OF TCR BITS
013160  013737  001366  013412          MOV     LINE,TXTCR      ;SET IMAGE OF TCR BITS
013166  104421                          LPRSET                  ;LOAD LINE PARAMETERS
013170  104422                          BUFSET                  ;ZERO DATA BUFFER
013172  012777  013414  166640          MOV     #RXSVC,@DZVRIV  ;SET UP REC INTR VECTOR
013200  012777  000200  166634          MOV     #MASK,@DZVRIS   ;STATUS
013206  012777  013300  166630          MOV     #TXSVC,@DZVTIV  ;SET UP TRANS INTR VECTOR
013214  012777  000200  166624          MOV     #MASK,@DZVTIS   ;STATUS
013222  052777  040140  166560          BIS     #MSENAB!RIE!TIE,@DZVCSR ;SET MASTER SCAN ENABLE
013230  113777  001366  166566          MOVB    LINE,@DZVTCR    ;SET TCR BITS
013236  106427  000000                  MTPS    #CLEAR          ;ALLOW INTERRUPTS

013242  005037  013276          SNAP:   CLR     4$              ;CLEAR DELAY COUNTER
013246  104414                  2$:     DELAY                   ;WAIT FOR RECEIVERS TO FINISH
013250  105737  013672                  TSTB    RXTCR           ;WAIT FOR ALL RECIEVERS TO FINISH
013254  001002                          BNE     3$              ;
013256  000137  013620                  JMP     OUT
013262  005237  013276          3$:     INC     4$              ;INCREMENT DELAY COUNTER
013266  001367                          BNE     2$              ;DELAY FINISHED?
013270  104007                          ERROR   7               ;*TRANSMITTER FAILED TO INTERRUPT
013272  104011                          ERROR   11              ;*RECEIVER FAILED TO INTERRUPT
013274  104400                          ADVANCE                 ;LEAVE THIS TEST
013276  000000                  4$:     0

                                ;TRANS INTR SVC ROUTINE
013300  117703  166506          TXSVC:  MOVB    @#DZVCSR,R3     ;FIND LINE NO.
013304  042703  177774                  BIC     #^C<3>,R3       ;ISOLATE LINE NO.
013310  010304                          MOV     R3,R4           ;SAVE LINE NO.
013312  010337  001374                  MOV     R3,SAVLIN       ;SAVE LINE NO.
013316  005777  166466                  TST     @DZVCSR         ;TRANS READY SET ?
013322  100401                          BMI     .+4
013324  104003                          ERROR   3               ;*TRANSMITTER FAILED
013326  012702  000001                  MOV     #1,R2           ;SET UP POSITION POINTER
013332  105303                  3$:     DECB    R3              ;IS IT THIS LINE ?
013334  100402                          BMI     4$              ;YES
013336  006302                          ASL     R2              ;UP THE LINE #
013340  000774                          BR      3$              ;GO 'ROUND AGAIN
013342  030237  001366          4$:     BIT     R2,LINE         ;VALID LINE?
013346  001001                          BNE     .+4             ;YES
013350  104015                          ERROR   15              ;NO,INVALID LINE!!!!
013352  030237  013412                  BIT     R2,TXTCR        ;DATA FINISHED?
013356  001003                          BNE     6$              ;IF NOT SEND CHAR.
013360  040277  166440                  BIC     R2,@DZVTCR      ;CLEAR TCR BIT
013364  000411                          BR      5$              ;RETURN
```

```
2320  013366  006304                  6$:     ASL     R4                      ;MAKE POWER OF 2
2321  013370  116477  001426  166436          MOVB    TDO(R4),@DZVTDR         ;LOAD CHARACTER
2322  013376  105264  001426                  INCB    TDO(R4)                 ;SET UP NEXT CHARACTER
2323  013402  001002                          BNE     5$                      ;LAST CHARACTER ?
2324  013404  040237  013412                  BIC     R2,TXTCR                ;INDICAT LINE FINISHED
2325  013410  000002                  5$:     RTI
2326
2327  013412  000000                  TXTCR:  0
2328
2329                                  ;REC INTR SVC ROUTINE
2330  013414  105777  166370          RXSVC:  TSTB    @DZVCSR                 ;REC DONE ?
2331  013420  100401                          BMI     .+4                     ;YES
2332  013422  104004                          ERROR   4                       ;FALSE INTERRUPT
2333  013424  032777  020000  166356          BIT     #SILOAL,@DZVCSR         ;SILO ALARM?
2334  013432  001401                          BEQ     .+4                     ;NO
2335  013434  104013                          ERROR   13                      ;SILO ALARM SHOULD NOT =1
2336  013436  017704  166352                  MOV     @DZVRBUF,R4             ;SAVE IT
2337  013442  010403                          MOV     R4,R3
2338  013444  000303                          SWAB    R3
2339  013446  042703  177774                  BIC     #^C(3),R3               ;STRIP JUNK
2340  013452  010337  001374                  MOV     R3,SAVLIN               ;SAVE LINE NUMBER
2341  013456  005704                          TST     R4                      ;DATA VALID?
2342  013460  100401                          BMI     4$                      ;IF YES SKIP ERROR PRINTOUT
2343  013462  104023                          ERROR   23                      ;YOU LOSE  ...DATA VALID WAS'NT SET
2344  013464  032704  040000          4$:     BIT     #OVRRUN,R4              ;TEST FOR OVERRUN
2345  013470  001401                          BEQ     1$                      ;IF NO OVERRUN SKIP ERROR
2346  013472  104024                          ERROR   24                      ;DATA OVERRUN
2347  013474  032704  020000          1$:     BIT     #FRMERR,R4              ;DATA FRAMING ERROR
2348  013500  001401                          BEQ     2$                      ;IF NO FRAMING ERROR CONTINUE
2349  013502  104025                          ERROR   25                      ;FRAMING ERROR
2350  013504  032704  010000          2$:     BIT     #PARER,R4               ;TEST FOR PARITY ERROR
2351  013510  001401                          BEQ     3$                      ;BRANCH IF NO ERROR
2352  013512  104026                          ERROR   26                      ;TYPE OUT PARITY ERROR
2353  013514  012702  000001          3$:     MOV     #1,R2                   ;SET UP POSITION POINTER
2354  013520  105303                  5$:     DECB    R3
2355  013522  100402                          BMI     6$
2356  013524  006302                          ASL     R2                      ;RE POSITION POINTER
2357  013526  000774                          BR      5$                      ;GO 'ROUND AGAIN
2358  013530  030237  001366          6$:     BIT     R2,LINE                 ;LINE VALID ?
2359  013534  001001                          BNE     .+4                     ;YES
2360  013536  104015                          ERROR   15                      ;INVALID LINE #
2361  013540  013703  001374                  MOV     SAVLIN,R3               ;GET THE LINE NUMBER AGAIN
2362  013544  006303                          ASL     R3                      ;USE R3 AS A POINTER IN THE DATA TABLE
2363  013546  126304  001436                  CMPB    TRO(R3),R4              ;DOES THE DATA CHARACTER COMPARE ?
2364  013552  001410                          BEQ     7$                      ;YES
2365  013554  013705  001374                  MOV     SAVLIN,R5               ;MOVE LINE NO INTO EXPECTED
2366  013560  000305                          SWAB    R5                      ;ADJUST TO HIGH BYTE
2367  013562  052705  100000                  BIS     #DVALID,R5              ;SET DVALID IN EXPECTED
2368  013566  056305  001436                  BIS     TRO(R3),R5              ;SET DATA IN EXPECTED
2369  013572  104005                          ERROR   5                       ;*NO, DATA DOES NOT COMPARE
2370  013574  005263  001436          7$:     INC     TRO(R3)                 ;SET UP FOR NEXT CHARACTER
2371  013600  105763  001436                  TSTB    TRO(R3)                 ;ALL CHARS DONE?
2372  013604  001002                          BNE     .+6
2373  013606  040237  013672                  BIC     R2,RXTCR                ;ZERO LINE DONE INDICATOR.
2374  013612  012716  013242                  MOV     #SNAP,(SP)              ;RESET THE BACKGROUND TIMING LOOP
2375  013616  000002                          RTI
```

```
                                        ;FINISH UP ROUTINE
013620  106427  000200        OUT:    MTPS    #MASK           ;STOP ALL INTERRUPTS
013624  104413                        DEVICE.CLR              ;CLEAR ALL INTERRUPTS AWAY
013626  005003                        CLR     R3
013630  005037  001374                CLR     SAVLIN
013634  012702  000001                MOV     #1,R2
013640  030237  001366        1$:     BIT     R2,LINE         ;VALID LINE ?
013644  001405                        BEQ     2$              ;NO
013646  022763  000400  001436        CMP     #400,TRO(R3)    ;RECEIVED A BINARY COUNT PATTERN ?
013654  001401                        BEQ     .+4             ;YES
013656  104030                        ERROR   30              ;THE LINE FAILED TO RECEIVE A FULL
                                                              ;BINARY COUNT PATTERN
013660  005237  001374        2$:     INC     SAVLIN          ;SET UP FOR NEXT LINE
013664  005723                        TST     (R3)+           ;ADD 2
013666  104420                        SHIFT                   ;SET UP NEXT LINE POINTER
013670  000763                        BR      1$              ;FINISHED ?
013672  000000                RXTCR:  0                       ;RX IMAGE OF TCR BITS

                        ;********************** TEST 4 ******************************
                        ;#DZV11 RELATIVE TIMING TEST.
                        ;#EACH SELECTED LINE WILL IN TURN RUN 16. CHARS
                        ;#AT ALL BAUD RATES AND THEN THE HIGHEST BAUD
                        ;#WITH ALL CHAR LENGTHS. EACH NEW PARAMETER SHOULD
                        ;#DECREASE IN TIME FROM THE PREVIOUS PARAMETERS SELECTED.
                        ;#THE TIME IS CHECKED AGAINST THE LAST PARAMETER USED
                        ;# AND A LOWER TIME IS EXPECTED ON THE CURRENT PARAMETER.
                        ;#PARAMETERS ARE:
                        ;#  EIGHT BITS/PER/CHAR - TWO STOP BITS AT
                        ;#       50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000
                        ;#       2400, 3600, 4800, 7200, 9600 BAUD.
                        ;#  19.2 K BAUD - TWO STOP BITS AT
                        ;#       SEVEN, SIX, FIVE BITS/PER/CHAR.
                        ;#AFTER EACH LINE HAS FINISHED ALL THE ABOVE PARAMETERS
                        ;#THE NEXT SELECTED LINE IS THEN TESTED.
                        ;#WHEN RUNNING UNDER THE APT MANUFACTURING SYSTEM
                        ;#THIS TEST IS ONLY RUN THE FIRST PASS
                    ;:*  TEST 4
                    ;:*************************************************************
013674  000004        TST4:   SCOPE
013676  012737  000004  001246          MOV     #4,@#TSTNM      ;LOAD THE NUMBER OF THIS TEST
013704  012737  014312  001362          MOV     #T5,@#NEXT      ;POINT TO THE START OF THE NEXT TEST
013712  012737  014032  001364          MOV     #3$,@#LOCK      ;USE THIS ADDRESS IF A TIGHT SCOPE LOOP IS SELECTED
013720  132737  000001  001140          BITB    #1,@#ENV        ;RUNNING UNDER APT?
013726  001405                        BEQ     10$             ;IF NOT CONTINUE WITH TEST
013730  005737  001126                TST     @#PASS          ;IF YES IS THIS FIRST PASS
013734  001402                        BEQ     10$             ;IF NOT 1ST PASS SKIP TEST
013736  000177  165420                JMP     @#NEXT
013742  012737  000002  001354  10$:  MOV     #2,@#TIMES      ;SET UP FOR 2 ITERATIONS
013750  005037  015314                CLR     @#OFFSET        ;RESET THIS VARIABLE
013754  005037  001374                CLR     @#SAVLIN        ;RESET LINE NUMBER INDICATOR
013760  005037  001376                CLR     @#XMTLIN        ;USE THIS WORD TO TELL WHAT LINE TRANSMITTED
013764  012702  000001                MOV     #1,R2           ;USE R2 AS A BIT POINTER
013770  012703  010070                MOV     #RCVON!SSO!EIGHT!TWOSTOP,R3 ;BUILD TEMPORARY PARAMETERS
```

```
2432  013774  030237  001366          1$:     BIT     R2,LINE         ;IS THIS LINE ACTIVE?
2433  014000  001014                          BNE     3$              ;IF SO, GO GET STARTED
2434  014002  012703  010070          2$:     MOV     #RCVON!S5O!EIGHT!TWOSTOP,R3 ;LOAD PARAMETERS TEMPORARILY
2435  014006  005237  001376                  INC     XMTLIN          ;POINT TO THE NEXT LINE TO TRANSMIT
2436  014012  042703  000007                  BIC     #7,R3           ;MAKE SURE TEMPORARY PARAMETERS POINT TO 0
2437  014016  053703  001376                  BIS     XMTLIN,R3       ;ADD DESIRED LINE NUMBER
2438  014022  005037  015314                  CLR     OFFSET
2439  014026  104420                          SHIFT                   ;POINT TO THE NEXT LINE
2440  014030  000761                          BR      1$              ;PROCESS THE NEXT LINE
2441  014032                          3$:
2442  014032  104417                          DCLASM                  ;CLEAR DEVICE AND SET MAINT BIT IF I MODE
2443  014034  042703  010000                  BIC     #RCVON,R3       ;ZERO PARAMTERS FOR TX LINE
2444  014040  010377  165754                  MOV     R3,@DZVLPR      ;LOAD PARAMTERS FOR TX
2445  014044  005737  001372                  TST     MODE            ;STAGGERED?
2446  014050  100007                          BPL     10O$            ;BR IF NO
2447  014052  000241                          CLC                     ;SET UP LINE
2448  014054  006003                          ROR     R3              ;
2449  014056  103002                          BCC     98$             ;BR IF LINE WAS EVEN
2450  014060  000241                          CLC                     ;PREPARE TO MKE LINE EVEN
2451  014062  000401                          BR      99$             ;CONTINUE
2452  014064  000261                  98$:    SEC                     ;PREPARE TO MAKE LINE ODD
2453  014066  006103                  99$:    ROL     R3              ;SET ALTERED LINE
2454  014070  052703  010000          10O$:   BIS     #RCVON,R3       ;SET RX ON
2455  014074  010377  165720                  MOV     R3,@DZVLPR      ;LOAD RX PARAMETERS
2456  014100  010337  001374                  MOV     R3,SAVLIN       ;SET FOR RECEIV. LINE
2457  014104  042737  177774  001374          BIC     #^C<3>,SAVLIN   ;ISOLATE LINE NO.
2458  014112  042703  000003                  BIC     #3,R3           ;CLEAR OLD LINE #
2459  014116  053703  001376                  BIS     XMTLIN,R3       ;SET LINE UP AGAIN
2460  014122  010337  001402                  MOV     R3,REGIST       ;SAVE PARAMETERS FOR PRINTOUT
2461  014126  104422                          BUFSET                  ;ZERO DATA BUFFER
2462  014130  005037  001342                  CLR     $TMP0           ;USE $TMP0 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
2463  014134  005037  001344                  CLR     $TMP1           ;INITIALIZE THE TIMER
2464  014140  005037  001350                  CLR     $TMP3           ;INITIALIZE THESE BITS ALSO
2465  014144  012737  000020  001400          MOV     #20,XMTCNT      ;SET HOW MANY CHARACTERS TO TRANSMIT
2466  014152  012777  014742  165664          MOV     #XMTSRV,@DZVTIV
2467  014160  012777  015112  165652          MOV     #RXISR1,@DZVRIV
2468  014166  012777  000200  165646          MOV     #MASK,@DZVRIS
2469  014174  012777  000200  165644          MOV     #MASK,@DZVTIS
2470  014202  110277  165616                  MOVB    R2,@DZVTCR      ;START THE VALID LINE
2471  014206  052777  040140  165574          BIS     #TIE!RIE!MSENAB,@DZVCSR
2472  014214  106427  000000                  MTPS    #0              ;LOWER THE PRIORITY TO ALLOW INTERRUPTS
2473  014220  032777  000100  165562  4$:     BIT     #RIE,@DZVCSR    ;IS ROUTINE DONE?
2474  014226  001407                          BEQ     5$              ;WHEN ALL IS DONE RX IE IS CLEARED IN ISR.
2475  014230  005237  001344                  INC     $TMP1           ;INCREMENT TIMER
2476  014234  001371                          BNE     4$              ;WHEN IT OVERFLOWS
2477  014236  005237  001350                  INC     $TMP3           ;CATCH CARRY
2478  014242  001366                          BNE     4$              ;CONTINUE TEST
2479  014244  104011                          ERROR   11              ;INTERRUPTS NOT FINISHED
2480  014246  004737  006772          5$:     JSR     PC,SERV.G       ;<^G>?
2481  014252  104401                          SCOP1                   ;LOOP?
2482  014254  062737  000002  015314          ADD     #2,OFFSET
2483  014262  022703  017400                  CMP     #17400,R3
2484  014266  003006                          BGT     6$
2485  014270  032703  000030                  BIT     #BIT4+BIT3,R3   ;IS  CHARACTER SIZE DONE?
2486  014274  001642                          BEQ     2$
2487  014276  162703  000010                  SUB     #BIT3,R3
```

```
                                        BR      3$
                                6$:     ADD     #400,R3
                                        BR      3$
                                ;************************ TEST 5 ********************************
                                ;*THE MAIN FUNCTION OF THIS TEST IS TO VERIFY
                                ;*THAT "PE" (PARITY ERROR) CAN BE FLAGGED BY
                                ;*THE UARTS. THIS TEST WILL NOT BE DONE UNLESS
                                ;*YOU ARE IN "STAGGERED" MODE.
                                ;*40(8) CHARS ARE USED FOR THIS TEST.
                                ;*ALL SELECTED LINES WILL BE ENABLED AT THE SAME TIME.
                                ;*THIS TEST FIRST CHECKS EVEN PARITY FOR ODD LINES AND
                                ;*ODD PARITY FOR EVEN LINES, THEN IT CHECKS THE REVERSE.
                                ;:*  TEST 5
                                ;;**************************************************************
014312  000004                TST5:   SCOPE
014314  012737  000005  001246        MOV     #5,$TSTNM       ;LOAD THE NUMBER OF THIS TEST
014322  012737  004064  001362        MOV     #$EOP,NEXT      ;POINT TO THE END-OF-PASS HANDLER
014330  005737  001372                TST     MODE            ;IS THIS STAGGERED MODE?
014334  100131                        BPL     17$             ;IF NOT, DON'T DO THIS TEST
014336  105037  001425                CLRB    DONFLG          ;SET UP FOR FIRST TEST PASS
014342  104413                1$:     DEVICE.CLR              ;SET DCLR IN CSR
014344  013701  001370                MOV     PAR,R1          ;USE R1 TO BUILD PARAMETERS TO BE LOADED
014350  042701  000200                BIC     #ODDPAR,R1      ;MAKE SURE ODD PARITY ISN'T SET
014354  052701  000100                BIS     #PARITY,R1      ;MAKE SURE PARITY IS TURNED ON
014360  012702  000001                MOV     #1,R2           ;USE R2 AS A LINE POINTER
014364  030237  001366          1$:   BIT     R2,LINE         ;IS THIS A VALID LINE?
014370  001420                        BEQ     3$              ;IF NOT, SKIP TO THE NEXT LINE
014372  105737  001425                TSTB    DONFLG          ;FIRST PASS THROUGH TEST?
014376  001004                        BNE     15$             ;IF NO BRANCH
014400  032701  000001                BIT     #BIT0,R1        ;IS THIS LINE AN ODD LINE?
014404  001006                        BNE     2$              ;IF IT'S ODD, USE EVEN PARITY
014406  000403                        BR      16$             ;IF EVEN SET FOR ODD PARITY
014410  032701  000001          15$:  BIT     #BIT0,R1        ;IF THE LINE  IS EVEN SET FOR EVEN PAR.
014414  001402                        BEQ     2$              ;GO LOAD PARAMETER
014416  052701  000200          16$:  BIS     #ODDPAR,R1      ;IF IT'S ODD, USE ODD PARITY
014422  010177  165372          2$:   MOV     R1,@DZVLPR      ;LOAD THE LINE PARAMETER REGISTER
014426  042701  000200                BIC     #ODDPAR,R1      ;SET UP THE NEXT PARITY TO EVEN
014432  005201                3$:     INC     R1              ;POINT TO THE NEXT LINE
014434  006302                        ASL     R2              ;
014436  032702  000020                BIT     #BIT4,R2        ;ALL LINES DONE?
014442  001750                        BEQ     1$              ;IF NOT, GO CHECK THE NEXT LINE
014444  005037  001374                CLR     SAVLIN          ;CLEAR THE LINE NUMBER INDICATOR
014450  005037  001342                CLR     $TMP0           ;USE $TMP0 TO COUNT TOTAL NUMBER OF TRANSMISSIONS
014454  005003                        CLR     R3              ;USE R3 TO COUNT TOTAL NUMBER OF RECEPTIONS
014456  012737  000040  001400        MOV     #40,XMTCNT      ;TRANSMIT A BINARY COUNT PATTERN(00-40)
014464  104422                        BUFSET                  ;ZERO BUFFER AREA
014466  012777  014742  165350        MOV     #XMTSRV,@DZVTIV ;SET UP THE TRANSMITTER INTERRUPT VECTOR
014474  012777  014622  165336        MOV     #9$,@DZVRIV     ;SET UP THE RECEIVER INTERRUPT VECTOR
014502  012777  000200  165332        MOV     #MASK,@DZVRIS   ;SET THE INTERRUPT VECTOR STATUS
014510  012777  000200  165330        MOV     #MASK,@DZVTIS   ;SET TRANSMITTER INTERRUPT PRIORITY
014516  052777  040140  165264        BIS     #RIE!TIE!MSENAB,@DZVCSR ;ENABLE THE DEVICE
014524  113777  001366  165272        MOVB    LINE,@DZVTCR    ;ENABLE ALL SELECTED LINES
014532  106427  000000                MTPS    #0              ;ALLOW INTERRUPTS
014536  005037  014614          4$:   CLR     7$
014542  005037  014616                CLR     8$
014546  032777  000100  165234  5$:   BIT     #RIE,@DZVCSR    ;WHEN RX DONE; RIE WILL =0
```

```
                                                BEQ     6$              ;BR IF ALL DONE
                                                INC     7$
                                                BNE     5$
                                                INCB    8$
                                                BPL     5$
                                                ERROR   11              ;*RX FAILED TO FINISH (INTERRUPT)
                                        6$:     MTPS    #MASK           ;SHUT OFF INTERRUPTS
                                                TSTB    DONFLG          ;IS THIS SECOND TEST PASS
                                                BNE     17$             ;IF SO GET OUT
                                                INCB    DONFLG          ;INDICATE FIRST TEST PASS DONE
                                                BR      14$             ;START OVER
                                        7$:     0
                                        8$:     0
                                        17$:    ADVANCE


                                                ;RECEIVER SERVICE ROUTINE

                                        9$:     MOV     @DZVRBUF,R4     ;GET THE CHARACTER
                                                MOV     R4,R1           ;COPY THE RECEIVED INFORMATION
                                                SWAB    R1              ;GET THE LINE NUMBER IN THE LOWER BYTE
                                                BIC     #^C<3>,R1       ;ISOLATE THE LINE NUMBER
                                                MOV     R1,SAVLIN       ;SET LINE INDIC. TO RECEIVING LINE
                                                TST     R4              ;IS DATA VALID SET?
                                                BMI     10$             ;IF YES DON'T PRINT ERROR
                                                ERROR   23              ;DATA VALID NOT SET
                                        10$:    MOV     R1,R5           ;BUILD LINE NO. FOR
                                                SWAB    R5              ;EXPECTED DATA IN RECEIVER BUFFER
                                                ASL     R1              ;ADJUST R1 FOR OFFSET
                                                BISB    TRO(R1),R5      ;LOAD CHARACTER IN EXPECTED
                                                BIS     #DVALID!PARER,R5 ;BUILD WHAT WAS EXPECTED
                                                CMP     R4,R5           ;DOES RECEIVED=EXPECTED
                                                BEQ     12$             ;IF YES DON'T PRINT ERROR
                                                ERROR   6               ;*ERROR- DID NOT GET CORRECT INFORMATION
                                        12$:    INC     TRO(R1)         ;SET UP THE NEXT CHARACTER
                                                INC     R3              ;ADD TO THE TOTAL RECEIVED COUNT
                                                BIT     #TIE,@DZVCSR    ;ARE TRANSMISSIONS DONE?
                                                BNE     13$             ;IF NO, GO RECEIVE SOME MORE
                                                CMP     $TMPO,R3        ;ARE ALL CHARACTERS RECEIVED?
                                                BNE     13$             ;IF NO, GO RECEIVE SOME MORE
                                                BIC     #RIE,@DZVCSR    ;DISABLE RECEIVER INTERRUPTS
                                                MOV     #6$,(SP)        ;CRUNCH THE STACK
                                                RTI                     ;RETURN AND FINISH
                                        13$:    MOV     #4$,(SP)        ;CRUNCH THE STACK
                                                RTI                     ;GO BACK TO RECEIVER WAIT LOOP
```

```
2589
2590                                    ;TRANSMITTER INTERRUPT SERVICE
2591                                    ;----------------------------
2592
2593  014742  117701  165044          XMTSRV: MOVB    @#DZVCSR,R1     ;GET THE LINE NUMBER.
2594  014746  042701  177774                  BIC     #^C<3>,R1       ;CLEAR JUNK
2595  014752  013705  001374                  MOV     SAVLIN,R5       ;SAVE REC. LINE NO.
2596  014756  010137  001374                  MOV     R1,SAVLIN       ;LOAD TRANS LINE NO FOR ERROR PRINTOUT
2597  014762  006301                          ASL     R1              ;ADJUST R1 FOR OFFSET
2598  014764  023761  001400  001426          CMP     XMTCNT,TDO(R1)  ;HAVE ALL CHAR. BEEN SENT
2599  014772  003414                          BLE     6$              ;IF YES GO CLEAR TCR
2600  014774  005777  165010                  TST     @#DZVCSR        ;TRDY SET?
2601  015000  100401                          BMI     2$              ;IF YES GO LOAD CHAR.
2602  015002  104003                          ERROR   3               ;#TRANSMITTER NOT READY- FALSE INTERRUPT
2603  015004  116177  001426  165022  2$:     MOVB    TDO(R1),@#DZVTDR ;LOAD THE CURRENT CHARACTER FOR THIS LINE
2604  015012  005261  001426                  INC     TDO(R1)         ;SET UP NEXT CHARACTER FOR THIS LINE
2605  015016  005237  001342                  INC     $TMP0           ;UP THE NUMBER OF TRANSMISSIONS
2606  015022  000415                          BR      7$              ;GO RETURN
2607  015024  012700  000001          6$:     MOV     #1,R0           ;SET UP A DESELECTION POINTER
2608  015030  006201                          ASR     R1              ;GET LINE NO. AGAIN
2609  015032  005301                  12$:    DEC     R1              ;REDUCE THE COUNT, WAS THIS THE LINE?
2610  015034  100402                          BMI     3$              ;IF SO, GO DISABLE THE ENABLE BIT FOR IT
2611  015036  006300                          ASL     R0              ;MOVE THE POINTER TO THE NEXT LINE
2612  015040  000774                          BR      12$             ;GO CHECK THE NEXT LINE
2613  015042  140077  164756          3$:     BICB    R0,@#DZVTCR     ;DISABLE THE LINE POINTED TO BY R0
2614  015046  001003                          BNE     7$              ;IF MORE LINES ARE ACTIVE , GO CONTINUE TRANSMIT
2615  015050  042777  040000  164732          BIC     #TIE,@#DZVCSR   ;IF NOT, DISABLE TRANSMITTER INTERRUPTS
2616  015056  010537  001374          7$:     MOV     R5,SAVLIN       ;RESTORE RECEIV. LINE
2617  015062  000002                          RTI                     ;RETURN TO THE TIMING LOOP
2618
2619                                    ; RELATIVE TIME BUILDING ROUTINE
2620                                    ; -----------------------------
2621
2622  015064  012737  000002  001346  BUILD:  MOV     #2,$TMP2        ;ROTATE 2 BITS BACK INTO $TMP1
2623  015072  006037  001350          1$:     ROR     $TMP3           ;GET THE BITS FROM $TMP3, THE HIGH BYTE
2624  015076  006037  001344                  ROR     $TMP1           ;OF THE RELATIVE TIME COUNTER. PUT THEM BACK
2625  015102  005337  001346                  DEC     $TMP2           ;INTO $TMP1 USING THE CARRY BIT WITH
2626                                                                  ;ROTATE INSTRUCTIONS
2627  015106  001371                          BNE     1$              ;REDUCE COUNT. ALL BITS BACK? IF NOT, GET MORE
2628  015110  000207                          RTS     PC              ;RETURN TO CALLING TEST
2629
```

```
                                        ;RECEIVER SERVICE ROUTINE

015112  105777  164672    RXISR1: TSTB    @DZVCSR         ;IS THE RECEIVER REALLY READY?
015116  100401                    BMI     1$              ;IF SO, GO SERVICE IT
015120  104004                    ERROR   4               ;#ERROR- RECEIVER DONE FLAG ISN'T SET
015122  017704  164666    1$:     MOV     @DZVRBUF,R4     ;SAVE THE RECEIVER INFORMATION
015126  100401                    BMI     2$              ;IF IT WAS VALID, GO PROCESS IT
015130  104023                    ERROR   23              ;ERROR- DATA VALID WASN'T SET
015132  032704  040000    2$:     BIT     #OVRRUN,R4      ;OVERRUN ERROR FLAG SET?
015136  001401                    BEQ     6$              ;IF NOT DON'T TYPE ERROR
015140  104024                    ERROR   24              ;OVERRUN ERROR
015142  032704  020000    6$:     BIT     #FRMERR,R4      ;FRAMING ERROR FLAG SET?
015146  001401                    BEQ     9$              ;IF NOT DON'T TYPE ERROR
015150  104025                    ERROR   25              ;FRAMING ERROR
015152  032704  010000    9$:     BIT     #PARER,R4       ;PARITY ERROR FLAG SET?
015156  001401                    BEQ     3$              ;IF NOT, GO CONTINUE PROCESSING
015160  104026                    ERROR   26              ;ERROR- RECEIVER ERROR FLAG SET
015162  013701  001374    3$:     MOV     SAVLIN,R1       ;CALCULATE THE DATA OFFSET
015166  006301                    ASL     R1              ;ALIGN IT ON A WORD BOUNDARY
015170  120461  001436            CMPB    R4,TRO(R1)      ;IS THE CHARACTER WHAT IT SHOULD BE?
015174  001407                    BEQ     4$              ;IF SO,GO CONTINUE PROCESSING
015176  116105  001436            MOVB    TRO(R1),R5      ;GET WHAT WAS EXPECTED FOR ERROR REPORTING
015202  042705  177400            BIC     #^C<377>,R5     ;ELIMINATE PROPAGATED SIGN
015206  042704  177400            BIC     #^C<377>,R4     ;ISOLATE THE ACTUAL CHARACTER
015212  104005                    ERROR   5               ;#DATA ERROR
015214  005261  001436    4$:     INC     TRO(R1)         ;SET UP THE NEXT EXPECTED CHARACTER
015220  122761  000020  001436    CMPB    #20,TRO(R1)     ;HAVE ALL CHARACTERS BEEN RECEIVED?
015226  001031                    BNE     8$              ;IF NOT RETURN
015230  126137  001436  001342    CMPB    TRO(R1),$TMPO   ;ALL CHARAC. RECEIVED?
015236  001025                    BNE     8$              ;IF SO,GO DETERMINE THE TIMING
015240  004737  015064            JSR     PC,BUILD        ;GET THE RELATIVE TIME (SIGNIFICANT BITS)
015244  013700  015314            MOV     OFFSET,R0       ;GET POINTER
015250  013760  001344  002050    MOV     $TMP1,^TMTBL(R0)  ; SAVE THIS TEST'S TIME
015256  005737  015314            TST     OFFSET^         ;FIRST TEST?
015262  001410                    BEQ     7$              ;IF NOT, GO CHECK THE TIME
015264  005740                    TST     -(R0)           ;POINT TO THE PREVIOUS  TIME TAKEN
015266  026037  002050  001344    CMP     TMTBL(R0),$TMP1 ;IS THIS TIME WHAT IT SHOULD BE?
015274  101003                    BHI     7$              ;IF SO, GO TO THE NEXT TEST
015276  016005  002050            MOV     TMTBL(R0),R5    ;PLACE WHAT WAS EXPECTED IN R5
015302  104021                    ERROR   21              ;TIMING ERROR
015304  042777  000140  164476  7$:     BIC     #RIE!MSENAB,@DZVCSR ;DISABLE THE DEVICE
015312  000002            8$:     RTI                     ;RETURN TO THE PROGRAM
015314  000000            OFFSET: 0
```

```
                                        ;ERROR TABLE
015316  000000          .ERRTAB:        0       ;ERROR 0
015320  000000                          0
015322  000000                          0

015324  015544                          EM1     ;ERROR
015326  016670                          DH1
015330  017070                          DT1

015332  015617                          EM2     ;ERROR 2
015334  016714                          DH2
015336  017102                          DT2

015340  015645                          EM3     ;ERROR 3
015342  016747                          DH3
015344  017120                          DT3

015346  015704                          EM4     ;ERROR 4
015350  016747                          DH3
015352  017120                          DT3

015354  015733                          EM5     ;ERROR 5
015356  016761                          DH4
015360  017126                          DT4

015362  015762                          EM6     ;ERROR 6
015364  016761                          DH4
015366  017126                          DT4

015370  000000                          0
015372  000000                          0
015374  000000                          0

015376  000000                          0
015400  000000                          0
015402  000000                          0

015404  016021                          EM11    ;ERROR 11
015406  016747                          DH3
015410  017120                          DT3

015412  000000                          0
015414  000000                          0
015416  000000                          0

015420  016057                          EM13    ;ERROR 13
015422  016747                          DH3
015424  017120                          DT3

015426  016110                          EM14    ;ERROR 14
015430  016747                          DH3
015432  017120                          DT3

015434  016142                          EM15    ;ERROR 15
015436  000000                          0
015440  000000                          0
```

```
2729
2730  015442  016204                  EM16
2731  015444  016747                  DH3
2732  015446  017120                  DT3
2733
2734  015450  016256                  EM17    ;ERROR 17
2735  015452  016747                  DH3
2736  015454  017120                  DT3
2737
2738  015456  016314                  EM20
2739  015460  016747                  DH3
2740  015462  017120                  DT3
2741
2742  015464  016355                  EM21    ;ERROR 21
2743  015466  017010                  DH5
2744  015470  017144                  DT5
2745
2746  015472  000000                  0
2747  015474  000000                  0
2748  015476  000000                  0
2749
2750  015500  016405                  EM23    ;ERROR 23
2751  015502  016747                  DH3
2752  015504  017120                  DT3
2753
2754  015506  016435                  EM24
2755  015510  016747                  DH3
2756  015512  017120                  DT3
2757
2758  015514  016463                  EM25
2759  015516  016747                  DH3
2760  015520  017120                  DT3
2761
2762  015522  016513                  EM26
2763  015524  016747                  DH3
2764  015526  017120                  DT3
2765
2766  015530  016542                  EM27
2767  015532  016747                  DH3
2768  015534  017120                  DT3
2769
2770  015536  016610                  EM30
2771  015540  016747                  DH3
2772  015542  017120                  DT3
```

```
                                        ;ERROR MESSAGES
015544 047200 020117 052502 EM1:    .ASCIZ  <200>/NO BUS REPLY RESPONSE FROM DZV11 REGISTER/
015617    200 042522 044507 EM2:    .ASCIZ  <200>?REGISTER R/W FAILURE?
015645    200 051124 047101 EM3:    .ASCIZ  <200>/TRANSMIT READY (TRDY) NOT SET/
015704 051200 041505 044505 EM4:    .ASCIZ  <200>/RECEIVER DONE NOT SET/
015733    200 040504 040524 EM5:    .ASCIZ  <200>/DATA COMPARISON ERROR/
015762 042200 053132 030461 EM6:    .ASCIZ  <200>/DZV11 *RECEIVER BUFFER* ERROR/
016021    200 042522 042503 EM11:   .ASCIZ  <200>/RECEIVER FAILED TO INTERRUPT/
016057    200 044523 047514 EM13:   .ASCIZ  <200>/SILO ALARM SET TOO SOON/
016110 051600 046111 020117 EM14:   .ASCIZ  <200>/SILO ALARM FAILED TO SET/
016142 040600 052103 047511 EM15:   .ASCIZ  <200>/ACTION DETECTED ON INVALID LINE./
016204 051200 040505 044504 EM16:   .ASCIZ  <200>/READING DZVRBUF DID NOT CLEAR SILO ALARM/
016256 042200 052101 020101 EM17:   .ASCIZ  <200>/DATA VALID SHOULD NOT BE SET/
016314 051200 041505 044505 EM20:   .ASCIZ  <200>/RECEIVER DONE SHOULD NOT BE SET/
016355    200 042522 040514 EM21:   .ASCIZ  <200>/RELATIVE TIMING ERROR./
016405    200 040504 040524 EM23:   .ASCIZ  <200>/DATA VALID IS NOT SET!/
016435    200 040504 040524 EM24:   .ASCIZ  <200>/DATA OVERRUN IS SET!/
016463    200 051106 046501 EM25:   .ASCIZ  <200>/FRAMING ERROR OCCURRED/
016513    200 040520 044522 EM26:   .ASCIZ  <200>/PARITY ERROR OCCURRED/
016542 051600 046111 020117 EM27:   .ASCIZ  <200>/SILO ALARM FAILED TO CAUSE INTERRUPT/
016610 046200 047111 020105 EM30:   .ASCIZ  <200>/LINE DID NOT RECEIVE FULL BINARY COUNT PATTERN/

016670 052200 040522 020120 DH1:    .ASCIZ  <200>/TRAP PC  DZV11 REG/
016714 042600 050130 041505 DH2:    .ASCIZ  <200>/EXPECTED  FOUND  REGISTER/
016747    200 044514 042516 DH3:    .ASCIZ  <200>/LINE NO./
016761    200 054105 042520 DH4:    .ASCIZ  <200>/EXPECTED  FOUND  LINE/
017010 052200 020130 044514 DH5:    .ASCIZ  <200>/TX LINE PREVIOUS TIME  ACTUAL TIME  PARAMETER/

       017070                       .EVEN
                                        ;DATA TABLES FOR ERROR MESSAGES
2775 017070 000002           DT1:   2
2776 017072     006 003              .BYTE   6,3
2777 017074 001330                   $REG1
2778 017076     006 001              .BYTE   6,1
2779 017100 001326                   $REG0
2780
2781 017102 000003           DT2:   3
2782 017104     006 004              .BYTE   6,4
2783 017106 001340                   $REG5
2784 017110     006 001              .BYTE   6,1
2785 017112 001336                   $REG4
2786 017114     006 001              .BYTE   6,1
2787 017116 001326                   $REG0
2788
2789 017120 000001           DT3:   1
2790 017122     003 001              .BYTE   3,1
2791 017124 001374                   SAVLIN
2792
2793 017126 000003           DT4:   3
2794 017130     006 004              .BYTE   6,4
2795 017132 001340                   $REG5
2796 017134     006 001              .BYTE   6,1
2797 017136 001336                   $REG4
2798 017140     003 001              .BYTE   3,1
2799 017142 001374                   SAVLIN
2800
```

```
2801  017144  000004                  DT5:    4
2802  017146      003      005                .BYTE   3,5
2803  017150  001374                          SAVLIN
2804  017152      006      011                .BYTE   6,9.
2805  017154  001340                          $REGS
2806  017156      006      007                .BYTE   6,7
2807  017160  001344                          $TMP1
2808  017162      006      001                .BYTE   6,1
2809  017164  001402                          REGIST
2810                                          ;TABLE OF DELAY TIMES FOR INDIVIDUAL BAUD RATES
2811                                          ;---------------------------------------------
2812
2813  017166  002450                  DLYTBL: 2450                    ;TIME FOR   50 BAUD
2814  017170  001560                          1560                    ;TIME FOR   75 BAUD
2815  017172  001120                          1120                    ;TIME FOR  110 BAUD
2816  017174  000750                           750                    ;TIME FOR  134 BAUD
2817  017176  000660                           660                    ;TIME FOR  150 BAUD
2818  017200  000330                           330                    ;TIME FOR  300 BAUD
2819  017202  000150                           150                    ;TIME FOR  600 BAUD
2820  017204  000060                            60                    ;TIME FOR 1200 BAUD
2821  017206  000040                            40                    ;TIME FOR 1800 BAUD
2822  017210  000030                            30                    ;TIME FOR 2000 BAUD
2823  017212  000020                            20                    ;TIME FOR 2400 BAUD
2824  017214  000010                            10                    ;TIME FOR 3600 BAUD
2825  017216  000001                             1                    ;TIME FOR 4800 BAUD
2826  017220  000001                             1                    ;TIME FOR 7200 BAUD
2827  017222  000001                             1                    ;TIME FOR 9600 BAUD
2828  017224  000001                             1                    ;TIME OF DELAY FOR 19200 BAUD
2829
2830                                          ;DELAYS WERE COMPUTED TO ALLOW MAXIMUM TIME AT EACH BAUD RATE
2831                                          ;FOR ALL TESTS TO FUNCTION CORRECTLY ON A LSI11.
2832
2833  017226                  CORMAX:
2834          000001          .END
```

```
ABASE = 160010          1#    342     383
ACOM1 = 000017          1#    342     385
ACOM2 = 000000        342     386
ACPUOP= 000000        342     357
ACTIVE  001420        500#    756*   1789*   1790    1792*   1796    1797*   1798    1801*
ADDM0 = 017470          1#    342     387
ADDM1 = 017470          1#    342     388
ADDM10= 017470          1#    342     397
ADDM11= 017470          1#    342     398
ADDM12= 017470          1#    342     399
ADDM13= 017470          1#    342     400
ADDM14= 017470          1#    342     401
ADDM15= 017470          1#    342     402
ADDM2 = 017470          1#    342     389
ADDM3 = 017470          1#    342     390
ADDM4 = 017470          1#    342     391
ADDM5 = 017470          1#    342     392
ADDM6 = 017470          1#    342     393
ADDM7 = 017470          1#    342     394
ADDM8 = 017470          1#    342     395
ADDM9 = 017470          1#    342     396
ADEVCT= 000000        342     348
ADEVM = 000001          1#    342     384
ADRCNT  005661       1323*   1360*   1370#
ADVANC= 104400        652#    1500    2297    2557
AENV  = 000000        342     353
AENVM = 000000        342     354
AFATAL= 000000        342     345
AMADR1= 000000        342     370
AMADR2= 000000        342     374
AMADR3= 000000        342     377
AMADR4= 000000        342     380
AMAMS1= 000000        342     364
AMAMS2= 000000        342     372
AMAMS3= 000000        342     375
AMAMS4= 000000        342     378
AMSGAD= 000000        342     350
AMSGLG= 000000        342     351
AMSGTY= 000000        342     344
AMTYP1= 000000        342     365
AMTYP2= 000000        342     373
AMTYP3= 000000        342     376
AMTYP4= 000000        342     379
APASS = 000000        342     347
APRIOR= 000000        342
APTCSU= 000040       1174    1279#
APTENV= 000001       1167    1235    1277#   1587
APTSIZ= 000200       1276#
APTSPO= 000100       1169    1237    1278#
ASWREG= 000000        342     355
ATESTN= 000000        342     346
AUNIT = 000000        342     349
AUSWR = 000000        342     356
AUTO.S  011464        931    1971#
AVECT1= 000300          1#    342     381
AVECT2= 000000        342     382
```

```
BINWRD  006134          1446#
BIT0   = 000001          139#  211   213   250   261  2517  2520
BIT00  = 000001          129#  139
BIT01  = 000002          128#  138
BIT02  = 000004          127#  137
BIT03  = 000010          126#  136
BIT04  = 000020          125#  135
BIT05  = 000040          124#  134
BIT06  = 000100          123#  133
BIT07  = 000200          122#  132
BIT08  = 000400          121#  131
BIT09  = 001000          120#  130
BIT1   = 000002          138#  212   213   251   262
BIT10  = 002000          119#  235   236   237   238   243   244   245   246   256   267   275
BIT11  = 004000          118#  239   240   241   242   243   244   245   246   257   268   276  1107
BIT12  = 010000          117#  178   194   229
BIT13  = 020000          116#  179   195
BIT14  = 040000          115#  180   196  1089  2050
BIT15  = 100000          114#  181   197
BIT2   = 000004          137#  252   263   992
BIT3   = 000010          136#  173   216   218   220   222   253   264  2485  2487
BIT4   = 000020          135#  174   217   218   221   222  1497  1512  2485  2527
BIT5   = 000040          134#  175   219   220   221   222   227  1983  1993  2050
BIT6   = 000100          133#  176   224
BIT7   = 000200          132#  165   177   225   929  1616  1630
BIT8   = 000400          131#  186   188   203   205   232   234   236   238   240   242   244   246
                          254  265   273
BIT9   = 001000          130#  187   188   204   205   233   234   237   238   241   242   245   246
                          255  266   274
BPTVEC= 000014           146#
BRK0   = 000400          273#
BRK1   = 001000          274#
BRK2   = 002000          275#
BRK3   = 004000          276#
BRW     004540           997  1127#
BUFSET= 104422           688#  2092  2208  2278  2461  2533
BUILD   015064          2622#  2660
CHRCNT  006132          1412*  1425*  1443#
CLEAR  = 000000           166#  2285*
CNVRT  = 104412           672#  1020  1022  1025  1028  1566  1568  1570  1623
CONVRT= 104411            670#   941  1584
CORMAX  017226          2833#  2834
COO    = 000400           265#
CO1    = 001000           266#
CO2    = 002000           267#
CO3    = 004000           268#
CR     = 000015            54#  1213  1223
CRLF   = 000200            55#  1184  1223
CSRMAP  011472          1974#
CYCLE   010436           998  1051  1779#
DATABP  006626          1555*  1558  1582  1585#
DATAHD  006614          1554*  1578  1581#
DCLASM= 104417           682#  2089  2205  2274  2442
DCLR   = 000020          174#  1466  1467
DDISP  = 177570           61#   434
DELAY  = 104414          676#  2107  2123  2133  2225  2245  2289
```

```
DEVADR  005656          1321#  1357  1368#
DEVICE= 104413           674#  1473  2380  2508
DH1     016670          2679  2774#
DH2     016714          2683  2774#
DH3     016747          2687  2691  2711  2719  2723  2731  2735  2739  2751  2755  2759  2763  2767
                        2771  2774#
DH4     016761          2695  2699  2774#
DH5     017010          2743  2774#
DISPLA  001306           434#   765*  1119*  1597*
DISPRE  000174           329#   765
DLYCNT  006230           916*   974  1479  1484#
DLYTBL  017166           916  2813#
DONFLG  001425           508#  2507*  2515  2551  2553*
DSWR  = 177570            60#   433
DTR0  = 000400           254#
DTR1  = 001000           255#
DTR2  = 002000           256#
DTR3  = 004000           257#
DT1     017070          2680  2775#
DT2     017102          2684  2781#
DT3     017120          2688  2692  2712  2720  2724  2732  2736  2740  2752  2756  2760  2764  2768
                        2772  2789#
DT4     017126          2696  2700  2793#
DT5     017144          2744  2801#
DVALID= 100000           197#  2150  2367  2574
DZCR0   001500           548#   796   799   844  2024
DZCR1   001512           554#   845
DZCR10  001620           596#
DZCR11  001632           602#
DZCR12  001644           608#
DZCR13  001656           614#
DZCR14  001670           620#
DZCR15  001702           626#
DZCR16  001714           632#
DZCR17  001726           638#
DZCR2   001524           560#
DZCR3   001536           566#
DZCR4   001550           572#
DZCR5   001562           578#
DZCR6   001574           584#
DZCR7   001606           590#
DZVACT  001406           493#   944*   954   957   962  1779  1785
DZVCSR  002010           696#  1055  1466*  1467  1474*  1858*  2094*  2100  2105  2115  2121  2155  2179*
                        2183*  2187*  2212*  2223  2233  2239  2243  2283*  2305  2330  2333  2471*  2473
                        2538*  2543  2580  2584*  2600  2615*  2632  2670*
DZVC0   001502           549#   808   811  2066
DZVC1   001514           555#
DZVC10  001622           597#
DZVC11  001634           603#
DZVC12  001646           609#
DZVC13  001660           615#
DZVC14  001672           621#
DZVC15  001704           627#
DZVC16  001716           633#
DZVC17  001730           639#
DZVC2   001526           561#
```

```
DZVC3   001540            567#
DZVC4   001552            573#
DZVC5   001564            579#
DZVC6   001576            585#
DZVC7   001610            591#
DZVLEV  011002           1811   1847#
DZVLPR  002020            700#  1509*  1863*  2444*  2455*  2523*
DZVMSR  002030            704#  1872*
DZVNUM  001414            496#   837*   945  1941*  1951*  1979*  2008*  2009  2015  2017
DZVRBU  002014            698#  1862*  2151  2165  2216  2336  2562  2635
DZVRIS  002042            712#  1849*  2211*  2280*  2468*  2536*
DZVRIV  002040            711#  1058  1803*  1847  2210*  2238*  2279*  2467*  2535*
DZVTCR  002024            702#  1868*  2099*  2169*  2182*  2221*  2259*  2284*  2318*  2470*  2539*  2613*
DZVTDR  002034            706#  1873*  2111*  2190*  2229*  2241*  2321*  2603*
DZVTIS  002046            714#  1853*  2282*  2469*  2537*
DZVTIV  002044            713#  1851*  2180*  2281*  2466*  2534*
DZV.EN  001740            644#   782  1790  1798  1977
DZV.MA  001500            500    546#   756   779   936  1792  1801  1935  1974  1980  2038
EIGHT =  000030            218#  2431  2434
EIGHTS=  000070            222#
EMTVEC=  000030            149#
EM1     015544           2678   2774#
EM11    016021           2710   2774#
EM13    016057           2718   2774#
EM14    016110           2722   2774#
EM15    016142           2726   2774#
EM16    016204           2730   2774#
EM17    016256           2734   2774#
EM2     015617           2682   2774#
EM20    016314           2738   2774#
EM21    016355           2742   2774#
EM23    016405           2750   2774#
EM24    016435           2754   2774#
EM25    016463           2758   2774#
EM26    016513           2762   2774#
EM27    016542           2766   2774#
EM3     015645           2686   2774#
EM30    016610           2770   2774#
EM4     015704           2690   2774#
EM5     015733           2694   2774#
EM6     015762           2698   2774#
ERRMSG  006602           1553*  1573  1576#
ERRVEC=  000004            142#  1094  1095*  1097*  1100*
ERTABO  006756           1568   1609#
EVEPAR=  000000            228#
EXITER  006706           1596   1599#
FIVE  =  000000            215#
FIVES =  000040            219#
FRMERR=  020000            195#  2347  2641
HALTS   006632           1538   1587#
HDRFLG  001423            506#   780*   879*   932   934*
HDZVCS  002012            697#  1860*  2301  2593
HDZVLP  002022            701#  1866*
HDZVMS  002032            705#  1875*
HDZVRB  002016            699#  1865*
HDZVTC  002026            703#  1870*
```

```
MDZVTD  002036            707#  1876*
MILIM   005654           1320*  1348   1367#
MT    = 000011             52#  1182   1223
INBUF   010270           1290   1326   1719*  1720   1725   1730   1748*  1749   1753   1763#  1887
INIFLG  001422            505#   766    771*  784*   865*   927   1973*
INSTER= 104404            660#  1342   1736   1758   1906
INSTR = 104403            658#   791    803    814    821    871    904    949    969   1817
INSTR2  005454           1297   1309#
IOTVEC= 000020            147#  2067*
LF    = 000012             53#  1217   1223
LIMITS  005602           1336   1348#
LINE    001366            485#  1804*  2095   2214   2275   2276   2284   2313   2358   2384   2432   2513   2539
LINE0   001504            550#   830*   876    886    912
LINE1   001516            556#
LINE10  001624            598#
LINE11  001636            604#
LINE12  001650            610#
LINE13  001662            616#
LINE14  001674            622#
LINE15  001706            628#
LINE16  001720            634#
LINE17  001732            640#
LINE2   001530            562#
LINE3   001542            568#
LINE4   001554            574#
LINE5   001566            580#
LINE6   001600            586#
LINE7   001612            592#
LOBITS  005660           1322*  1352   1369#
LOCK    001364            480#  1135   1137   1490*  1562   2088*  2204*  2420*
LOLIM   005652           1319*  1350   1366#
LPRSET= 104421            686#  2090   2206   2277
LP0   = 000000            210#
LP1   = 000001            211#
LP2   = 000002            212#
LP3   = 000003            213#
MAINT = 000010            173#  1125   1728   1810
MANT0   001510            552#   817    884   2025
MANT1   001522            558#
MANT10  001630            600#
MANT11  001642            606#
MANT12  001654            612#
MANT13  001666            618#
MANT14  001700            624#
MANT15  001712            630#
MANT16  001724            636#
MANT17  001736            642#
MANT2   001534            564#
MANT3   001546            570#
MANT4   001560            576#
MANT5   001572            582#
MANT6   001604            588#
MANT7   001616            594#
MASK  = 000200            165#   310    752*   989*  2036   2211   2280   2282   2379*  2468   2469   2536   2537
                         2550*
MASTEK  007705           1564   1707#
```

```
MBADLN  010020            893  1707#
MCSRX   007635           1019  1569  1707#
MDATA   010374           1423  1433  1767#
MEPASS  007453           1018  1707#
MERRPC  007765           1567  1707#
MERRX   007643           1027  1707#
MERR2   007513           1707# 1781  2029
MERR3   007562            959  1707#
MLOCK   007606            994  1707#
MMEM    007710            950  1707#
MNTFLG  001424            507# 1122* 1125* 1474  1723* 1728* 1733* 1807* 1810*
MODE    001372            487# 1123  1806* 1808  2138  2445  2505
MPASSX  007651           1024  1707#
MPFAIL  007410           1700  1707#
MR      007477           1000  1707#
MSENAB= 000040            175# 2094  2179  2187  2212  2283  2471  2538  2670
MTITLE  001000            336#  770
MTSTN   007673           1565  1707#  1818
MVECX   007643           1021  1707#
NEXT    001362            479# 1489  1605  2087* 2203* 2273* 2419* 2425  2504*
NOLIST= ###### U            1
ODDPAR= 000200            225# 2510  2522  2524
OFFSET  015314           2427* 2438* 2482* 2661  2663  2672#
ONESTO= 000000            226#
OUT     013620           2292  2379#
OVRRUN= 040000           1968  2162  2344  2638
PAR     001370            486# 1507  1805* 2509
PARAM = 104405            662#  793   805   823   873   906   951   971  1819
PARAM1  005522           1325# 1343
PARER = 010000            194# 2350  2574  2644
PARERR  005576           1328  1330  1332  1342# 1349  1351  1353
PARITY= 000100            224# 2511
PARMD = 104415            678#
PAR0    001506            551#  831*  909   913
PAR1    001520            557#
PAR10   001626            599#
PAR11   001640            605#
PAR12   001652            611#
PAR13   001664            617#
PAR14   001676            623#
PAR15   001710            629#
PAR16   001722            635#
PAR17   001734            641#  923
PAR2    001532            563#
PAR3    001544            569#
PAR4    001556            575#
PAR5    001570            581#
PAR6    001602            587#
PAR7    001614            593#
PAWCH = 104416            680#
PIRQ  = 177772             59#
PIROVE= 000240            153#
POPRO = 012600            162#
POP1SP= 005726            160#
POP2SP= 022626            164#  1499  2063  2255  2258
PRO   = 000000             76#
```

```
PR1     = 000040            77#
PR2     = 000100            78#
PR3     = 000140            79#
PR4     = 000200            80#
PR5     = 000240            81#
PR6     = 000300            82#
PR7     = 000340            83#
PS      = 177776            56#    57
PSW     = 177776            57#
PUSHAD= 010046             161#
PUSHIS= 005746             159#
PUSH2S= 024646             163#
PWRVEC= 000024             148#   1669*  1670*  1679*  1685*  1697*  1698*
RCVON   = 010000           229#   2431   2434   2443   2454
ROONE   = 000200           177#
REGIST  001402             491#   2460*  2809
RESREG  006630            1583    1586#
RESTAR  010776            1702    1843#
RESVEC= 000010             143#
RESOS   = 104410           668#   1586
RIE     = 000100           176#   2212   2283   2471   2473   2538   2543   2584   2670
RING0   = 000001           261#
RING1   = 000002           262#
RING2   = 000004           263#
RING3   = 000010           264#
RL0     = 000000           202#
RL1     = 000400           203#
RL2     = 001000           204#
RL3     = 001400           205#
RUN     001412             495#    757*  1785   1787*  1788*  1794*  1795*
RXISR1  015112            2467    2632#
RXSVC   013414            2279    2330#
RXTCR   013672            2275*   2290   2373*  2394#
SAVACT  001410             494#    836*   841*   843    944    957   1942*  1950*  2018*  2022*  2027   2046
SAVLIN  001374             488#   2091*  2097   2137   2171*  2207*  2219   2261*  2304*  2340*  2361   2365   2382*
                          2390*   2428*  2456*  2457*  2529*  2566*  2595   2596*  2616*  2647   2791   2799   2803
SAVNO   001416             498#    945*   961*   965*   999   1032
SAVNUM  001415             497#    999*  1030*  1032*
SAVPC   001404             492#   1375*  1611
SAVOS   = 104407           666#   1543
SCOPI   = 104401           654#   2170   2188   2260   2481
SERV.G  006772            1121    1531   1600   1615#  1620   2480
SETAPT  011336             774    1935#
SETFLG= 104406             664#    816
SEVEN   = 000020           217#
SEVENS= 000060             221#
SHIFT   = 104420           684#   2172   2262   2392   2439
SILOAL= 020000             179#   2115   2121   2155   2233   2243   2333
SILOEN= 010000             178#   2094   2179   2187   2212
SIX     = 000010           216#
SIXS    = 000050           220#
SNAP    013242            2288#   2374
SPACNT  006133            1444#
STACK   = 001120            47#    751    988   1606
STKLMT= 177774              58#
STOP    001446             519#   1523
```

```
SVOS    005670          1379#
SWR     001304           433#   764*   776    833    861    929    946    967    992   1089   1107   1133   1140
                         1532   1537   1595   1601   1603   1621   1626*  1638*  1639*  1640*  1647*  1651*  1677
                         1690*  1814   1959*
SWREG   000176           330#   764
SW0   = 000001           111#
SW00  = 000001           101#   111    776
SW01  = 000002           100#   110   1814
SW02  = 000004            99#   109
SW03  = 000010            98#   108    833    861
SW04  = 000020            97#   107    967
SW05  = 000040            96#   106
SW06  = 000100            95#   105    946
SW07  = 000200            94#   104
SW08  = 000400            93#   103   1601
SW09  = 001000            92#   102   1133
SW1   = 000002           110#
SW10  = 002000            91#  1603
SW11  = 004000            90#
SW12  = 010000            89#  1140   1532
SW13  = 020000            88#  1537
SW14  = 040000            87#
SW15  = 100000            86#
SW2   = 000004           109#
SW3   = 000010           108#
SW4   = 000020           107#
SW5   = 000040           106#
SW6   = 000100           105#
SW7   = 000200           104#
SW8   = 000400           103#
SW9   = 001000           102#
S110  = 001000           233#
S1200 = 003400           238#
S134  = 001400           234#
S150  = 002000           235#
S1800 = 004000           239#
S19200= 007400           246#
S2000 = 004400           240#
S2400 = 005000           241#
S300  = 002400           236#
S3600 = 005400           242#
S4800 = 006000           243#
S50   = 000000           231#  2431   2434
S600  = 003000           237#
S7200 = 006400           244#
S75   = 000400           232#
S9600 = 007000           245#
TBITVE= 000014           144#
TCR0  = 000001           250#
TCR1  = 000002           251#
TCR2  = 000004           252#
TCR3  = 000010           253#
TD0     001426           511#  1521   2111   2112*  2229   2230*  2231   2241   2321   2322*  2598   2603   2604#
TD1     001430           512#
TD2     001432           513#
TD3     001434           514#
```

```
TEIGHT  002106            737#
TEMP    010332           1765#
TFIVE   002114            740#
TIE   = 040000            180#  2183  2187  2283  2471  2538  2580  2615
TKVEC = 000060            151#
TLAST = 014312           1837   2810#
TLO   = 000000            185#
TL1   = 000400            186#
TL2   = 001000            187#
TL3   = 001400            188#
TMTBL   002050            721#  2662*  2666  2668
TPVEC = 000064            152#
TRAPVE= 000034            150#
TROY  = 100000            181#
TRTVEC= 000014            145#
TRO     001436            515#  2363  2368  2370*  2371  2386  2573  2578*  2649  2651  2655*  2656  2658
TR1     001440            516#
TR2     001442            517#
TR3     001444            518#
TSEVEN  002110            738#
TSIX    002112            739#
TST1    012170           1083   1825  1841  2085#
TST2    012632           2087   2201#
TST3    013132           2203   2271#
TST4    013674           2273   2417#
TST5    014312           2419   2502#  2810
TST6  = ****** U         2504
TTST    004312            995*   997*  1085#
TWOSTO= 000040            227#  2431  2434
TXSVC   013300           2281   2301#
TXTCR   013412           2276*  2316  2324*  2327#
TYPDAT  006616           1559   1579  1582#
TYPE  = 104402            656#   770   892   893   935   959   994  1000  1018  1019  1021  1024  1027
                         1187  1288  1307  1400  1433  1560  1561  1564  1565  1567  1569  1571  1575
                         1580  1622  1624  1652  1699  1781  1816  1834  1839  2029
TYPMSG  006506           1557  1560#
T110    002054            724#
T1200   002066            729#
T134    002056            725#
T150    002060            726#
T1800   002070            730#
T2000   002072            731#
T2400   002074            732#
T300    002062            727#
T3600   002076            733#
T4800   002100            734#
T50     002050            722#
T600    002064            728#
T7200   002102            735#
T75     002052            723#
T9600   002104            736#
VECMAP  012000           2028   2036#
WRDCNT  006130           1408*  1434*  1442#
WTBS.F  006604           1574   1577#
XBX     006374           1533   1535   1537#
XCSR    004250           1020   1053#  1570
```

```
XERR    004272            1028   1062#
XHEAD   007772             935   1707#
XMTCNT  001400             490#  2465*  2532*  2598
XMTLIN  001376             489#  2429*  2435*  2437   2459
XMTSRV  014742            2466   2534   2593#
XPASS   004264            1025   1059#
XSTATQ  010062             942   1707#
XTSTN   006764            1566   1612#
XVEC    004256            1022   1056#
XX    = 160210             547#   553#   559#   565#   571#   577#   583#   589#   595#   601#   607#   613#   619#
                           625#   631#   637#   643#
YY    = 000500             547#   553#   559#   565#   571#   577#   583#   589#   595#   601#   607#   613#   619#
                           625#   631#   637#   643#
ZZ    = 000020             547#   553#   559#   565#   571#   577#   583#   589#   595#   601#   607#   613#   619#
                           625#   631#   637#   643#
$APTHD  001446             529    535#
$ASTAT= ****** U          1257   1272
$ATYC   005114            1228   1230#
$ATY1   005070            1226#
$ATY3   005076            1172   1227#
$ATY4   005106            1229#  1590
$AUTOB  001300             430#
$BASE   001174             383#   799*  1802*  1857   1936   2024*
$BDADR  001266             425#
$BDDAT  001272             427#
$CDW1   001200             385#  1954
$CDW2   001202             386#  1956   2025*
$CHARC  005064            1189*  1199*  1206   1215*  1220#
$CHTAG  001244             413#
$CH1  = 000006             445#   446#   447#   448#   449#   450#   451#
$CH2  = 000014             445#   446#   447#   448#   449#   450#   451#
$CH3  = 000006             443#   445
$CH4  = 000005             451#   452#   453#   454#   455#   456#
$CPUOP  001146             357#
$CRLF   001357             458#  1188   1223   1400   1560   1561   1571   1652   1816   1834
$DDWD   001204             387#   846   1939   1975
$DDWI   001206             388#
$DDWI10 001230             397#
$DDWI11 001232             398#
$DDWI12 001234             399#
$DDWI13 001236             400#
$DDWI14 001240             401#
$DDWI15 001242             402#
$DDWI2  001210             389#
$DDWI3  001212             390#
$DDWI4  001214             391#
$DDWI5  001216             392#
$DDWI6  001220             393#
$DDWI7  001222             394#
$DDWI8  001224             395#
$DDWI9  001226             396#
$DEVCT  001130             348#  1029*
$DEVM   001176             384#  1940   2027*
$DDRGN  004244            1031   1038   1043   1049#
$E    = 000007               1#  2087   2088#  2203   2204#  2273   2274#  2419   2421#  2504   2505#
$ENDAD  004234             323    768   1045#  1593
```

```
$ENDCT  004220          1040#
$ENV    001140           353#  1167    1235    1259    1587    2421
$ENVM   001141           354#   772    1169    1174    1237
$EOP    004064          1014#  2504
$EOPCT  004212          1037#  1041
$ERFLG  001247           416#   755*   1017*   1074    1103    1105*   1129    1542*   1556    1572*
$ERMAX  001261           422#  1129
$ERROR  006344           313   1531#
$ERRPC  001262           423#   759*   1016*   1082*   1539    1541*
$ERRTB  001325           474#
$ERTTL  001256           420#   758*   1064    1599*
$ETABL  001140           352#
$ETEND  001244           405#   541
$FATAL  001122           345#  1263#
$FFLG   005334          1226*  1229*   1257    1266*   1274#
$FILLC  001322           441#  1192    1223
$FILLS  001321           440#  1223
$FLIP = 177777             1#  2071#  2083#  2192#  2199#  2264#  2269#  2397#  2415#  2491#  2500#
$GOADR  001264           424#
$GOOAT  001270           426#
$GET42  004224          1342#
$HD     000001             10     11
$HIBTS  001446           536#
$ICNT   001250           417#  1111*   1112    1114*   1128
$ILLUP  007402          1669   1685    1704#
$INTAG  001301           431#
$ITEMB  001260           421#  1547*   1589
$LF     001360           459#  1223
$LFLG   005333          1267*  1273#
$LPADR  001252           418#   761*    998*   1001    1118*   1120    1128    1605*   1607    1832*   1833*   1841*   1843
$LPERR  001254           419#
$MADR1  001152           370#
$MADR2  001156           374#
$MADR3  001162           377#
$MADR4  001166           380#
$MAIL   001120           343#   537     541     1117    1167
$MARS1  001150           364#
$MARS2  001154           372#
$MARS3  001160           375#
$MARS4  001164           378#
$MBADR  001450           537#
$MFLG   005332          1227*  1233    1268*   1272#
$MSGAD  001134           350#  1243*   1246
$MSGLG  001136           351#  1248*
$MSGTY  001120           344#  1241    1249*   1261    1265*
$MTYP1  001151           365#
$MTYP2  001155           373#
$MTYP3  001161           376#
$MTYP4  001165           379#
$MXCNT  004542          1115   1128#   1784
$N    = 000005             1#  2071    2083    2088#   2192    2199    2204#   2264    2269    2274#   2397    2415    2421#
                        2491   2500    2505#   2810#
$NULL   001320           439#  1194    1223
$NWTST= 000000          2084#  2200#   2270#   2416#   2501#
$OVER   004500          1087   1090    1101    1113    1119#
$PASS   001126           347#   754*   1023*   1026*   1034*   1035*   1052    1061    1109    1129    2423
```

```
$PASTM  001454          539#
$PUR0D  007376         1702#
$PUR0N  007236          311     753    1669#   1697
$PURMG  007372         1700#
$PURUP  007310         1679    1685#
$QUES   001356          457#    892    1223    1307    1839
$REG0D  001324          443#
$REG0   001326          445#   1384*   1389    2779    2787
$REG1   001330          446#   1383*   1390    2777
$REG2   001332          447#   1382*   1391
$REG3   001334          448#   1381*   1392
$REG4   001336          449#   1380*   1393    2785    2797
$REG5   001340          450#   1379*   1394    2783    2795    2805
$RTN0D  004246         1051#
$SAVR5  007406         1678*   1686    1687*   1688*   1706#
$SCOPE  004300         1081#
$SETUP= 000000         1033    1082
$SVLAD  004462         1098    1116#
$SVPC = 000040          321#    326
$SWR  = 164000            1#      10     456     457    1011    1033    1044    1050    1052    1075    1076    1077    1078
                       1089    1101    1103    1104    1105    1106    1107    1119    1128    1703    2086    2202    2272
                       2418    2503
$SWREG  001142          355#   1959
$SWRMK= 000000         1078
$TESTN  001124          346#   1117*
$TIMES  001354          456#   1033*   1106*   1112    1115*   1128    1784*   2426*
$TKB    001312          436#   1088    1294    1300    1615    1629
$TKS    001310          435#   1086    1292    1619    1627
$TMP0   001342          451#   2462*   2530*   2582    2605*   2658
$TMP1   001344          452#    826     837     838*    937*    962*    963*   1709    2463*   2475*   2624*   2662    2666
                       2807
$TMP2   001346          453#    843*    848*    849*    938*    939    1711    2622*   2625*
$TMP3   001350          454#   2464*   2477*   2623*
$TMP4   001352          455#
$TN   = 000006           10#   2084    2086#   2200    2202#   2270    2272#   2416    2418#   2501    2503#
$TPB    001316          438#   1212*   1223    1300*   1536*   1636*
$TPFLG  001323          442#   1161    1223
$TPS    001314          437#   1210    1223    1298    1534    1633
$TSTM   001452          538#
$TSTNM  001246          415#    760*   1074    1116*   1117    1119    1129    1614    1822    1830    2086*   2202*   2272*
                       2418*   2503*
$TYPE   004606         1141    1161#   1254
$TYPEC  005020         1191    1198    1205    1210#   1211
$TYPEX  005066         1216    1218    1221#
$UNIT   001132          349#
$UNITM  001456          540#
$USWR   001144          356#
$VECT1  001170          381#    811*   1937    2066*
$VECT2  001172          382#
$XTSTR  004340         1084    1092#
$Y    = 000023          645#    652     654#    656#    658#    660#    662#    664#    666#    668#    670#    672#    674#
                        676#    678#    680#    682#    684#    686#    688#    690#
$$GET4= 000000         1044#
$40CAT= ****** U       1089
.     = 017226          304#    305     308#    321     322#    324#    326#    328#    331#    335#    337#    460     493#
                        494#    496#    498#    499#    525     526#    528#    530#    545#    548#    549#    550#    551#
```

```
                    552#   554#   555#   556#   557#   558#   560#   561#   562#   563#   564#   566#   567#
                    568#   569#   570#   572#   573#   574#   575#   576#   578#   579#   580#   581#   582#
                    584#   585#   586#   587#   588#   590#   591#   592#   593#   594#   596#   597#   598#
                    599#   600#   602#   603#   604#   605#   606#   608#   609#   610#   611#   612#   614#
                    615#   616#   617#   618#   620#   621#   622#   623#   624#   626#   627#   628#   629#
                    630#   632#   633#   634#   635#   636#   638#   639#   640#   641#   642#   926#  1052
                   1128   1129   1223   1275#  1664#  1681   1705   1764#  1766#  1768#  1783   2032   2055
                   2101   2135   2217   2234   2240   2306   2314   2331   2334   2359   2372   2387   2774#
.ADVAN  006232       653   1489#
.BEGIN  003770       988#
.BUFSE  006322       689   1520#
.CNVRT  005760       673   1401#
.CONVR  005754       671   1400#
.DCLAS  006200       683   1473#
.DELAY  006212       677   1477#
.DEVIC  006160       675   1465#
.ERRTA  015316      1552   2674#
.INSTE  005442       661   1305#
.INSTR  005336       659   1284#
.INSTI  005356      1288#  1308
.LPRSE  006262       687   1505#
.MSG    005360      1286*  1289#
.PARAM  005462       663   1316#
.PARMD  011142       679   1879#
.PAWCH  010214       681   1747#  1759
.RESOS  005722       669   1389#
.SAVOS  005662       667   1375#
.SCOPE  004300       309   1082#  2067
.SCOPI  004544       655   1133#
.SETFL  010074       665   1718#  1737
.SHIFT  006244       685   1496#
.START  002116       332    749#   761
.TRPSR  006136       315   1454#
.TRPTA  001742       651#  1459
.TYPE   004570       657   1140#
.$ASTA= ****** U    1227   1230
.$X   = 001446       525#   530
```

```
COMMEN   154#
ENDCOM   154#
ERROR     48#  2102  2110  2117  2126  2154  2157  2168  2228  2235  2248  2253  2256  2295  2296
             2307  2315  2332  2335  2343  2346  2349  2352  2360  2369  2388  2479  2549  2569  2577
             2602  2634  2637  2640  2643  2646  2654  2669
ESCAPE   154#
GETPRI   154#
GETSWR   154#
MULT     154#
NEWTST   154#  2084  2200  2270  2416  2501
PASEND     1#  1015
POP      154#  1269  1270  1690  1691
PRGEND     1#  1002
PRGFRT     1#
PUSH     154#  1230  1232  1253  1671  1677
REPORT     1#   154#
SC         1#  1082
SCOPE     49#  1015  2085  2201  2271  2417  2502
SC1        1#  1121
SETPRI   154#
SETUP    154#
SKIP     154#
SLASH    154#
SPACE    154#
STARS    154#   319   338   341   409   522   524   531  1009  1071  1146  1225  1667  1683  2084
            2200  2270  2416  2501
SWRSU    154#
TYPBIN   154#
TYPDEC   154#
TYPNAM   154#
TYPNUM   154#
TYPOCS   154#
TYPOCT   154#
TYPTXT   154#
$BUFFE     1#  1760
$CYCLE     1#  1770
$EOP       1#  1002
$GETFL     1#   814
$GETPA     1#   791   803   821   870   903   949   969  1817
$HEADE     1#    11
$INTSE     1#  2534
$JUNK      1#   547   553   559   565   571   577   583   589   595   601   607   613   619   625
            631   637
$MRESE     1#  2441
$MSG       1#  1707
$PARTS     1#  2491
$SCOPE     1#  1065
$SETFL     1#  1713
$STAG      1#  2138
$STAGF     1#
$TRPDE     1#   652   654   656   658   660   662   664   666   668   670   672   674   676   678
            680   682   684   686   688
$TSTN      1#  2083  2199  2269  2415  2500
$VARIA     1#   334
$XZ        1#  2071  2083  2192  2199  2264  2269  2397  2415  2491  2500
$$CMRE   337#   445   446   447   448   449   450
```

```
MD-11-DVDZB-A   MACY11 30(1046)  28-JUL-77  07:37  PAGE 76
DVDZBA.P11    28-JUL-77 07:37          CROSS REFERENCE TABLE -- MACRO NAMES                    SEQ 0092

$SCMTM    337#    451     452     453     454     455
$SESCA    154#
$SNEWT    154#   2084    2200    2270    2416    2501
$SSKIP    154#
.EQUAT      1#     44
.HEADE      1#
.SETUP      1#
.SACT1      1#    317
.SAPTB      1#    339#
.SAPTH      1#    520
.SAPTY      1#   1223
.SCATC      1#
.SCMTA    337#
.SEOP       1#   1007
.SERRO      1#
.SPOWE      1#   1665
.SSCOP      1#   1069
.STRAP      1#
.STYPE      1#   1144

. ABS.  017226     000

 ERRORS DETECTED:  0

 DVDZBA,DVDZBA.SEQ=DVDZBA.P11
 RUN-TIME: 19 11 1 SECONDS
 RUN-TIME RATIO: 119/32=3.6
 CORE USED:  34K  (67 PAGES)
```