VT61

EXERCISER

MD-11-DZVTH-A

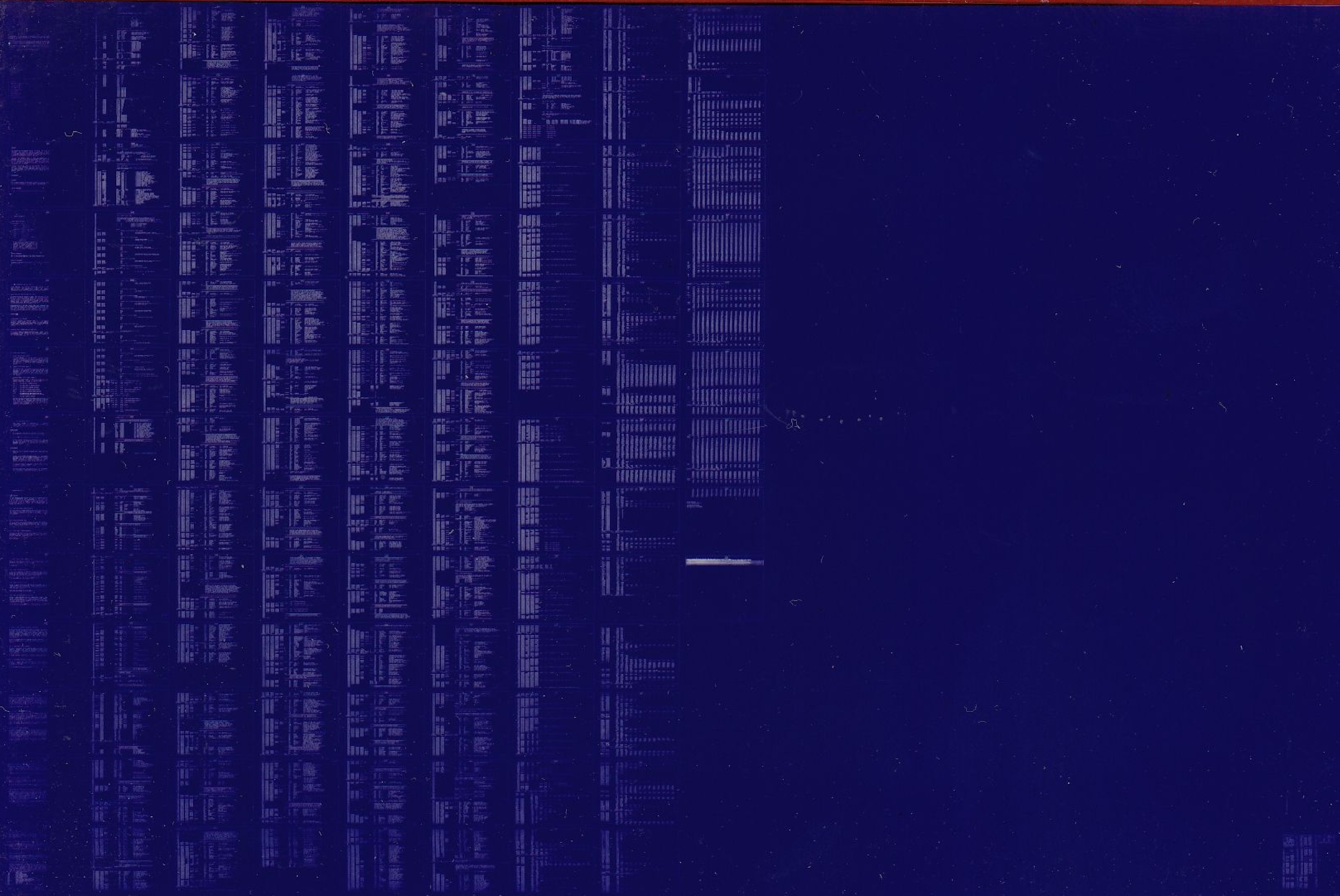
EP-DZVTH-A-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA



IDENTIFICATION

Product Code:	MAINDEC-11-DZVTH-A
Product Name:	VT61 Exerciser
Date:	30-JAN-76
Maintenance:	Diagnostic Group
Author:	Paul Nelson

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation assumes no responsibility for any errors that may appear in this manual.

The software described in this document is furnished to the purchaser under a license for use on a single computer system and can be copied (with inclusion of Digital's copyright notice) only for use in such system, except as may otherwise be provided in writing by Digital

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Digital.

Table of Contents

1. ABSTRACT

This program is an acceptance test for the entire VT61 family of terminals. The functional testing is based upon a set of terminal functions which are common throughout the entire family of vt61 type terminals. The functions and their derived testing is designed to completely check(at the functional level) the terminal micro-processor and associated rams.

There are two distinct modes in which the program can be operated. In "auto" mode all DL11's with operational VT61's will be mapped and all will be tested sequentially. All tests which do not require manual intervention or visual screen observation (Tests 1 thru 20) will be executed for each VT61 repetitively. All errors will be reported on the system console (which is not tested even if it is a VT61).

In Manual mode console entry of the addresses and tests is required. The addresses and tests can be entered in a non-sequential manner and the subsequent execution will follow the entry sequence. This mode must be utilized to enter the keyboard tests, data loop test, and printer controller test. Sequence completion will exit to the re-start point for the manual test.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP 11 family computer with 8K words of memory, a console, and up to 16 VT61's connected to the host computer via DL11-A,B,C or D. VT61 must be in remote; full duplex and at least 300 baud.

3. LOADING PROCEDURE

Procedure for normal binary papertapes should be followed.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

Standard PDP 11 format

SW15 = 1	Halt on error.
SW14 = 1	Loop on test
SW13 = 1	Inhibit error typeouts
SW11 = 1	Inhibit iterations
SW10 = 1	Bell on error
SW9 = 1	Loop on error
SW8 = 1	Loop on test in SWR<7:0>

SPECIAL NOTE

If the computer utilized is a LSI 11 or a computer without a switch register. The program will utilize locations 174 and 176 as a "display" register and a "switch" register respectively. The operator will be responsible for the loading of the "switch" register location prior to starting or restarting the program.

4.2 STARTING ADDRESSES

200 is the starting address of the "Auto" acceptance test
204 is the starting address on the "Manual" select test.

5. OPERATING PROCEDURE

Following is the operating procedure for the "auto" and "manual" modes of testing.

5.1 AUTO ACCEPTANCE MODE (SA = 200).

In this mode the only operator intervention required is SWR option selections such as loop on test (SWR 11), bell on error (SWR C), ect.. The program will, without any external intervention, locate the DL11's with VT61 type units attached and sequentially test all units repetitively with tests 1 thru 20.

5.2 MANUAL UNIT/TEST SELECTION MODE (SA = 204)

This mode requires the operator to enter the addresses of the DL11's to be tested (format is 17XXXX, ect, -up to 16 entries). The entries must be separated by commas and terminated with a carriage return. The operator must then, upon program request, enter a list of tests to be executed in the same format as the address entry (i.e.yy,zz,c/r). Preceeding the terminating carriage return with a 377 octal will result in the tests being repetitively executed for all addresses entered.

Simply depressing a carriage return when unit addresses are requested will result in the mapping and testing of all good DL11's with operational VT61's attached. However, the test list must still be entered via the console!! When running the exercisor in manual mode a control C (03 octal) will result in the termination of testing at the end of the current subtest.

6. ERRORS-GENERAL

6.1 NO OPERATIONAL VT61 ATTACHED

If the unit selected (in "manual" mode) or in the mapping operation ("auto" mode) does not result in a unit which is capable of responding to the test the message "NO VT61 RESPONDED TO ESCZ SEQ. AUTO RETRY IN 30 SEC". will be displayed on the console every 30 seconds until the test is stopped or a unit responds.

6.2 EXCESSIVE "FATAL" ERRORS FROM UNIT UNDER TEST

If ten fatal errors (incomplete transmit/recieve cycles) occurs the message "TESTING ABORTED-TOO MANY FATAL XMITS" will be displayed and the test will exit to the initial setup sequence of the requested mode. If the test then locates an operational unit, it will begin testing it.

6.3 COMMON ERROR MESSAGES

a. ESCAPE SEQUENCE ERROR (ERROR 1)

This error message is returned when a specific escape sequence did not elicit the expected response from the unit under test. Message returns test #, error program count and two words which contain up to 4 bytes of the failing escape sequence (i.e. if "TRANSMIT ALL" failed;the ESC,O, V would be displayed in the format BYTE 1+2=015517,BYTE 3+4=000126).

b. RECEIVE STATUS ERROR (ERROR 2)

This error message is returned if any of bits 12, 13, or 14 are set in the interface receive buffer register. Data displayed is the address of the CSR (Control and Status Register) of the failing unit. The contents of the forementioned CSR , the error bits from the receive buffer register, and the character which was stored when the errors were detected.

c. SOFTWARE STATUS (VSTAT) ERROR (ERROR 3)

The location tagged "VSTAT" is used by the program to store dynamic conditions relating to the unit under test. The bits which may cause a software status error are:

BIT 15	SET FOR XOFF, CLEARED FOR XON
BIT 14	SET WHEN START OF MESSAGE RECEIVED
BIT 13	SET WHEN END OF MESSAGE RECEIVED
BIT 12	SET FOR A PERIPHERAL ABORT MESSAGE
BIT 10	SET WHEN AN INTERFACE ERROR DETECTED
BIT 7	SET WHEN AN XOFF WAS DETECTED AND THE TRANSMITTER WAS SHUT DOWN BY THE SOFTWARE.
BIT 1	SET WHEN TRANSMIT COMPLETE

The only bit which will unconditionally cause this error is BIT 12 (Peripheral Abort) all other bits will be set and reset and an error is dependent upon expected conditions (i.e. after a complete transmission bits 1, 13 and 14 must be set and others mentioned reset or an error will be reported). Data displayed is the pass #, the test #, expected status and actual status.

D. VT61 HUNG ERROR (ERROR 11)

This error message is displayed if a complete transmission(s) does not result in a SOM(s), an EOM(s) and transmit done. This error is a fatal error and ten of these errors will result in the test aborting.

7. RESTRICTIONS

A. It is imperative that bcth the interface and the VT61 should be placed in full duplex and remote (not local) mode.

B. Unit to be tested cannot be the console device.

C. For the automatic test mapping of the DI11's, all addresses for the units to be tested must be within the standard DEC addresses and vectors. if this is not the case, the procedure outlined in Section 8-B must be followed before testing is begun.

8. MISCELLANEOUS

A. Execution time for the auto selection tests (test 1-20) with units set to a baud rate of 9600 baud is approximately 90 seconds.

B. To test a device (DI11 with vt61 attached) at non-standard addresses the location "STRTAB" can be modified to contain the lowest of the non-standard addresses and locaton "ENDTAB" modified to contain the highest non-standard address. All interfaces within the new addresses will be mapped and tested if the proper responses are obtained.

C. To change the number of fatal errors allowed before testing is aborted, location "ALWCNT" (loaded with 10) can be modified to the desired count.

D. All tests except Test 1 and Test 23 are run in MAINTENANCE mode, therefore all transmissions from the vt61 are expected to be preceeded by a SOM and terminated with a EOM.

9. PROGRAM DESCRIPTION

9.0 INITIALIZATION

In "Auto" sequence mode this section of the test maps all devices in the pre-determined areas. Devices are then tested for interrupt capability via the "MAINTENANCE" bit and all units which do not or cannot respond are purged from the table. All units are then issued the "ESCAPE Z" sequence and those which do not respond, or do not respond with the proper "IDENT" are purged. All operational units are stored in a table(DLTBL) and tested sequentially.

9.1 TEST 1 CHECK ALL COMMON ESCAPE SEQUENCES.

This test issues all escape sequences and insures the VT61 has not failed during an ESC sequence by issuing a ESC Z to force a VT61 response. The purpose of the test is to attempt to insure that subsequent tests will not result in a "hung" unit. Data is not evaluated.

All errors are reporteded as Escape Sequence failures(Error 1).

9.2 TEST 2 CHECK MAINTENANCE MODE.

Routine to insure entering maintenance mode causes SOM and EOM to be appended to all transmits from VT61 under test. Maintenance mode is entered, then an escape Z sequence is issued to the unit and the resulting response from the vt61 is checked for SOM/EOM.

Error 22 will be issued if either component(SOM/EOM) is missing.

9.3 TEST 3 CHECK DIRECT CURSOR ADDRESSING

This test insures that the cursor will respond to direct cursor addressing. The unit is reset and the cursor position is verified to be home. The cursor is then moved to row 23 column 80 and the position is again verified.

Cursor positioning errors(ERROR 7) are reported if the positions are incorrect.

9.4 TEST 4 CHECK LINEAR ADDRESSING MODE.

Routine to insure the unit can enter linear addressing mode. 81 characters are issued to the unit under test then the cursor position is read and must be rowl, col.0.

An Escape Sequence error (ERROR 1)is issued if the cursor is not at rowl,col.0

9.5 TEST 5 CHECK XON/XOFF FROM VT61

Test to insure operation of XON/XOFF commands from VT61. XOFF is forced by transmitting the data on line 23 while simultaneously filling the silo with new data. After sensing the XOFF, the test waits for the transmit to finish and insures XON occurs before the maximum transfer time has elapsed. (30 seconds)

Errors are reported if the format of ERROR 3(VSTAT errors) and will reflect either lack or excess of Bit 15.

9.6 TEST 6 CHECK XON/XOFF TO VT61

Routine to verify operation of XOFF and XON to the VT61. A full screen transmit is initiated and a series of XOFFs and XONs are issued to the terminal sequentially Errors are reported if a XOFF does not stop, or a XON restart the transmission. Test is ended when EOM is sensed.

Errors are reported(Error 15 for XOFF failure and Error 16 for a XON failure)as specific error messages.

9.7 TEST 7 CHECK RAM AND COMMUNICATIONS PATHS

Routine to test VT61 RAM and the communication paths. This routine issues a series of full screen patterns (77/100, 100/77, 52/125, incrementing, and rev. video incrementing) to the VT61. The full screen is then transmitted to the host and after each iteration received data is checked and all errors (including transmission) are reported.

Errors reported could be ERROR 2 for a Receive Status error,ERROR 4 for data errors and ERROR 5 for a Receive Byte Count error.

9.10 TEST 10 CHECK TRANSMIT AND RECEIVE CHECKSUMS.

Routine to test the ability of the VT61 to calculate and transmit checksums of both transmitted and received data. Subtest "A" transmits a full buffer updating a calculated checksum on each character transmitted. An escape sequence requesting the receiver checksum is embedded at the end of xmit buffer and the received checksum is compared to the calculated. Subtest "B" performs the same type of check on the VT61 transmit checksum, utilizing the data sent to the VT61 in subtest "A", during a full screen transmit.

Error 13 is issued(with calculated and received checksum) if a Receive Checksum error is detected. Error 14 is issued (with same data as ERROR 13) if a VT61 Transmit Checksum error is detected.

9.11 TEST 11 CHECK BASIC CURSOR COMMANDS

Routine to insure basic cursor commands result in correct cursor movement. Commands are issued in the sequence: reset, cursor right, cursor down, cursor left, and cursor up. The read cursor position command is issued after every move cursor command and received position is compared to the expected position and any errors reported.

An Escape Sequence error(Error 1) and a Cursor Positioning error(Error 6) are issued if any functions are detected to fail.

9.12 TEST 12 CHECK READ CHARACTER AT CURSOR

Routine to insure that read character at cursor functions correctly. Command sequence is: reset, A, cursor left, read character at cursor. An error is reported if the character received is not an "A".

An Escape Sequence error(Error 1) and a Data Compare error(Error 4) are issued if a failure is detected.

9.13 TEST 13 CHECK REPLACE AND INSERT CHARACTER MODES

Routine to verify operation of replace and insert mode. Initially row 0 is written to 80 incrementing characters; on the first pass (replace mode) a character(172) is replaced at the home position and the characters at row0, col.0 and row1, col.0 are read and verified to be a "172" and a "Null" respectively. On the second pass, insert mode is entered and the resulting insertion (at the home position) is verified. Row0, col.0 should be "172" and row1, col.0 should be "161".

If an error is detected in either mode,the appropriate Escape Sequence error(Error 1) is issued.

9.14 TEST 14 CHECK VT61 SCROLL CAPABILITIES.

Routine to insure VT61 will scroll if a line feed is issued from row 23 or a data insert from row 23 col. 79. In subtest "A", row 0 is initially written to a 0 and row 1 A 1. After completion of a line feed (and resulting scroll) row 00, col.00 is expected to contain A 1. In subtest "B", the cursor is placed at row23, col.79 and a data character "A" is entered. The cursor position is then read and should be row23, col.00. The char. at home is verified to be a null.

A Scroll error(Error 23) is issued if either functions fail to elicit the proper response from the unit under test. the ERROR PC will distinguish between the failing functions.

9.15 TEST 15 CHECK ALL SCREEN ADDRESSES.

This test insures that the VT61 cursor can be positioned to every possible row/column position on the screen. This is tested by filling the complete screen (except Row 23,Col.79 which will contain a "Null") with the character "A" and then positioning the cursor (via DCA) to every position and the "A" at that position is replaced with a space(octal 40). The screen is then read to verify that only spaces exist on the screen. All positions containing non-spaces are reported.

All errors detected will be reported as Direct Cursor Address erros(Error 7),and will contain the position the bad data(non -space) was detected at.

9.16 TEST 16 CHECK LINE FEED AND CARRIAGE RETURN

Routine to insure proper operation of carriage return and line feed during normal mode. Initially the cursor is set (via D.C.A.) to row0, col 20 and a line feed is issued, the cursor position is then read and must be row1, col.20. A carriage return is then issued and cursor position verified to be row1, col0.

An Escape Sequence error(Error 1) and a Cursor positioning error(Error 6) will be issued if an error is detected.

9.17 TEST 17 CHECK ERASE TO END OF SCREEN

Routine to verify proper operation of erase to end-of-screen. Screen is written to 1920 incrementing char. Erase to end of screen is then issued and the entire screen is read verifying that it is all nulls.

If any non-null positions are detected, and Escape Sequence error (Error 1) and a Data error(Error 4) will be issued.

9.20 TEST 20 CHECK SELF TEST, COPIER, AND ISSUE END OF PASS.

SELF TEST (ESC T) is issued to the unit under test and an Self Test error(Error 10) is issued if the unit cannot respond to an "Escape Z" sequence after self test is complete. If self test is successful the screen is written to 23 lines of incrementing characters and 23 lines of incrementing char. in reverse video. The "Ident" is then checked and if a copier is present a copy screen command is issued (NOTE: This command will cause the unit to be "busy" and not respond to any further commands until the screen has been completely copied.)

If the Ident indicates a copier is present and the COPY SCREEN is initiated,but not completed, a "PERIPHERAL ABORT" (Error 20) Error is issued.

END OF AUTO-ACCEPTANCE TESTS

9.21 TEST 21 KEYBOARD ECHO TEST

Routine to echo the keyboard. Keys for tab, bell, carriage and line feed echc a mnemonic, non-display char. echo octal equivalents and display char. echo themselves. (examples-char.. space, ESC, space or 037, space.) A Control C (003) will cause a test exit.

9.22 TEST 22 TEST A LINE PRINTER(PRINTER CONTROLLER MODE)

Routine to utilize the VT61 as a printer controller. After test message is displayed, the test waits for a C/R before actually entering test. A pattern of incrementing, rolling char. will be outputted until a Control C (003) is received.

If the Line Printer is disabled after the initialization of the test, a "PERIPHERAL ABORT" (Error 20) is issued.

9.23 TEST 23 UNIT SIMULATOR TEST

Routine to loop data/commands from the VT61 back to the VT61. Data transmissions resulting from a ESC sequence will also be looped and will enter the screen at the cursor position. This test can be used to simulate, or create, specific screen patterns and operations. A control C (003) exits test.

9.24 TEST 24 PRODUCTION KEYBOARD TEST

Production keyboard test. All keys must be depressed in the sequence indicated on the screen. All errors or mistakes are displayed in octal positional format and the correct key position in the row is displayed in decimal. This test is run in maintenance mode, therefore the keys will echo their position, not their indicated mnemonic. The exceptions are the individual tests for the shift and control functions.These tests are explicitely defined by messages to the operator. 10 errors will cause an automatic exit from test.

```
     1                                          .NLIST  MD,MC,CND
     2                                          .LIST   ME
     3                                  .TITLE  MAINDEC-11-DZVTH-A
```

B02

```
                        ;*COPYRIGHT (C) 1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY P. NELSON
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-B1),AUG 19,1975.
                        ;*

                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        11            INHIBIT ITERATIONS
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         8            LOOP ON TEST IN SWR<7:0>

                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
        001100          STACK=  1100
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
        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
                        .EQUIV  R6,SP           ;;STACK POINTER
                        .EQUIV  R7,PC           ;;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

                                      .SBTTL  TRAP CATCHER

         000000                               .=0
                                      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
         000174                               .=174
 000174  000000                               DISPREG: .WORD  0                 ;;SOFTWARE DISPLAY REGISTER
 000176  000000                               SWREG:   .WORD  0                 ;;SOFTWARE SWITCH REGISTER
         000200                               .=200
 000200  000137  002230               START:  JMP     AUTO             ;USE AUTO SELECTION OF UNITS
 000204  000137  002262               MSTRT:  JMP     MANS             ;ALLOW OPERATOR SELECTIONOF UNITS/TESTS
```

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

                                      .SBTTL  COMMON TAGS

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

         001100                               .=1100
 001100                               $CMTAG:                           ;;START OF COMMON TAGS
 001100  000000                       $PASS:  .WORD   0                 ;;CONTAINS PASS COUNT
 001102     000                       $TSTNM: .BYTE   0                 ;;CONTAINS THE TEST NUMBER
 001103     000                       $ERFLG: .BYTE   0                 ;;CONTAINS ERROR FLAG
 001104  000000                       $ICNT:  .WORD   0                 ;;CONTAINS SUBTEST ITERATION COUNT
 001106  000000                       $LPADR: .WORD   0                 ;;CONTAINS SCOPE LOOP ADDRESS
 001110  000000                       $LPERR: .WORD   0                 ;;CONTAINS SCOPE RETURN FOR ERRORS
 001112  000000                       $ERTTL: .WORD   0                 ;;CONTAINS TOTAL ERRORS DETECTED
 001114     000                       $ITEMB: .BYTE   0                 ;;CONTAINS ITEM CONTROL BYTE
 001115     001                       $ERMAX: .BYTE   1                 ;;CONTAINS MAX. ERRORS PER TEST
 001116  000000                       $ERRPC: .WORD   0                 ;;CONTAINS PC OF LAST ERROR INSTRUCTION
 001120  000000                       $GDADR: .WORD   0                 ;;CONTAINS ADDRESS OF 'GOOD' DATA
 001122  000000                       $BDADR: .WORD   0                 ;;CONTAINS ADDRESS OF 'BAD' DATA
 001124  000000                       $GDDAT: .WORD   0                 ;;CONTAINS 'GOOD' DATA
 001126  000000                       $BDDAT: .WORD   0                 ;;CONTAINS 'BAD' DATA
 001130  000000                               .WORD   0                 ;;RESERVED--NOT TO BE USED
 001132  000000                               .WORD   0
 001134  000000                               .WORD   0
 001136  177570                       SWR:    .WORD   DSWR              ;;ADDRESS OF SWITCH REGISTER
 001140  177570                       DISPLAY: .WORD  DDISP             ;;ADDRESS OF DISPLAY REGISTER
 001142  177560                       $TKS:   177560                    ;;TTY KBD STATUS
 001144  177562                       $TKB:   177562                    ;;TTY KBD BUFFER
 001146  177564                       $TPS:   177564                    ;;TTY PRINTER STATUS REG. ADDRESS
 001150  177566                       $TPB:   177566                    ;;TTY PRINTER BUFFER REG. ADDRESS
 001152     000                       $NULL:  .BYTE   0                 ;;CONTAINS NULL CHARACTER FOR FILLS
 001153     002                       $FILLS: .BYTE   2                 ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
 001154     012                       $FILLC: .BYTE   12                ;;INSERT FILL CHARS. AFTER A "LINE FEED"
 001155     000                       $TPFLG: .BYTE   0                 ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
 001156  000000                       $TIMES: 0                         ;;MAX. NUMBER OF ITERATIONS
 001160  000000                       $ESCAPE:0                         ;;ESCAPE ON ERROR ADDRESS
 001162  177607  000377               $BELL:  .ASCIZ  <207><377><377>   ;;CODE FOR BELL
 001166     077                       $QUES:  .ASCII  /?/               ;;QUESTION MARK
 001167     015                       $CRLF:  .ASCII  <15>              ;;CARRIAGE RETURN
 001170  000012                       $LF:    .ASCIZ  <12>              ;;LINE FEED
```

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

                                    .SBTTL  ERROR POINTER TABLE

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

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

        001172                      $ERRTB:
                                            ;GENERAL ESCAPE SEQUENCE ERROR MESSAGE

        001172  023201                      EM1                     ;AN ESCAPE SEQUENCE TO VT61   FAILED.
        001174  023266                      DH1                     ;TEST#,ERROR PC,2 SEQUENCE BYTES,2 SEQUENCE BYTES.
        001176  001422                      DT0
        001200  001442                      DF0

                                            ;RECEIVE STATUS ERROR MESSAGE

        001202  023331                      EM2                     ;RECEIVE STATUS ERROR
        001204  023361                      DH2                     ;ADDRESS,STATUS ,ERR. BITS,CHAR.
        001206  001452                      DT2
        001210  001442                      DF0


                                            ;RECIEVE SOFTWARE STATUS ERROR MESSAGE.

        001212  023420                      EM3                     ;SFTWARE (SIAT) STATUS ERROR
        001214  023461                      DH3                     ;PASS#,TEST#,GOOD STATUS,RECEIVED STATUS
        001216  001500                      DT4
        001220  001543                      DF6

                                            ;DATA ERROR

        001222  023530                      EM4                     ;DATA EXPECTED DOES NOT MATCH RECEIVE DATA.
        001224  023574                      DH4                     ;TEST#,REC.CNT.,EXPECTED DATA, RECEIVE DATA
        001226  001512                      DT5
        001230  001442                      DF0

                                            ;RECEIVE BYTE COUNT ERROR

        001232  023642                      EM5                     ;BYTES EXPECTED DOES NOT EQUAL BYTES RECEIVED.
        001234  023721                      DH5                     ;BYTES EXPECTED, BYTES RECEIVED
        001236  001434                      DT1
        001240  001450                      DF2

                                            ;GENERAL DIRECT CURSOR ADDRESS FAILURE

        001242  023752                      EM6                     ;CURSOR POSITION ERROR
```

```
        001244  024005                      DH6                     ;GD LINE, GD COL., BD LINE, BAD COL.
        001246  001452                      DT2
        001250  001474                      DF3
```

```
                                  ;DIRECT CURSOR ADDRESS ERROR

001252  024052                    EM7               ;DIRECT CURSOR ADDRESS ERROR
001254  024152                    DH10              ;PASS#,TEST#,BD. ROW,BD. COL.
001256  001500                    DT4
001260  001543                    DF6



                                  ;LAST TEST-SELF TEST FAILED

001262  024354                    EM10              ;VT61 FAILED SELF-TEST FUNCTION
001264  024723                    DH11              ;CSR, VECTOR
001266  001434                    DT1
001270  001446                    DF1

                                  ;VT61 FAIL/HUNG ERROR MESSAGE
001272  024211                    EM11              ;LAST TRANSMISSION TO VT61 CAUSED VT61 TO FAIL/HANG
001274  024115                    DH7               ;PASS#,TEST#,ERROR PC
001276  001464                    DT3
001300  001534                    DF4

                                  ;GENERAL TEST FAILURE-PRECEEDS DATA/POSITION ERROR

001302  024277                    EM12              ;VT61 UNDERR TEST FAILED-ERROR DATA FOLLOWS
001304  024115                    DH7               ;PASS#,TEST#,ERROR PC.
001306  001464                    DT3
001310  001534                    DF4

                                  ;RECEIVE CHECKSUM ERROR

001312  024530                    EM13              ;VT61 RECEIVER CHECKSUM ERROR
001314  024415                    DH12              ;PASS#,TEST#,GD.CKSUM,BD CKSUM
001316  001500                    DT4
001320  001543                    DF6

                                  ;TRANSMITTER CHECKSUM ERRPOR

001322  024577                    EM14              ;VT61 TRANSMITTER CHECKSUM ERROR
001324  024415                    DH12
001326  001500                    DT4
001330  001543                    DF6




                                  ;XOFF FAILED TO HALT BLOCK XMIT

001332  025016                    EM15              ;XOFF TO VT61 FAILED TO HALT BLOCK XMIT
001334  025533                    DH13              ;PASS,TEST,VSTAT
001336  001524                    DT6
001340  001534                    DF4

                                  ;XON FAILED TO RESTART BLOCK XMIT
```

```

001342  025067                    EM16              ;XON TO VT61 FAILED TO RESTART BLOCK XMIT
001344  025533                    DH13
001346  001524                    DT6
001350  001534                    DF4

                                  ;NO XON AFTER UNIT WAS RESET

001352  025142                    EM17              ;NO XON AFTER UNIT WAS RESET.
```

G02

```
001354  024115                          DH7                             ;PASS#,TEST#,ERROR PC
001356  001524                          DT6
001360  001534                          DF4

                                        ;PERIPHERAL ABORT ERROR

001362  025220                          EM20                            ;LAST PERIPHERAL OPERATION ABORTED.
001364  025565                          DH14                            ;PASS,TEST,ERROR PC, VSTAT
001366  001500                          DT4
001370  001543                          DF6

                                        ;CANT CLEAR PERIPHERAL ABORT FLAG.

001372  025264                          EM21                            ;COULD NOT CLEAR LAST ABORT FLAG.
001374  025565                          DH14
001376  001500                          DT4
001400  001543                          DF6

                                        ;MAINTENANCE MODE DID NOT FORCE A SOM/EOM.

001402  025327                          EM22                            ;SOM OR EOM NOT REC. IN MAINT. MODE.
001404  023461                          DH3                             ;PASS#,TEST#,EXP.STAT, ACT.STAT
001406  001500                          DT4
001410  001543                          DF6

                                        ;LINE FEED OR CURSOR RIGHT AT ROW 23 DID NOT CAUSE A SCROLL.

001412  025415                          EM23                            ;NO SCROLL FROM LINE FEED OR CURSOR RIGHT.
001414  024115                          DH7
001416  001524                          DT6
001420  001534                          DF4

001422  002226  001116  001124  DT0:    .WORD   TSTNM,$ERRPC,$GDDAT,$BDDAT,0
001430  001126  000000
001434  001124  001126  000000  DT1:    .WORD   $GDDAT,$BDDAT,0
001442     000     000     000  DF0:    .BYTE   0,0,0,0
001445     000
001446     000     000          DF1:    .BYTE   0,0

001450     001     001          DF2:    .BYTE   1,1                     ;DECIMAL TYPE

001452  001120  001124  001122  DT2:    .WORD   $GDADR,$GDDAT,$BDADR,$BDDAT,0
001460  001126  000000
001464  001100  002226  001116  DT3:    .WORD   $PASS,TSTNM,$ERRPC,0
001472  000000
001474     001     001     001  DF3:    .BYTE   1,1,1,1
001477     001
```

```
001500  001100  002226  001124  DT4:    .WORD   $PASS,TSTNM,$GDDAT,$BDDAT,0
001506  001126  000000
001512  002226  001120  001124  DT5:    .WORD   TSTNM,$GDADR,$GDDAT,$BDDAT,0
001520  001126  000000
001524  001100  002226  001120  DT6:    .WORD   $PASS,TSTNM,$GDADR,0
001532  000000
001534     001     000     000  DF4:    .BYTE   1,0,0
001537     000     000     001  DF5:    .BYTE   0,0,1,1
001542     001
001543     001     000     000  DF6:    .BYTE   1,0,0,0
001546     000
        001550                          .EVEN
                                                ;INSTRUCTION DEFINITIONS
        022626                  POP2SP  =22626
        024646                  PUSH2SP =24646

                                ;*************************************************************
                                        ;DEFINITION SOFTWARE STATUS(VSTAT) REGISTER BITS
                                ;*************************************************************

        100000                  RXOFF   =100000                 ;SET FOR XOFF, CLEARED FOR XON
        040000                  RSOM    =040000                 ;SET FOR SOM (START OF MESSAGE).
        020000                  REOM    =020000                 ;SET FOR EOM (END OF MESSAGE).
        010000                  PABRT   =010000                 ;SET FOR A PERIPHERAL ABORT.
        004000                  RSTT    =004000                 ;SET FOR RECEIVE STATUS ERROR.
        002000                  CKSUM   =002000                 ;SET TO CALCULATE 61 REC.  CHECKSUM
        001000                  EPL     =001000                 ;SET WHEN END OF LINE DETECTED
        000400                  ESC     =000400                 ;SET WHEN OCTAL 33 RECEIVED.
        000200                  XMKIL   =000200                 ;SET WHEN TRANSMIT KILLLED.
        000100                  TXSUM   =000100                 ;SET TO CALCULATE 61 XMIT CHECKSUM
        000040                  REVID   =000040                 ;SET WHEN REVERSE VIDEO MODE RECEIVED.
        000020                  COMGP   =000020                 ;SET TO CONVERT REC. CHAR. BY -137.
        000004                  CURPOS  =000004                 ;SET WHEN CURSOR POS. RECEIVED
        000002                  TRMID   =000002                 ;SET WHEN TERMINAL I.D. RECEVED.
        000001                  XMDNE   =000001                 ;SET UPON TRANSMIT COMPLETE

                                ;*************************************************************
                                                ;DEFINITION OF DL11 CONTROL BITS
                                ;*************************************************************

        000200                  RECDN   =200
        100000                  DSCHNG  =100000
        000100                  RDENA   =000100
        100000                  RERR    =100000
        040000                  RORUN   =40000
        020000                  RFMER   =20000
        010000                  RPAR    =10000
        000200                  TRDY    =00200
        000100                  TENA    =00100
        000004                  MAINT   =00004
        000104                  TCOMB   =00104                  ;COMBINATION INTERRUPT ENABLE AND MAINT.

        003600          TOTCH   =1920.                  ;TOTAL CHARACTERS ON SCREEN
        003601          TOTC1   =1921.                  ;TOTAL SCREEN +1
                        ;*************************************************************
                                ;FOLLOW ARE DL11 ADDRESS AND VECTOR STORAGE TABLES
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 9
DZVTH.P11       ERROR POINTER TABLE                                                     SEQ 0020

                        ;*************************************************************
001550  000020          VVECT:  .BLKW   20              ;GOOD DL11 VECTOR TABLE
001610  000020          DLTBL:  .BLKW   20              ;GOOD DL11 ADDRESS TABLE
001650  000020          INTAB:  .BLKW   20              ;TABLE OF POSSIBLE DL11 ADDRESSES


                                ;CURRENT POINTERS FOR ADDRESSES AND VECTORS
001710  000000          VECPT:  .WORD                   ;VECTOR INDEX
001712  000000          DLTPT:  .WORD                   ;ADDRESS INDEX
                                        ;ADDRESS TABLES FOR DL11 INTERFACES
001714  176500          STRTAB: .WORD   176500          ;DL11A/B
001716  175610                  .WORD   175610          ;DL11 C/D/E
001720  000000                  .WORD   0
001722  176676          ENDTAB: .WORD   176676          ;DL11 A/B
001724  176170                  .WORD   176170          ;DL11 C/D/E
001726  000000                  .WORD   0               ;END OF LIST MARKER
                        ;*************************************************************
                                        ;VT61 ADDRESSES IN TABLE REFLECT UNIT UNDER TEST
                        ;*************************************************************
001730  000000          VRCSR:  .WORD   0
001732  000000          VRBUF:  .WORD   0               ;RECEIVE DATA BUFFER
001734  000000          VXCSR:  .WORD   0               ;XMITTER CSR
001736  000000          VXBUF:  .WORD   0               ;XMITTER DATA BUFFER
001740  000000          VECT:   .WORD   0               ;VECTOR FOR UNIT UNDER TEST
001742  000000          CRCSR:  .WORD   0               ;CONSOLE RECEIVE CSR
001744  000000          CRBUF:  .WORD   0               ;CONSOLE DATA BUFFER


                        ;*************************************************************
                                ;TABLE OF VT61 COMMAND AND SEQUENCES
                        ;*************************************************************

        000007          .BEL    =007
001746  000007          BEL:    .WORD   007             ;BELL
        000015          .CARRT  =015
001750  000015          CARRT:  .WORD   015             ;CARRIAGE RETURN
        000012          .LNFED  =012
001752  000012          LNFED:  .WORD   012             ;LINE FEED
        000011          .TAB    =011
001754  000011          TAB:    .WORD   011             ;TAB
                        ;*************************************************************
001756  000001                  .WORD   01              ;TABLE DELIMITER (ESCN)
                        ;*************************************************************

        000110          .CHOM   =110
001760  000110          CHOM:   .WORD   110             ;HOME CURSOR  H

        000103          .CRT    =103
001762  000103          CRT:    .WORD   103             ;CURSOR RIGHT  C

        000102          .CDWN   =102
001764  000102          CDWN:   .WORD   102             ;CURSOR DOWN  B

        000104          .CLFT   =104
001766  000104          CLFT:   .WORD   104             ;CURSOR LEFT  D
```

```
                000101          .CUP    =101
        001770  000101          CUP:    .WORD   101             ;CURSOR UP  A

                000112          .EOS    =112
        001772  000112          EOS:    .WORD   112             ;ERASE TO END OF SCREEN  J


                000127          .EPNT   =127
        001774  000127          EPNT:   .WORD   127             ;ENABLE PRINT MODE.  W
                000130          .DPNT   =130
        001776  000130          DPNT:   .WORD   130             ;DISABLE PRINT MODE  X

                                ;************************************************************
        002000  000002                  .WORD   2               ;TABLE DELIMITER (ESCO)
                                ;************************************************************


                000101          .EMAIN  =101
        002002  000101          EMAIN:  .WORD   101             ;ENTER MAINTENANCE MODE  A
                000141          .DMAIN  =141
        002004  000141          DMAIN:  .WORD   141             ;EXIT MAINTENANCE MODE  SA

                000105          .LKKB   =105
        002006  000105          LKKB:   .WORD   105             ;LOCK KEYBOARD  E
                000145          .UNLKKB =145
        002010  000145          UNLKKB: .WORD   145             ;UNLOCK KEYBOARD  SE

                000103          .DRECT  =103
        002012  000103          DRECT:  .WORD   103             ;ENABLE LINEAR MODE  C

                000133          .CLRCK  =133
        002014  000133          CLRCK:  .WORD   133             ;CLEAR RECEIVER CHECKSUM I

                000134          .CLTCK  =134
        002016  000134          CLTCK:  .WORD   134             ;CLEAR TRANSMITTER CHECKSUM


                000112          .EEMP   =112
        002020  000112          EEMP:   .WORD   112             ;ENABLE REVERSE  VIDEO  J
                000152          .DEMP   =152
        002022  000152          DEMP:   .WORD   152             ;DISABLE REVERSE VIDEO  SJ

                000137          .IABT   =137
        002024  000137          IABT:   .WORD   137             ;INITIALIZE ABORT FLAG  -

                                ;************************************************************
        002026  000003                  .WORD   3               ;TABLE DELIMITER (ESCAPE P)
                                ;************************************************************

                000131          .EAPNT  =131
        002030  000131          EAPNT:  .WORD   131             ;ENABLE AUTO PRINT MODE  Y
                000171          .DAPNT  =171
        002032  000171          DAPNT:  .WORD   171             ;DISABLE AUTO PRINT MODE  SY

                000111          .EINST  =111
        002034  000111          EINST:  .WORD   111             ;ENABLE INSERT  I
```

```
                000151          .ERPL   =151
```

```
      002036  000151                ERPL:   .WORD   151             ;ENABLE REPLACE  SI



                                    ;************************************************************
      002040  000004                        .WORD   4                       ;TABLE DELIMITER (I/O)
                                    ;************************************************************

              054433                .DCRAD  =054433
      002042  054433                DCRAD:  .WORD   054433                  ;DIRECT CURSOR ADDRESSING
              067467                .R23C79 =067467
      002044  067467                R23C79: .WORD   067467                  ;CURSOR TO LOWER RIGHT
      002046  000000                        .WORD   0

      002050  047433                RCUR:   .WORD   047433                  ;DIRECT CURSOR ADDRESSING
              000131                .Y      =131
              000131                .RDCUR  =00131
      002052  000131                RDCUR:  .WORD   00131                   ;READ CURSOR POSITION  Y
      002054  000000                        .WORD   0

              000117                .0      =117
      002056  047433                ESC0:   .WORD   047433                  ;ESCAPE 0
              000126                .XMTAL  =000126
      002060  000126                XMTAL:  .WORD   000126                  ;TRANSMIT ALL  V
      002062  000000                        .WORD   0

      002064  047433                        .WORD   047433                  ;ESCAPE 0
              000127                .TCUCH  =127
      002066  000127                TCUCH:  .WORD   127                     ;XMIT CHARACTER AT CURSOR.  W
      002070  000000                        .WORD   0

      002072  047433                        .WORD   047433                  ;ESCAPE 0
              000135                .TXRCK  =135
      002074  000135                TXRCK:  .WORD   135                     ;XMIT RECIEVER CHECKSUM ]
      002076  000000                        .WORD   0

      002100  047433                        .WORD   047433                  ;ESCAPE 0
              000136                .TXTCK  =136
      002102  000136                TXTCK:  .WORD   136                     ;XMIT TRANSMITTER CHECKSUM
      002104  000000                        .WORD   0

      002106  147433                        .WORD   147433                  ;ESCAPE 0
              000140                .RABT   =140
      002110  000140                RABT:   .WORD   140                     ;READ THE ABORT FLAG.  \
      002112  000000                        .WORD   0

                                    ;************************************************************
      002114  177777                        .WORD   -1                      ;END OF TABLE TERMINATOR
                                    ;************************************************************


                                    ;************************************************************
                                            ;PERIPHERAL COMMANDS
                                    ;************************************************************

              000135                .CPYSC  =135            ;COPY SCREEN  ]
              000136                .ENAC   =136            ;ENABLE AUTO COPY MODE  ESC ↑
```

```
              000137                .DISAC  =137            ;DISABLE AUTO COPY MODE  ESC -
              000150                .PSCRN  =000150         ;PRINT THE SCREEN  H/SH



                                    ;************************************************************
                                            ;ESCAPE CODE EQUIVALENCES AND IDENTIFIERS
```

L02

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

                 000033          .ESC     =033                      ;PRIMARY ESCAPE CODE.
                 000120          .P       =120
        002116   050033          ESCP:    .WORD    050033           ;ESCAPE P
                 000124          .TSTER   =124
        002120   000124          TSTER:   .WORD    124              ;TEST TERMINAL(ESC 0 T)
                 002042          ESCYI    =DCRAD                    ;ESCYI EQUALS DCRAD/DCRADI
                 000057          SLSH     =000057                   ;SLASH CODE FOR TERMINAL  IDENT ESC.
                 000106          CKGP     =106                      ;ENABLE REC.TO SUB 137 FROM ALL REC DATA
                 000107          NCKGP    =107                      ;ENABLE NORMAL RECEIVED DATA.
                 000171          CPABRT   =171                      ;COPIER ABORT
                 000172          PRABRT   =172                      ;PRINTER ABORT
                 000170          NABRT    =170                      ;NO ABORT  SX
        002122   000000          IDENT:   .WORD    0                ;VT61 IDENT CODE
                 002056          ESCOI    =ESCO
                 002116          ESCPI    =ESCP
                 002124          ESCZI    =ESCZ
                 055033          .ESCZ    =055033
        002124   055033          ESCZ:    .WORD    055033           ;OCTAL EQUIV. OF ESZ SEQUENCE
                 000122          .RESET   =122
        002126   000122          RESET:   .WORD    122              ;VT61 INITIALIZE R

        002130   000033          ESCN:    .WORD    000033           ;ESCAPE N-FLAG
        002132   020041          R01C00:  .WORD    020041           ;ROW1,COL. 0
        002134   032041          R01C20:  .WORD    032041           ;ROW01,COLUMN 20
        002136   020066          R22C00:  .WORD    020066           ;ROW22,COL.00
        002140   020054          R12C00:  .WORD    020054           ;ROW 12,COLUMN 00
                 020067          .R23C00  =020067
        002142   020067          R23C00:  .WORD    020067           ;ROW23,COL.00
                 025440          .R00C11  =025440
        002144   025440          R00C11:  .WORD    025440           ;ROW,COL.11
                 032040          .R00C20  =032040
        002146   032040          R00C20:  .WORD    032040           ;ROW 0,COLUMN 20
        002150   024040          R00C08:  .WORD    024040           ;ROW 00.COLUMN 8
        002152   020040          CUI.ME:  .WORD    020040           ;OCTAL EQUIV. OF CURSOR HOME.
        002154   067440          R00C80:  .WORD    067440           ;ROW 0, COLUMN 80.
        002156   067067          R23C78:  .WORD    067067           ;ROW 23,COL. 78.
                 000040          .R00     =40                       ;ROW 0
                 000041          .R01     =41                       ;ROW 1
                 000054          .R12     =54                       ;ROW 12
                 000066          .R22     =66                       ;ROW 22
                 000067          .R23     =67                       ;ROW 23
                 000040          .C00     =40                       ;COLUMN 0
                 000043          .C03     =43                       ;COL. 3
                 000050          .C08     =50                       ;COL. 8
                 000053          .C11     =53                       ;COL. 11
                 000064          .C20     =64                       ;COL. 20
                 000065          .C21     =65                       ;COL. 21
                 000110          .C40     =110                      ;COL. 40
```

```
                 000157          .C79     =157                      ;COL. 79


                                 ;*************************************************************************
                                          ;TEMPORARY STORAGE LOCATIONS AND
                                          ;SPECIAL RECEIVE CODE EQUIVALENCES.
                                 ;*************************************************************************
                 000002          SOM      =02                       ;START OF MESSAGE
                 000004          EOM      =04                       ;END OF MESSAGE
                 000023          XOFF     =23                       ;TURN OFF TRANSMISSION
                 000021          XON      =21                       ;TURN ON TRANSMISSION
        002160   000000          CHRD:    .WORD    0                ;STORAGE FOR SINGLE CH. READ
```

M02

```
002162  000000                   SVER1:  .WORD                   ;TEMP. STORAGE R1.
002164  000000                   SVER2:  .WORD                   ;TEMP. STORAGE R2.
002166  000000                   ZERO:   .WORD   0               ;MUST BE LEFT AS ZERO.
002170  003000                   TYP6:   .WORD   3000            ;TYPE 6 OCTAL CHAR-NO ZEROS
002172  000000                   TSTPTR: .WORD   0               ;TEST POINTER IN MANUAL SELECT MODE
002174  000000                   MODE:   .WORD   0               ;BYTE0=TESTING MODE,BYTE1=INTERFACE TYPE
002176  000000                   FTLCNT: .WORD   0               ;COUNT OF INCOMPLETE XMITS.
002200  000012                   ALWCNT: .WORD   10.             ;# OF ALLOWABLE INCOMPLETE XMITS.
002202  000001                   ONE:    .WORD   1
002204  000000                   TOFDD:  .WORD
002206  000000                   BUBCT:  .WORD
002210  000000                   TPREG:  0
002212  000000                   PRESC:  .WORD                   ;PRIMARY ESC COMMAND
002214  000000                   ESSEQ:  .WORD                   ;SEQUENCE ASSEMBLY AREA
002216  000000                   DLAY:   .WORD
002220  000000                   ROSVE:  .WORD                   ;TEMP STORAGE FOR RO ONLY.
002222  000000                   VSTAT:  .WORD   0
002224  000000                   BLKM:   .WORD   0               ;FLAG LOCATION FOR BLOCK MODE XMITS.
002226  000000                   TSTNM:  .WORD   0               ;DISPLAY STORAGE FOR TEST NUMBER.


                                 ;*************************************************************
                                         ;AUTOMATIC SELECTION OF UNITS. TESTS 1 THROUGH 33 WILL BE
                                         ;REPITIVELY EXECUTED FOR ALL UNITS.
                                 ;*************************************************************

002230  005037  002174           AUTO:   CLR     MODE            ;ZERO THE MODE SWITCH
002234  000137  011604                   JMP     SETA            ;;DO VECTOR SETUP
002240  004037  012122           AUTOA:  JSR     RO,TRPVEC       ;GO FIND GOOD DL11S
002244  004037  012242                   JSR     RO,CDEV         ;CHECK DL11S FOUND
002250  004037  012620                   JSR     RO,INITA        ;INSURE VT61S ON DL11
002254  000137  002502                   JMP     MODCK           ;VT61 PRESENT -BEGIN TESTING
002260  000767                           BR      AUTOA           ;NO VT61 FOUND LOOP IN CHECKING

                                 ;*************************************************************

                                         ;MANUAL UNIT AND TEST SELECTION.  UNITS CAN BE
                                         ;SELECTED VIA CONSOLE OR AUTO SELECTION CAN
                                         ;BE UTILIZED.  TESTS ENTERED VIA CONSOLE WILL
                                         ;BE EXECUTED IN THE ORDER ENTERED.

                                 ;*************************************************************

002262  012737  000001  002174   MANS:   MOV     #1,MODE         ;SET MODE TO MANUAL SELECT.
002270  000137  011604                   JMP     SETA            ;GO SET UP CONSTANTS
```

```
002274  104400  023051           MANSA:  TYPE    ,DMANA
002300  004037  012122                   JSR     RO,TRPVEC       ;FIND GOOD DL11'S
002304  012703  001650                   MOV     #INTAB,R3
002310  005002                   BLDADD: CLR     R2

002312  004037  017406           BLDADA: JSR     RO,GTNUM        ;GET A KEYBOARD INPUT
002316  120127  000054                   CMPB    R1,#54          ;CHAR. = COMMA?
002322  001002                           BNE     1$              ;NO
002324  010223                           MOV     R2,(R3)+        ;YES - STORE THIS ADDRESS
002326  000770                           BR      BLDADD          ;AND LOOK FOR ANOTHER ADDRESS
002330  120137  001752           1$:     CMPB    R1,LNFED        ;CHAR. = LINE FEED?
002334  001022                           BNE     3$              ;NO
002336  005702                           TST     R2              ;ANY ENTRIES CREATED?
002340  001411                           BEQ     2$              ;NO USE AUTO SELECTION OF UNITS
002342  010223                           MOV     R2,(R3)+        ;YES STORE LAST ADDRESS,
002344  013723  002166                   MOV     ZERO,(R3)+      ;SET ATERMINATOR IN TABLE
```

```
002350  004037  012242                    JSR     R0,CDEV         ;CHECK DL11 ON VT 61 SELECTED
002354  005737  001610                    TST     DLTBL           ;ANY DL11S GOOD?
002360  001745                            BEQ     MANSA           ;NO-BACK TO SQUARE ONE
002362  000412                            BR      BLCTST          ;YES- GO GET TESTS
002364  004037  012242          2$:       JSR     R0,CDEV         ;CHECK DL11'S
002370  004037  012620                    JSR     R0,INITA        ;VERIFY DL11 HAVE VT61 ATTACHED
002374  000137  002410                    JMP     BLDTST          ;BEGIN TEST SELECTION
002400  000735                            BR      MANSA           ;NO UNIT FOUND-LOOP
002402  004037  017342          3$:       JSR     R0,OCTBIN       ;KEEP BUILDING ADDRESS
002406  000741                            BR      BLDADA

002410  104400  023151          BLDTST:   TYPE    ,DMANB          ;TYPE 2ND PART OF MANUAL MESSAGE
002414  012703  001650                    MOV     #INTAB,R3       ;USE INTAB AS TEST # STORAGE.
002420  005004                            CLR     R4              ;CLEAR TEST COUNTER
002422  005002                  11$:      CLR     R2              ;CLEAR ASSEMBL WORD
002424  004037  017406          10$:      JSR     R0,GTNUM        ;GET A NUMERIC CHAR.
002430  120127  000054                    CMPB    R1,#54          ;CHAR.=COMMA?
002434  001006                            BNE     1$              ;NO
002436  110223                            MOVB    R2,(R3)+        ;YES STORE A TEST #
002440  005204                            INC     R4              ;AND INCREMENT TEST COUNT.
002442  020437  000040                    CMP     R4,32.          ;COUNT =32?
002446  001415                            BEQ     MODCK           ;YES ACCEPT NO MORE ENTRIES.
002450  000764                            BR      11$             ;NO KEEP LOOKING
002452  120137  001752          1$:       CMPB    R1,LNFED        ;CHAR. = LINE FEED?
002456  001006                            BNE     2$              ;NO
002460  110223                            MOVB    R2,(R3)+        ;LOAD THE LAST TEST
002462  105013                            CLRB    (R3)            ;AND INSERT TEST TABLE TERMINATOR
002464  112737  000001  002174            MOVB    #1,MODE         ;SET MODE SWITCH TO MANUAL
002472  000403                            BR      MODCK           ;AND BEGIN TESTING.

002474  004037  017342          2$:       JSR     R0,OCTBIN       ;CONVERT CHAR.
002500  000751                            BR      10$


                                ;*************************************************************
                                          ;THIS ROUTINE LOOKS FOR THE OPERATIONAL MODE REQUESTED AND
                                          ;SELECTS THE NEXT UNIT TO BE TESTED.

                                          ;MODE 0 = ACCEPTANCE TYPE TEST
                                          ;MODE 1 = OPERATOR SELECTION OF UNITS AND SEQUENCE OF TESTS.
                                ;*************************************************************
```

```

002502  012737  001610  001712  MODCK:    MOV     #DLTBL,DLTPT    ;INITIAL SETUP OF ADDRESS
002510  012737  001550  001710            MOV     #VVECT,VECPT    ;AND VECTOR POINTERS.
002516  012701  001730          MODCA:    MOV     #VRCSR,R1       ;LOAD ADDRESS DESTINATION
002522  013702  001712                    MOV     DLTPT,R2        ;LOAD CURRENT ADDRESS POINTER
002526  017703  177156                    MOV     @VECPT,R3       ;LOAD CURRENT VECTOR POINTER
002532  005712                            TST     (R2)            ;ALL UNITS CHECKED?
002534  001013                            BNE     1$              ;NO - CONTINUE
002536  005737  002174                    TST     MODE            ;CHECK MODE
002542  001002                            BNE     10$
002544  000137  002240                    JMP     AUTOA           ;GO RESTART AUTO MODE
002550  105777  177416          10$:      TSTB    @TSTPTR         ;MANUAL LOOP REQUESTED?
002554  100001                            BPL     2$              ;NO
002556  000751                            BR      MODCK           ;YES-RESTART COMPLETE TEST.
002560  000137  002274          2$:       JMP     MANSA           ;GO RESTART MANUAL MODE
002564  004037  013040          1$:       JSR     R0,LDADD        ;NO-LOAD NEXT ADDRESSES
002570  010337  001740                    MOV     R3,VECT         ;STORE VECT. OF UNIT UNDER TEST
002574  012723  013746                    MOV     #INTRC,(R3)+    ;YES - NOW SET UP RECEIVE VECTOR
002600  012723  000340                    MOV     #340,(R3)+      ;AND SET RECEIVER PSW TO 7
002604  012723  014670                    MOV     #INTXM,(R3)+    ;SET UP TRANSMIT VECTOR
002610  012723  000340                    MOV     #340,(R3)+      ;AND SET PSW TO 7.
```

```
002614  005046                                CLR     -(SP)      B03 ;CLEAR THE PSW,LSI11 STYLE.
002616  012746  002624                        MOV     #100$,-(SP)
002622  000002                                RTI
002624  010237  001712                100$:   MOV     R2,C_TPT           ;SAVE ADDRESS POINTER.
002630  012737  030327  014632                MOV     #RCRLB+477,REBUF          ;SET UP END OF BUFFER
002636  012737  031027  015140                MOV     #TCRLB+477,TEBUF
002644  012737  027630  014630                MOV     #RCRLB,RBBUF       ;INITIIALIZE REC.BUFFER.
002652  012737  030330  015136                MOV     #TCRLB,TBBUF       ;INITIALIZE TRANSMIT BUFFER.
002660  004037  016136                        JSR     R0,RESPTR          ;RESET INTERRUPT POINTERS.
002664  005037  002224                        CLR     BLKM               ;CLEAR BLOCK MODE FLAG.
002670  005037  020466                        CLR     XMZER              ;CLEAR ZERO TRANSMIT FLAG
002674  005037  002222                        CLR     VSTAT              ;CLEAR ALL INTERRUPT FLAGS
002700  004037  015326                        JSR     R0,ZFLAG           ;ISSUE ESC Z TO VT61
002704  012637  002122                        MOV     (SP)+,IDENT        ;;POP STACK INTO IDENT
002710  100002                                BPL     11$                ;IF IDENT IS -1,CLEAR IT.
002712  005037  002122                        CLR     IDENT
002716                                11$:
002716  012637  002160                        MOV     (SP)+,CHRD         ;;POP STACK INTO CHRD
002722  001375                                BNE     11$
002724  104400  001167                        TYPE    ,$CRLF
002730  104400  024652                        TYPE    ,DVUNIT            ;ISSUE UNIT UNDER TEST MESSAGE
002734  013746  001730                        MOV     VRCSR,-(SP)        ;;SAVE VRCSR FOR TYPEOUT
                                                                         ;;TYPE THE ADDRESS
002740  104402                                TYPOS                      ;;GO TYPE--OCTAL ASCII
002742     006                                .BYTE   6                  ;;TYPE 6 DIGIT(S)
002743     001                                .BYTE   1                  ;;TYPE LEADING ZEROS
002744  017746  176740                        MOV     @VECPT,-(SP)       ;;SAVE @VECPT FOR TYPEOUT
                                                                         ;;TYPE THE VECTOR
002750  104402                                TYPOS                      ;;GO TYPE--OCTAL ASCII
002752     006                                .BYTE   6                  ;;TYPE 6 DIGIT(S)
002753     000                                .BYTE   0                  ;;SUPPRESS LEADING ZEROS
002754  013746  002122                        MOV     IDENT,-(SP)        ;;SAVE IDENT FOR TYPEOUT
                                                                         ;;TYPE THE IDENT
002760  104402                                TYPOS                      ;;GO TYPE--OCTAL ASCII
002762     006                                .BYTE   6                  ;;TYPE 6 DIGITS
```

```
002763     000                                .BYTE   0                  ;;SUPPRESS LEADING ZEROS
002764  104400  001167                        TYPE    ,$CRLF             ;CARRIAGE RETURN AND LINE FEED
002770  032737  000001  002122                BIT     #BIT00,IDENT       ;UNIT HAVE A COPIER?
002776  001402                                BEQ     20$                ;NO
003000  104400  024771                        TYPE    ,DCOPYR            ;YES-ISSUE COPIER MESSAGE
003004  032737  000002  002122        20$:    BIT     #BIT01,IDENT       ;UNIT HAVE A PRINTER?
003012  001402                                BEQ     21$                ;NO
003014  104400  024743                        TYPE    ,DPRTR             ;YES-ISSUE PRINTER MESSAGE.
003020  062737  000002  001710        21$:    ADD     #2,VECPT           ;LEAVE WITH VECPOINT AT NEXT VECTOR.
003026  005037  002176                        CLR     FTLCNT             ;CLEAR COUNT OF FATAL XMITS.
003032  012737  031032  031030                MOV     #ABBUF,ABUFP       ;RESET THE REC. DATA POINTER
003040  052777  000100  176662                BIS     #RDENA,@VRCSR      ;SET THE REC. INT. ENABLE  FOR TESTS
003046  105737  002174                        TSTB    MODE               ;CHECK TESTING MODE
003052  001403                                BEQ     ASTRT              ;AUTO MODE
003054  012737  001650  002172                MOV     #INTAB,TSTPTR      ;LOAD THE INITIAL TEST NUMBER

                                      ;*****************************************************
                                      ;*****************************************************
                                              ;THIS TEST ISSUES ALL ESCAPE SEQUENCES AND
                                              ;INSURES THE VT61 HAS NOT FAILED DURING AN
                                              ;ESC SEQUENCE BY ISSUING A ESC Z TO FORCE A
                                              ;VT61 RESPONSE.  THE PURPOSE OF THE TEST IS TO ATTEMPT TO
                                              ;INSURE THAT SUBSEQUENT TESTS WILL NOT RESULT IN
                                              ;A "HUNG" UNIT.  DATA IS NOT EVALUATED.
                                      ;*****************************************************
```

```
003062                                  ASTRT:
                                        ;*****************************************************************
003062  000004                          TST1:   SCOPE
003064  012737  000001  001156                  MOV     #1,$TIMES       ;;DO 1 ITERATION
003072  012737  003100  001106                  MOV     #ESTST,$LPADR   ;;SET SCOPE LOOP ADDRESS

003100  012701  001746                  ESTST:  MOV     #BEL,R1         ;POINT TO FIRST COMMAND
003104  042777  000100  176616                  BIC     #RDENA,@VRCSR   ;CLEAR REC. INT. ENABLE
003112  113737  001102  002226                  MOVB    $TSTNM,TSTNM    ;LOAD THE TEST NUMBER.
003120  005037  002212                          CLR     PRESC
003124  005004                                  CLR     R4
003126                                  ZERST:
003126  013746  002166                          MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
003132  012702  002212                          MOV     #PRESC,R2       ;SET UP SEQNENCE ADDRESS
003136  012103                          GCMD:   MOV     (R1)+,R3        ;LOAD THE COMMAND
003140  001405                                  BEQ     1$              ;IF CHAR. ZERO,MUST BE XMIT TERMINATOR
003142  100535                                  BMI     ESTEX           ;TABLE EXPENDED - EXIT TEST.
003144  120327  000004                          CMPB    R3,#4           ;IS COMMAND ACTUALLY A DELIMITER?
003150  103442                                  BLO     DELIM           ;YES, GO UPDATE FUNCTIONS
003152  001471                                  BEQ     SPTN            ;NO, ITS A "10" - SPECIAL CASE.
003154  005704                          1$:     TST     R4              ;SEE IF FLAG INDICATING SEQ.
003156  100472                                  BMI     SEQ4            ;4 IS SET. - YES EXIT
003160  010337  002214                  2$:     MOV     R3,ESSEQ        ;PUSH THE SEQUENCE TO BE TESTED
003164                                  INXMT:
003164  013746  002214                          MOV     ESSEQ,-(SP)     ;;PUSH ESSEQ ON STACK
003170  005704                                  TST     R4              ;DOES THIS SEQUENCE REQUIRE
003172  001402                                  BEQ     3$              ;ADDITIONAL ESC?
003174  013746  002212                          MOV     PRESC,-(SP)     ;;PUSH PRESC ON STACK

003200  004037  013322                  3$:     JSR     R0,TESC         ;GO TRANSMIT THIS SEQUENCE.
```

```

003204  005704                          4$:     TST     R4              ;IN I/O SEQUENCES?
003206  100007                                  BPL     40$             ;NO
003210  012737  000054  017074                  MOV     #44.,DCOUNT     ;YES,SET UP TO DELAY 1+ SEC.
003216  004037  017032                          JSR     R0,DELAY
003222  005777  176504                          TST     @VRBUF          ;CLEAR ANY RECEIVE FLAGS

003226  004037  015326                  40$:    JSR     R0,ZFLAG        ;ISSUE ESC Z SEQUENCE-GET IDENT
003232                                  5$:
003232  012637  002160                          MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
003236  123737  002160  002122                  CMPB    CHRD,IDENT      ;HAVE WE POPPED THE IDENT?
003244  001045                                  BNE     T1ERR           ;NO-ERROR CONDITION
003246                                  POPIT:
003246  012637  002160                          MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
003252  001375                                  BNE     .-4
003254  000724                                  BR      ZERST           ;GET NEXT COMMAND

003256  120327  000001                  DELIM:  CMPB    R3,#1
003262  001407                                  BEQ     1$              ;FIRST DELIMITER - SET ESCN

003264  120327  000002                          CMPB    R3,#BIT01       ;SECOND DELIMITER - SET ESCO
003270  001412                                  BEQ     2$

003272  120327  000003                          CMPB    R3,#3           ;THIRD DELIMITER - SET ESCP
003276  001413                                  BEQ     3$
003300  000716                                  BR      GCMD            ;INVALID CHARACTER - GET ANOTHER

003302  012704  000001                  1$:     MOV     #1,R4           ;SET FIRST DELIMITER FLAG.
003306  013737  002130  002212                  MOV     ESCN,PRESC      ;INSERT ESCN.
003314  000710                                  BR      GCMD

```

```
003316  013737  002056  002212  2$:     MOV     ESCO,@#PRESC    ;INSERT ESCO
003324  000704                          BR      GCMD

003326  013737  002116  002212  3$:     MOV     ESCP,@#PRESC    ;INSERT ESCP
003334  000700                          BR      GCMD

003336  012704  177777          SPTN:   MOV     #-1,R4          ;SET FLAG INDICATING I/O
003342  000675                          BR      GCMD            ;SEQUENCES

003344  005703                  SEQ4:   TST     R3              ;CHECK IF COMMAND = 0
003346  001706                          BEQ     INXMT           ;YES, COMPLETE SEQUENCE ASSEMBLED
003350  110322                          MOVB    R3,(R2)+        ;NO - KEEP ASSEMBLING
003352  000303                          SWAB    R3              ;POSITION HIGH  ORDER BIT
003354  110322                          MOVB    R3,(R2)+        ;AND ASSEMBLE IT
003356  000667                          BR      GCMD            ;GET ANOTHER BYTE

003360  004037  015530          T1ERR:  JSR     R0,CLREG
003364  013737  002212  001124          MOV     PRESC,$GDDAT    ;AND INSERT IN ERROR
003372  000337  001124                  SWAB    $GDDAT          ;REASSEMBLE FAILING SEQUENCES
003376  013737  002214  001126          MOV     ESSEQ,$BDDAT
003404  105737  002215                  TSTB    ESSEQ+1         ;IF UPPER BYTE IS CLEAR DO NOT SWAP
003410  001402                          BEQ     1$
003412  000337  001126                  SWAB    $BDDAT          ;MESSAGE 1
003416  104001                  1$:     ERROR   1               ;ISSUE ERROR MESSAGE
003420  005237  002176                  INC     FTLCNT          ;INCREMENT FATAL XMIT COUNT.
```

```
003424  023737  002176  002200          CMP     FTLCNT,ALWCNT   ;FATAL XMITS EXCEEDED ALLOWED?
003432  103003                          BHIS    FTEX1           ;YES-EXIT.
003434  000704                          BR      POPIT           ;CLEAR THE STACK AND TRY ANOTHER COMMAND
003436                          ESTEX:
003436  012637  002160                  MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
003442  052777  000100  176260  FTEX1:  BIS     #RDENA,@VRCSR   ;SET THE REC. INT. ENABLE FOR TESTS

                                ;*******************************************************************
                                        ;ROUTINE TO INSURE ENTERING MAINTENANCE MODE CAUSES SOM AND
                                        ;EOM TO BE APPENDED TO ALL TRANSMITS FROM VT61 UNDER TEST.
                                        ;MAINTENANCE MODE IS ENTERED , THEN AN ESCAPE Z SEQUNECE
                                        ;IS ISSUED TO THE UNIT AND THE RESULTING TRANSMISSION IS
                                        ;CHECKED OF SOM/EOM.
                                ;*******************************************************************

                                ;*******************************************************************
003450  000004                  TST2:   SCOPE
003452  012737  000005  001156          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
003460  012737  003466  001106          MOV     #CKMNT,$LPADR   ;;SET SCOPE LOOP ADDRESS

003466  004037  015146          CKMNT:  JSR     R0,RESETV       ;RESET THE UNIT AND SETMAINT. MODE.
003472  112777  000002  011442          MOVB    #SOM,@TBUFP     ;ISSUE START OF MESSAGE.
003500  004037  016026                  JSR     R0,XMIT1
003504  113777  002124  011430          MOVB    ESCZ,@TBUFP
003512  004037  016026                  JSR     R0,XMIT1        ;SEND AN IDENT REQUEST.
003516  113777  002125  011416          MOVB    ESCZ+1,@TBUFP
003524  004037  016026                  JSR     R0,XMIT1
003530  112777  000004  011404          MOVB    #EOM,@TBUFP     ;ISSUE END OF MESSAGE.
003536  004037  016026                  JSR     R0,XMIT1
003542  005037  002216                  CLR     DLAY            ;SET UP SOM DELAY OF 100M.S.
003546  032737  040000  002222  1$:     BIT     #RSOM,VSTAT     ;RECEIVED THE START OF MESSAGE?
003554  001003                          BNE     CKEOM           ;YES-GO LOOK FOR EOM.
003556  005337  002216                  DEC     DLAY            ;NO-RUN TIMEOUT DELAY
003562  001371                          BNE     1$              ;AND KEEP LOOKING.

003564  012701  000062          CKEOM:  MOV     #50.,R1         ;SET MAX DELAY FOR 500 M.S.
```

```
        003570  032737  020000  002222  1$:     BIT     #REOM,VSTAT     ;REECIVED END OF MESSAGE?
        003576  001007                          BNE     10$             ;YES-CHECK FOR BOTH RECEIVED.
        003600  012737  000001  017074          MOV     #1,DCOUNT       ;DELAY FOR 10 M..S.
        003606  004037  017032                  JSR     R0,DELAY
        003612  005301                          DEC     R1              ;AND KEEP LOOKING.
        003614  001365                          BNE     1$
        003616  032737  040000  002222  10$:    BIT     #RSOM,VSTAT     ;RECEIVED SOM?
        003624  001404                          BEQ     2$              ;NO ISSUE ERROR
        003626  032737  020000  002222          BIT     #REOM,VSTAT     ;RECEIVED EOM?
        003634  001007                          BNE     EXMNT           ;YES, NO ERRORS-EXIT.
        003636  012737  006001  001124  2$:     MOV     #6001,$GDDAT    ;LOAD ERROR WITH EXPECTED
        003644  013737  002222  001126          MOV     VSTAT,$BDDAT    ;AND ACTUAL STATUS.
        003652  104022                          ERROR   22

        003654  000240                  EXMNT:  NOP
                                        ;*****************************************************************
                                                ;THIS TEST INSURES THAT THE CURSOR WILL RESPOND
                                                ;TO DIRECT CUSROR ADDRESSING. THE UNIT IS RESET AND THE CURSOR
                                                ;POSITION IS VERIFIED TO BE HOME . THE CURSOR IS THEN MOVED
                                                ;TO POSITION ROW 23 COLUMN 80 AND THE POSITION IS AGAIN
```

```
                                                ;VERIFIED. ERRORS ARE REPORTED IF THE POSITIONS ARE INCORRECT.

                                        ;*****************************************************************



                                        ;*****************************************************************
        003656  000004                  TST3:   SCOPE
        003660  012737  000005  001156          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
        003666  012737  003674  001106          MOV     #CURS1,$LPADR   ;;SET SCOPE LOOP ADDRESS

        003674  013701  015136          CURS1:  MOV     TBBUF,R1        ;USE R1 AS XMIT BUFFER POINTER.
        003700  004037  015146                  JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
        003704  013721  002056                  MOV     ESC0I,(R1)+     ;CLFT. RESET, READ CURSOR
        003710  113721  002052                  MOVB    RDCUR,(R1)+     ;POSITION, CURSOR LEFT.
        003714  012737  000003  015144          MOV     #3,XMCNT        ;XMIT 3 BYTES

        003722  004037  015552                  JSR     R0,XMREC        ;XMIT AND RECEIVE.
        003726  000402                          BR      10$             ;NORMAL EXIT.
        003730  104011                          ERROR   11              ;TRANSMISSION CAUSED VT61 TO FAIL/HANG
        003732  000446                          BR      2$              ;EXIT TEST.
        003734  013701  027630          10$:    MOV     RCRLB,R1        ;GET THE CURRENT CURSOR POSITION.
        003740  020137  002152                  CMP     R1,CUHME        ;CURSOR REALLY HOME?
        003744  001405                          BEQ     1$              ;YES EXIT
        003746  104012                          ERROR   12              ;VT61 FAILURE MESSAGE
        003750  013746  002152                  MOV     CUHME,-(SP)     ;;PUSH CUHME ON STACK
        003754  004037  016216                  JSR     R0,CURER        ;GO LOAD AND ISSUE CURSOR ERROR

        003760  013701  015136          1$:     MOV     TBBUF,R1        ;LOAD XMIT BUFFER WITH
        003764  013721  002042                  MOV     DCRAD,(R1)+
        003770  013721  002044                  MOV     R23C79,(R1)+    ;CURSOR TO ROW 23,COL.79
        003774  013721  002056                  MOV     ESC0I,(R1)+     ;READ CURSOR POSITION
        004000  013721  002052                  MOV     RDCUR,(R1)+     ;IT AND CURSOR RIGHT
        004004  012737  000007  015144          MOV     #7,XMCNT        ;XMIT 7 BYTES.
        004012  004037  015552                  JSR     R0,XMREC        ;XMIT AND RECEIVE
        004016  000402                          BR      20$             ;NORMAL EXIT.
        004020  104011                          ERROR   11              ;TRANSMISSION CAUSED VT61 TO FAIL/HANG
        004022  000412                          BR      2$              ;EXIT TEST.
        004024  012701  027630          20$:    MOV     #RCRLB,R1

        004030  023711  002044                  CMP     R23C79,(R1)     ;CHECK CURSOR POSITION TO LOWER RT.
        004034  001405                          BEQ     2$              ;OK, EXIT
```

F03

```
004036  104012                             ERROR   12              ;VT61 FAILURE MESSAGE
004040  013746  002044                     MOV     R23C79,-(SP)    ;;PUSH R23C79 ON STACK
004044  004037  016216                     JSR     R0,CURER        ;LOAD AND ISSUE CURSOR ERROR .
004050  000240                     2$:     NOP
                                   ;************************************************************
                                           ;ROUTINE TO INSURE THE UNIT CAN ENTER LINEAR ADDRESSING
                                           ;MODE.  81 CHARACTERS ARE ISSUED TO THE UNIT UNDER TEST
                                           ;THEN THE CURSOR POSITION IS READ AND MUST BE ROW1,COL.0.
                                   ;************************************************************


                                   ;************************************************************
004052  000004                     TST4:   SCOPE
004054  012737  000005  001156             MOV     #5,$TIMES       ;;DO 5 ITERATIONS
004062  012737  004070  001106             MOV     #CKLIN,$LPADR   ;;SET SCOPE LOOP ADDRESS
```

```
004070  004037  015146             CKLIN:  JSR     R0,RESETV       ;RESET THE UNIT-SET MAINT AND LINEAR MODES
004074  013701  015136                     MOV     TBBUF,R1
004100  012703  000120                     MOV     #80.,R3
004104  004037  017076                     JSR     R0,BLDINC       ;LOAD XMIT BUFFER WITH 80 CHAR AND
004110  013721  002050                     MOV     RCUR,(R1)+
004114  013721  002052                     MOV     RDCUR,(R1)+     ;READ CURSOR POSINION.
004120  012737  000123  015144             MOV     #83.,XMCNT
004126  004037  015552                     JSR     R0,XMREC        ;XMIT THE BUFFER.
004132  000402                             BR      1$
004134  104011                             ERROR   11              ;LAST XMIT CAUSED UNIT TO HANG.
004136  000421                             BR      LINXT           ;EXIT TEST
004140  023777  002132  010462     1$:     CMP     R01C00,@RBBUF   ;CURSOR  AT ROW1,COL. 0?
004146  001415                             BEQ     LINXT           ;YES-EXIT
004150  013737  002056  001124             MOV     ESCO,$GDDAT
004156  000337  001124                     SWAB    $GDDAT
004162  013737  002012  001126             MOV     DRECT,$BDDAT    ;ISSUE ESC SEQUENCE AND CURSOR
004170  104001                             ERROR   1
004172  013746  002132                     MOV     R01C00,-(SP)    ;;PUSH R01C00 ON STACK
004176  004037  016216                     JSR     R0,CURER
004202  000240                     LINXT:  NOP


                                   ;************************************************************
                                           ;TEST TO INSURE OPERATION OF XON/XOFF COMMANDS
                                           ;FROM VT61. XOFF IS FORCED BY TRANSMITTING LINE 23 WHILE SIMUL-
                                           ;TANEOUSLY FILLING THE SILO WITH DATA. AFTER SENSING
                                           ;THE XOFF, THE TEST WAITS FOR THE TRANSMIT TO FINISH AND
                                           ;INSURES XON OCCURS BEFORE THE MAX. TRANSFER TIME HAS ELAPSED.
                                           ;(30 SECONDS)
                                   ;************************************************************

                                   ;************************************************************
004204  000004                     TST5:   SCOPE
004206  012737  000010  001156             MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004214  012737  004222  001106             MOV     #BASC3,$LPADR   ;;SET SCOPE LOOP ADDRESS
004222  013701  015136             BASC3:  MOV     TBBUF,R1        ;R1 = 1ST XMIT BUFFER ADDRESS.
004226  012737  001001  002224             MOV     #1001,BLKM      ;SET UP TO XMIT A SOM -DATA- EOM.
004234  005037  002222                     CLR     VSTAT
004240  004037  015146                     JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
004244  013721  002042                     MOV     DCRAD,(R1)+
004250  013721  002142                     MOV     R23C00,(R1)+    ;CURSOR TO ROW 23, COL.0
004254  013721  002056                     MOV     ESCO,(R1)+
004260  013721  002060                     MOV     XMTAL,(R1)+     ;TRANSMIT THE LINE.
004264  012703  000050                     MOV     #40.,R3
004270  004037  017076                     JSR     R0,BLDINC       ;40 CHAR. OF INCREMENTING CHAR.
004274  012737  000057  015144             MOV     #47.,XMCNT      ;SET UP TO XMIT 47 BYTES
```

```
 004302  052777  000100  175424          BIS     #TENA,@VXCSR    ;TRANSMIT ENABLES
 004310  012703  000050                  MOV     #40.,R3         ;MAXIMUM DELAY EQUAL 400 M.S.
 004314  012737  000001  017074  2$:     MOV     #1,DCOUNT
 004322  004037  017032                  JSR     R0,DELAY        ;DELAY FOR 10 MILLISEC.
 004326  032737  100000  002222          BIT     #RXOFF,VSTAT    ;CHECK FOR XOFF
 004334  001007                          BNE     3$              ;FOUND IT EXIT THIS SECTION.
 004336  005303                          DEC     R3              ;DELAYED 400 M.S.?
 004340  001365                          BNE     2$              ;NO-KEEP LOOKING FOR XOFF.
 004342  104012                          ERROR   12              ;GENERAL VT61 FAILURE MESSAGE
 004344  012746  100000                  MOV     #100000,-(SP)   ;;PUSH #100000 ON STACK
```

```
 004350  004037  015366                  JSR     R0,CKSFT        ;GO REPORT ERROR
 004354                          3$:
 004354  012746  000001                  MOV     #XMDNE,-(SP)    ;;PUSH #XMDNE ON STACK
 004360  012746  000062                  MOV     #50.,-(SP)      ;;PUSH #50. ON STACK
 004364  004037  020470                  JSR     R0,WTBGND
 004370  000411                          BR      EXIT3           ;TIMEOUT-EXIT TEST.
 004372  127727  024432  000021          CMPB    @ABUFP,#XON     ;RECEIVED A XON?
 004400  001405                          BEQ     EXIT3           ;YES-NO ERROR-EXIT

 004402  104012                          ERROR   12              ;GENERAL VT61 FAILURE MESSAGE
 004404  012746  000001                  MOV     #000001,-(SP)   ;;PUSH #000001 ON STACK
 004410  004037  015366                  JSR     R0,CKSFT
 004414  004037  016136          EXIT3:  JSR     R0,RESPTR       ;RESET INTERRUPT POINTERS.

                                 ;**************************************************************
                                         ;ROUTINE TO VERIFY OPERATION OF XOFF AND XON TO THE VT61.
                                         ;A FULL SCREEN TRANSMIT IS INITIATED AND A SERIES OF XOFF AND
                                         ;XON ARE ISSUED TO THE TERMINAL SEQUENTIALLY.
                                         ;ERRORS ARE REPORTED IF XOFF DOES NOT STOP OR XON RESTART
                                         ;THE TRANSMISSION. TEST IS ENDED WHEN EOM IS SENSED.
                                 ;**************************************************************

                                 ;**************************************************************
 004420  000004                  TST6:   SCOPE
 004422  012737  000001  001156          MOV     #1,$TIMES       ;;DO 1 ITERATION
 004430  012737  004436  001106          MOV     #ONOF61,$LPADR  ;;SET SCOPE LOOP ADDRESS

 004436  004037  015146          ONOF61: JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
 004442  042737  077577  002222          BIC     #77577,VSTAT    ;CLEAR THE FLAGS
 004450  013746  002166                  MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
 004454  013746  002060                  MOV     XMTAL,-(SP)     ;;PUSH XMTAL ON STACK
 004460  013746  002056                  MOV     ESCO,-(SP)      ;;PUSH ESCO ON STACK
 004464  004037  013322                  JSR     R0,TESC
 004470  012737  000010  017074  ONOFLP: MOV     #10,DCOUNT      ;ALLOW 100 M.S. FOR OPERATION
 004476  004037  017032                  JSR     R0,DELAY        ;TO BEGIN.
 004502  112777  000023  010432          MOVB    #XOFF,@TBUFP
 004510  004037  016026                  JSR     R0,XMIT1        ;SEND A XOFF TO VT61.
 004514  012704  000036                  MOV     #30.,R4
 004520  013705  031030          OFFLP:  MOV     ABUFP,R5        ;ALLOW 300M.S. FOR XMIT TO CEASE
 004524  012737  000001  017074          MOV     #1,DCOUNT
 004532  004037  017032                  JSR     R0,DELAY
 004536  023705  031030                  CMP     ABUFP,R5
 004542  001406                          BEQ     ONOFA           ;XMIT STOPPFD-GO RESTART IT.
 004544  005304                          DEC     R4
 004546  001364                          BNE     OFFLP           ;COUNTER NO EQUAL 300 MS-LOOP
 004550  013737  002222  001120          MOV     VSTAT,$GDADR    ;UNIT DID NOT RESPOND TO XOFF
 004556  104015                          ERROR   15              ;ISSUE ERROR

 004560  112777  000021  010354  ONOFA:  MOVB    #XON,@TBUFP
 004566  004037  016026                  JSR     R0,XMIT1        ;SEND A XON TO THE VT61.
 004572  012704  000036                  MOV     #30.,R4         ;SET UP FOR 300MS DELAY.
```

```
004576  032737  020000  002222  ONLP:   BIT     #REOM,VSTAT     ;EOM RECEIVED?
004604  001020                          BNE     ONOFXT          ;YES-EXIT
004606  013705  031030                  MOV     ABUFP,R5
004612  012737  000001  017074          MOV     #1,DCOUNT
004620  004037  017032                  JSR     R0,DELAY        ;ALLOW 300 MS FOR XMIT TO RESTART
```

```
004624  023705  031030                  CMP     ABUFP,R5
004630  001317                          BNE     ONOFLP          ;IT RESTARTED-GO STOP IT.
004632  005304                          DEC     R4
004634  001360                          BNE     ONLP            ;NOT YET 300 MS LOOP.
004636  013737  002222  001120          MOV     VSTAT,$GDADR    ;XMIT DIT NOT RESTART-ISSUE
004644  104016                          ERROR   16              ;ERROR AND EXIT
004646  000240                  ONOFXT: NOP

                                ;************************************************************
                                        ;ROUTINE TO TEST VT61 RAM AND THE COMMUNICATION PATHS.
                                        ;THIS ROUTINE ISSUES A SERIES OF PATTERNS(77/100,100/77,
                                        ;52/125,INCREMENTING,AND REV. VIDEO INCREMENTING) TO THE VT61.
                                        ;THE SCREEN IS THEN TRANSMITTED TO THE HOST AND AFTER EACH
                                        ;ITERATION RECEIVED DATA IS CHECKED AND ALL ERRORS(INCLUDING
                                        ;TRANSMISSION) ARE REPORTED.
                                        ;MITTED TO THE HOST COMPUTER AND THE RESULTS ARE CHECKED AND
                                        ;ALL ERRORS(INCLUDING TRANSMISSION) REPORTED.
                                ;************************************************************

                                ;************************************************************
004650  000004                  TST7:   SCOPE
004652  012737  000001  001156          MOV     #1,$TIMES       ;;DO 1 ITERATION
004660  012737  004666  001106          MOV     #MEM1,$LPADR    ;;SET SCOPE LOOP ADDRESS

004666  004037  015146          MEM1:   JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
004672  005005                          CLR     R5              ;CLEAR PATTERN OFFSET.
004674  016504  005402          MEMA:   MOV     MPATT(R5),R4    ;LOAD PATTERN TO BE TRANSMMITTED
004700  004037  016136                  JSR     R0,RESPTR       ;RESET POINTERS
004704  042737  077577  002222          BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
004712  012702  003600                  MOV     #TOTCH,R2       ;LOAD A COUNT OF SCREEN
004716  112777  000002  010216          MOVB    #SOM,@TBUFP     ;ISSUE START OF MESSAGE.
004724  004037  016026                  JSR     R0,XMIT1
004730  005302                  MEMB:   DEC     R2              ;DECREMENT XMIT COUNT
004732  001414                          BEQ     10$             ;COUNT = ZERO?

004734  004037  005350          12$:    JSR     R0,PATGN        ;NO-GENERATE NEXT BYTE TO XMIT.
004740  110477  010176                  MOVB    R4,@TBUFP       ;LOAD THE CHARACTER.
004744  004037  016026                  JSR     R0,XMIT1        ;NO-XMIT ANOTHER BYTE.
004750  023737  002176  002200          CMP     FTLCNT,ALWCNT   ;EXCEEDED FATAL ERROR COUNT?
004756  103764                          BLO     MEMB            ;NO-CHECK IF ANOTHER TRANSMISSION REQUIRED.
004760  000137  005422                  JMP     MEMXT           ;YES-GO ABORT TEST.
004764  112777  000004  010150  10$:    MOVB    #EOM,@TBUFP     ;ISSUE END OF MESSAGE.
004772  004037  016026                  JSR     R0,XMIT1
004776  004037  016136                  JSR     R0,RESPTR       ;RESET INTERRUPT POINTERS.

005002  013701  015136                  MOV     TBBUF,R1        ;LOAD XMIT BUFFER WITH
005006  013721  002130                  MOV     ESCN,(R1)+
005012  013721  001760                  MOV     CHOM,(R1)+      ;CURSOR HOME
005016  013721  002124                  MOV     ESCZ1,(R1)+     ;ESCAPE Z
005022  013721  002056                  MOV     ESCO,(R1)+
005026  013721  002060                  MOV     XMTAL,(R1)+     ;TRANSMIT ALL
005032  013711  001752                  MOV     LNFED,(R1)      ;LINE FEED.
005036  012737  000010  015144          MOV     #8.,XMCNT       ;SET UP TO XMIT 8 BYTES
005044  004037  015552                  JSR     R0,XMREC        ;XMIT,WAIT FOR REC. EOM
005050  000402                          BR      1$              ;NORMAL EXIT
005052  104011                          ERROR   11              ;LAST TRANSMIT CAUSED VT61 TO HANG
```

```
005054  000562                          BR      MEMXT           ;EXIT TEST
005056  042737  077577  002222  1$:     BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
005064  005002                          CLR     R2              ;CLEAR RECEIVE COUNTER.
005066  016504  005402                  MOV     MPATT(R5),R4    ;LOAD PATTERN
005072  012703  030630                  MOV     #TCRLB+300,R3   ;SET UT ERROR STORAGE
005076  013701  014630                  MOV     RBBUF,R1        ;SET UP RECEIVE POINTER
005102  005037  002216          MEMC:   CLR     DLAY            ;SET UP TIME OUT DELAY
005106  013737  014630  014634          MOV     RBBUF,RBUFP     ;RESET RECEIVE POINTER
005114  023701  014634          1$:     CMP     RBUFP,R1        ;RECEIVED A CHAR?
005120  001013                          BNE     MEMD            ;YES-GO CHECK IT.
005122  032737  020000  002222          BIT     #REOM,VSTAT     ;HAVE WE RECEIVED EOM?
005130  001033                          BNE     CKDAT           ;YES, GO CHECK FOR DATA ERRORS
005132  005337  002216                  DEC     DLAY            ;RUN TIME OUT DELAY.
005136  001366                          BNE     1$              ;NOT EXPIRED-KEEP LOOKING.
005140  005237  002176                  INC     FTLCNT          ;TRANSMISSION FAILED-INCR. FATAL COUNT
005144  104011                          ERROR   11
005146  000525                          BR      MEMXT
005150  005202                  MEMD:   INC     R2              ;DATA IN. INCREMENT COUNTER
005152  004037  005350                  JSR     R0,PATGN        ;GET GOOD CHARACTER ,PUT IN R4 AND
005156  122705  000010                  CMPB    #10,R5          ;CHECKING REV. VIDEO DATA?
005162  001002                          BNE     1$              ;NO-DO NOT MODIFY
005164  052704  000200                  BIS     #BIT07,R4       ;YES-FORCE BIT 7.
005170  121104                  1$:     CMPB    (R1),R4         ;COMPARE DATA
005172  001743                          BEQ     MEMC
005174  020227  003600                  CMP     R2,#TOTCH       ;COMPARING LAST CHAR?
005200  001740                          BEQ     MEMC            ;YES-NEVER COUNT AS A ERROR.

005202  020327  030700                  CMP     R3,#TCRLB+350   ;STORED 20 ERRORS?
005206  103335                          BHIS    MEMC            ;YES-STORE NO MORE.
005210  110423                          MOVB    R4,(R3)+        ;STORE THE GOOD DATA.
005212  111123                          MOVB    (R1),(R3)+      ;STORE THE BAD DATA.
005214  010223                          MOV     R2,(R3)+        ;STORE THE RECEIVE COUNT.
005216  000731                          BR      MEMC
005220  022703  030630          CKDAT:  CMP     #TCRLB+300,R3
005224  001415                          BEQ     CKMEM
005226  012701  030630                  MOV     #TCRLB+300,R1   ;LOAD FIRST ERROR ADDRESS.
005232  004037  015530          1$:     JSR     R0,CLREG        ;CLEAR ERROR REGISTERS
005236  112137  001124                  MOVB    (R1)+,$GDDAT    ;LOAD THE GOOD DATA.
005242  112137  001126                  MOVB    (R1)+,$BDDAT    ;LOAD THE ERROR BUFFER
005246  012137  001120                  MOV     (R1)+,$GDADR    ;LOAD RECEIVE COUNT
005252  104004                          ERROR   4               ;ISSUE DATA ERROR MESSAGE.
005254  020103                          CMP     R1,R3           ;ISSUED ALL ERRORS?
005256  103765                          BLO     1$              ;NO-CONTINUE

005260  020227  003600          CKMEM:  CMP     R2,#TOTCH       ;DID WE XFER 1920 TIMES?
005264  001406                          BEQ     1$              ;YES - GO CHECK STATUS
005266  012737  003600  001124          MOV     #TOTCH,$GDDAT   ;NO, PUT GOOD COUNT IN GDDAT
005274  010237  001126                  MOV     R2,$BDDAT       ;AND ACTUAL COUNT IN BDDAT.
005300  104005                          ERROR   5               ;ISSUE COUNT ERROR.

005302                          1$:
005302  012746  060000                  MOV     #60000,-(SP)    ;;PUSH #60000 ON STACK
005306  004037  015366                  JSR     R0,CKSFT
005312  062705  000002                  ADD     #2,R5           ;INCREMENT PATTERN POINTER
005316  005765  005402                  TST     MPATT(R5)       ;TEST NEXT PATTERN
005322  001437                          BEQ     MEMXT           ;ZERO-END OF TEST EXIT.
```

```
005324  100007                          BPL     2$              ;NOT INCRMENTING PATTERN.
005326  122705  000010                  CMPB    #10,R5          ;SET REVERSE VIDEO?
```

```
005332  001004                                  BNE     2$              ;NO.
005334  012703  005416                          MOV     #SETREV,R3      ;YES-ENTER REVERSE VIDEO
005340  004037  016076                          JSR     R0,LDXMIT       ;AND RE-ISSUE INCREMENTING PATTERN.
005344  000137  004674                  2$:     JMP     MEMA            ;NOT ZERO, GO EXERCISE IT.

005350  042704  000200                  PATGN:  BIC     #200,R4         ;CLEAR REV. VIDEO BIT IF SET.
005354  005704                                  TST     R4              ;CHECK R4 FOR PATTERN
005356  100402                                  BMI     1$              ;IF MINUS, DO INCREMENTING.
005360  000304                                  SWAB    R4              ;OTHERWISE SWAP BYTES AND
005362  000200                                  RTS     R0              ;EXIT.
005364  105204                          1$:     INCB    R4              ;ADD ONE TO INCREMENTING
005366  120427  000177                          CMPB    R4,#177         ;HAVE WE EXCEEDED LIMIT
005372  103402                                  BLO     2$              ;NO, EXIT
005374  016504  005402                          MOV     MPATT(R5),R4    ;YES, RESET PATTERN AND
005400  000200                          2$:     RTS     R0              ;EXIT.
                                                .EVEN
        005402                          MPATT   =.
005402  037500                                  .WORU   037500          ;PATTERN 77,100
005404  040077                                  .WORD   040077          ;FATTERN 100,77
005406  025125                                  .WORD   025125          ;PATTERN 52,125
005410  100040                                  .WORD   100040          ;PATTERN INCREMENTING
005412  100040                                  .WORD   100040          ;PATTERN INCREMENTING-REV. VIDEO.
005414  000000                                  .WORD   0               ;PATTERN TABLE TERMINATOR
                                                ;SEQUENCE TO ENTER REVERSE VIDEO.
005416     033     117     112          SETREV: .BYTE   .ESC,.O,.EEMP,O
005421     000
005422  000240                          MEMXT:  NOP


                                        ;************************************************************

                                                ;ROUTINE TO TEST THE ABILITY OF THE VT61 TO CALCULATE
                                                ;AND TRANSMIT CHECKSUMS OF BOTH TRANSMITTED AND RECEIVED
                                                ;DATA.  SUBTEST A TRANSMITS A FULL BUFFER UPDATING A CALCULATED
                                                ;CHECKSUM ON EACH CHARACTER TRANSMITTED.  AN ESCAPE SEQUENCE
                                                ;REQUESTING THE RECEIVER CHECKSUM IS EMBEDDED AT THE END OF
                                                ;XMIT BUFFER AND THE RECEIVED CHECKSUM IS COMPARED TO THE
                                                ;CALCULATED.  SUBTEST B PERFORMS THE SAME TYPE OF CHECK ON
                                                ;THE VT61 TRANSMIT CHECKSUM,UTILIZING THE DATA SENT TO THE VT61
                                                ;IN SUBTEST A,DURING A FULL SCREEN TRANSMIT.

                                        ;************************************************************

                                        ;************************************************************
005424  000004                          TST10:  SCOPE
005426  012737  000003  001156                  MOV     #3,$TIMES       ;;DO 3 ITERATICNS
005434  012737  005442  001106                  MOV     #CKSUMA,$LPADR  ;;SET SCOPE LOOP ADDRESS

005442  004037  015146                  CKSUMA: JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
005446  012737  001001  002224                  MOV     #1001,BLKM      ;SET UP TO XMIT A SOM -DATA- EOM.
005454  004037  016136                          JSR     R0,RESPTR       ;RESET INTERRUPT POINTERS
005460  012703  006070                          MOV     #ITSUMA,R3      ;DIS. RECT. MODE AND CLEAR CHECKSUM
005464  004037  016076                          JSR     R0,LDXMIT
005470  042737  077577  002222                  BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
005476  013701  015136                          MOV     TBBUF,R1        ;LOAD XMIT BUFFER WITH
```

```
005502  012703  000473                          MOV     #315.,R3
005506  004037  017076                          JSR     R0,BLDINC       ;314 INCREMENTING CHAR.
005512  113721  002130                          MOVB    ESCN,(R1)+
005516  113721  001760                          MOVB    CHOM,(R1)+      ;CURSOR HOME
005522  113721  002056                          MOVB    ESCO,(R1)+
005526  113721  002057                          MOVB    ESCO+1,(R1)+
005532  113711  002074                          MOVB    TXRCK,(R1)      ;TRANSMIT RECEIVER CHECKSUM.
```

```
005536  005004                                   CLR     R4                  ;CLEAR CHECKSUM REGISTER
005540  012705  000004                           MOV     #EOM,R5             ;PRELOAD CHECKSUM REG. WITH
005544  004037  017516                           JSR     R0,CALCK            ;EOM FROM PRIOR XMIT.
005550  052737  002000  002222                   BIS     #CKSUM,VSTAT        ;REQJEST CHECKSUM CALCULATIONS.
005556  012737  000500  015144                   MOV     #320.,XMCNT         ;SETUP TO XMIT 320 BYTES
005564  052777  000100  174142                   BIS     #TENA,@VXCSR        ;ENABLE XMIT INTERRUPTS
005572  012746  020000                           MOV     #REOM,-(SP)         ;;PUSH #REOM ON STACK
005576  012746  000012                           MOV     #10.,-(SP)          ;;PUSH #10. ON STACK
005602  004037  020470                           JSR     R0,WTBGND           ;LOOK FOR EOM.
005606  000534                                   BR      CKEXT               ;ERROR EXIT IF NOT FOUND
005610  127704  007014                           CMPB    @RBBUF,R4           ;COMPARE CHECKSUMS
005614  001414                                   BEQ     CKSUMB              ;GOOD GO TO SUBTEST B
005616  004037  015530                           JSR     R0,CLREG            ;BAD COMPARE
005622  110437  001124                           MOVB    R4,$GDDAT           ;LOAD CALCULATED CHECKSUM
005626  117737  006776  001126                   MOVB    @RBBUF,$BDDAT       ;AND VT61 RECEIVER CHECKSUM
005634  104013                                   ERROR   13                  ;ISSUE ERROR
005636  012746  060001                           MOV     #60001,-(SP)        ;;PUSH #60001 ON STACK
005642  004037  015366                           JSR     R0,CKSFT            ;ERROR.

005646  042737  077577  002222   CKSUMB: BIC     #77577,VSTAT        ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
005654  005004                                   CLR     R4                  ;CLEAR CHECKSUM REGISTER
005656  012737  001001  002224                   MOV     #1001,BLKM          ;SET UP TO XMIT A SOM -DATA- EOM.
005664  052737  000100  002222                   BIS     #TXSUM,VSTAT        ;SET UP FOR XMIT CHECKSUM GENERATION.
005672  013701  015136                           MOV     TBBUF,R1            ;LOAD XMIT BUFFER WITH
005676  004037  017564                           JSR     R0,LDBUF            ;LOAD THE BUFFER WITH:
005702     033     117     134                   .BYTE   .ESC,.O,.CLTCK,.ESC,.O,.XMTAL,.ESC,.O,.TXTCK,0
005705     033     117     126
005710     033     117     136
005713     000
005714  012737  000011  015144                   MOV     #9.,XMCNT           ;SET UP TO XMIT 9 BYTES
005722  052777  000100  174004                   BIS     #TENA,@VXCSR        ;ALLOW XMIT INTERRUPTS
005730  012746  000001                           MOV     #XMDNE,-(SP)        ;;PUSH #XMDNE ON STACK
005734  012746  000002                           MOV     #2,-(SP)            ;;PUSH #2 ON STACK
005740  004037  020470                           JSR     R0,WTBGND           ;LOOK FOR XMIT DONE.
005744  000455                                   BR      CKEXT               ;TIME OUT - EXIT TEST.
005746  005037  002216           CKSRC:  CLR     DLAY                ;SET UP TIME OUT DELAY
005752  013702  031030                           MOV     ABUFP,R2            ;RESET THE RECEIVER FLAG
005756  023702  031030           1$:     CMP     ABUFP,R2            ;RECEIVED A CHAR?
005762  001007                                   BNE     2$                  ;YES-GO CHECK IT.
005764  005337  002216                           DEC     DLAY                ;RUN TIME OUT DELAY.
```

```
005770  001372                                  BNE     1$
005772  005237  002176                          INC     FTLCNT          ;TIMED OUT-INCREMENT FATAL XMIT COUNT
005776  104011                                  ERROR   11              ;ISSUE HUNG MESSAGE AND EXIT.
006000  000437                                  BR      CKEXT
006002  122777  000004  023020          2$:     CMPB    #EOM,@ABUFP     ;RECEIVED EOM CHAR?
006010  001356                                  BNE     CKSRC
006012  042737  020000  002222                  BIC     #REOM,VSTAT     ;CLEAR THE EOM FLAG
006020  032737  020000  002222                  BIT     #REOM,VSTAT     ;NOW WAIT FOR LAST EOM FLAG
006026  001774                                  BEQ     .-6             ;FROM XMIT TRANSMITTER CHECKSUM.
```

```
006030  120477  006574                          CMPB    R4,@RBBUF       ;COMPARE 61 TO HOST CHECKSUM.
006034  001421                                  BEQ     CKEXT           ;EQUAL - EXIT TEST
006036  004037  015530                          JSR     R0,CLREG
006042  110437  001124                          MOVB    R4,$GDDAT       ;LOAD THE HOST CALCULATED CHECKSUM
006046  117737  006556  001126                  MOVB    @RBBUF,$BDDAT   ;LOAD THE VT61 TRANSMITTED CHECKSUM
006054  104014                                  ERROR   14              ;ISSUE VT61 XMIT CHECKSUM ERROR
006056  012746  060001                          MOV     #60001,-(SP)    ;;PUSH #60001 ON STACK
006062  004037  015366                          JSR     R0,CKSFT        ;CHECK FOR STATUS ERROR
006066  000404                                  BR      CKEXT

006070     033     117     103          ITSUMA: .BYTE   .ESC,.O,.DRECT,.ESC,.O,.CLRCK,0,0
006073     033     117     133
006076     000     000

006100  004037  016136                  CKEXT:  JSR     R0,RESPTR

                                        ;*************************************************************
                                                ;ROUTINE TO INSURE BASIC CURSOR COMMANDS
                                                ;RESULT IN CORRECT CURSOR MOVEMENT. COMMANDS
                                                ;ARE ISSUED IN THE SEQUENCE:  RESET, CURSOR RIGHT,
                                                ;CURSOR DOWN, CURSOR LEFT, AND CURSOR UP. THE READ
                                                ;CURSOR POSITION COMMAND IS ISSUED AFTER EVERY
                                                ;CURSOR COMMAND AND CURRENT IS COMPARED TO GOOD
                                                ;AND ANY ERRORS REPORTED.
                                        ;*************************************************************


                                        ;*************************************************************
006104  000004                          TST11:  SCOPE
006106  012737  000005  001156                  MOV     #5,$TIMES       ;;DO 5 ITERATIONS
006114  012737  006122  001106                  MOV     #CURS1A,$LPADR  ;;SET SCOPE LOOP ADDRESS

006122  013701  015136                  CURS1A: MOV     TBBUF,R1        ;LOAD XMIT BUFFER ADDRESS
006126  004037  015146                          JSR     R0,RESETV       ;RESET THE UNIT AND WAIT FOR XON.
006132  004037  017564                          JSR     R0,LDBUF        ;LOAD THE BUFFER WITH:
006136     033     103     033                  .BYTE   .ESC,.CRT,.ESC,.O,.RDCUR,.ESC,.CDWN,.ESC
006141     117     131     033
006144     102     033
006146     117     131     033                  .BYTE   .O,.RDCUR,.ESC,.CLFT,.ESC,.O,.RDCUR
006151     104     033     117
006154     131
006155     033     101     033                  .BYTE   .ESC,.CUP,.ESC,.O,.RDCUR,.BEL,0
006160     117     131     007
006163     000
006164  012737  000024  015144                  MOV     #20.,XMCNT      ;SET TO XMIT 20 CHARACTERS
006172  012737  000004  016020                  MOV     #4,RECITT       ;SET RECEIVE ITERATION TO 4
006200  012737  030430  016022                  MOV     #TCRLB+100,WDSTOR;SET UP WORD STORAGE POINTER
006206  004037  015552                          JSR     R0,XMREC        ;XMIT ,AND WAIT FOR REC.DONE
006212  000402                                  BR      11$             ;NORMAL EXIT
006214  104011                                  ERROR   11              ;LAST XMIT CAUSED VT61 TO HANG.
006216  000436                                  BR      CUR1XT          ;EXIT TEST
006220  012701  006304                  11$:    MOV     #GDCURP,R1      ;R1=GOOD POSITION TABLE
```

MQ3

```
006224  012702  030430                   MOV     #TCRLB+10,R2      ;R2=ACTUAL CURSOR POSITION
006230  012703  001762                   MOV     #CRT,R3           ;R3=CURSOR COMMAND TABLE

006234  021112                   12$:    CMP     (R1),(R2)         ;COMPARE GOOD TO ACTUAL
006236  001415                           BEQ     2$                ;OK-GO UPDATE POINTERS.
```

```
006240  113737  002130  001125           MOVB    ESCN,$GDDAT+1
006246  111337  001124                   MOVB    (R3),$GDDAT       ;LOAD COMMAND IN ESC ERROR
006252  005037  001126                   CLR     $BDDAT
006256  104001                           ERROR   1                 ;AND ISSUE IT
006260  011237  027630                   MOV     (R2),RCRLB        ;LOAD BAD CURSOR POSITION
006264  011146                           MOV     (R1),-(SP)        ;;PUSH (R1) ON STACK
006266  004037  016216                   JSR     R0,CURER          ;LOAD AND ISSUE CURSRO ERROR MESSAGE
006272  022122                   2$:     CMP     (R1)+,(R2)+       ;INCREMENT POSITION POINTERS.
006274  022337  001770                   CMP     (R3)+,CUP         ;CHECK FOR COMMAND TERM.(CUP).
006300  001355                           BNE     12$               ;NOT AT TERMINATOR-COMPARE AGAIN
006302  000404                           BR      CUR1XT            ;EXIT TEST


006304  020440                   GDCURP: .WORD   20440             ;ROW 0, COL. 1
006306  020441                           .WORD   20441             ;ROW 1, COL. 1
006310  020041                           .WORD   20041             ;ROW 1, COL. 0
006312  020040                           .WORD   20040             ;ROW 0, COL. 0
006314  000240                   CUR1XT: NOP

                                 ;*****************************************************************
                                         ;ROUTINE TO INSURE THAT READ CHARACTER AT CURSOR
                                         ;FUNCTIONS CORRECTLY. COMMAND SEQUENCE IS:  RESET, A, CURSOR
                                         ;LEFT, READ  CHARACTER AT CURSOR.
                                         ;AN ERROR IS REPORTED IF THE LAST READ IS NOT AN "A".
                                 ;*****************************************************************

                                 ;*****************************************************************
006316  000004                   TST12:  SCOPE
006320  012737  000005  001156           MOV     #5,$TIMES         ;;DO 5 ITERATIONS
006326  012737  006334  001106           MOV     #CURS18,$LPADR    ;;SET SCOPE LOOP ADDRESS

006334  013701  015136           CURS18: MOV     TBBUF,R1
006340  004037  015146                   JSR     R0,RESETV         ;RESET THE UNIT AND WAIT FOR XON.
006344  012721  000101                   MOV     #101,(R1)+        ;A
006350  113721  002130                   MOVB    ESCN,(R1)+
006354  113721  001766                   MOVB    CLFT,(R1)+        ;CURSOR LEFT
006360  013721  002056                   MOV     ESCOI,(R1)+
006364  013711  002066                   MOV     TCUCH,(R1)        ;TRANSMIT CH. AT CURSOR
006370  012737  000006  015144           MOV     #6,XMCNT          ;SET UP TO XMIT 6 CHARACTERS
006376  004037  015552                   JSR     R0,XMREC          ;XMIT STRING AND WAIT FOR EOM.
006402  000402                           BR      10$               ;NORMAL EXIT
006404  104011                           ERROR   11                ;LAST XMIT CAUSED VT61 TO HANG/FAIL
006406  000430                           BR      2$                ;EXIT TEST
006410  127727  006214  000101   10$:    CMPB    @RBBUF,#101       ;CHARACTER READ=A
006416  001424                           BEQ     2$                ;YES-NEXT SUBTEST
006420  013737  002056  001124           MOV     ESCOI,$GDDAT
006426  000337  001124                   SWAB    $GDDAT            ;REASSEMBLE ESC DATA
006432  005037  001126                   CLR     $BDDAT
006436  113737  002066  001127           MOVB    TCUCH,$BDDAT+1    ;LOAD FAILING ESC SEQUENCE
006444  104001                           ERROR   1                 ;AND ISSUE IT
006446  004037  015530                   JSR     R0,CLREG
006452  112737  000101  001124           MOVB    #101,$GDDAT       ;LOAD GOOD CH. AND CH.
006460  117737  006144  001126           MOVB    @RBBUF,$BDDAT
006466  104004                           ERROR   4                 ;READ AND ISSUE THEM.

006470  000240                   2$:     NOP                       ;END OF TEST
```

```
                                          ;*************************************************************
                                                  ;ROUTINE TO VERIFY OPERATION OF REPLACE AND INSERT MODE.
                                                  ;INITIALLY ROW 0 IS WRITTEN TO 80 INCREMENTING CHAR.
                                                  ;ON THE FIRST PASS(REPLACE MODE) A CHARACTER IS REPLACED
                                                  ;AT HOME AND THE CHAR. AT ROWO,COL.0(172) AND ROW1,COLO(NULL)
                                                  ;ARE VERIFIED. ON THE SECOND PASS, INSERT MODE IS ENTERED
                                                  ;AND THE RESULTING INSERTION(AT HOME) IS VERIFIED.ROWO,COLO
                                                  ;SHOULD BE 172 AND ROW1,COLO SHOULD BE 161.
                                          ;*************************************************************

                                          ;*************************************************************
006472  000004                            TST13:  SCOPE
006474  012737  000005  001156                    MOV     #5,$TIMES       ;;DO 5 ITERATIONS
006502  012737  006510  001106                    MOV     #INRPL,$LPADR   ;;SET SCOPE LOOP ADDRESS

006510  004037  015146                    INRPL:  JSR     R0,RESETV       ;RESET THE UNIT
006514  013701  015136                            MOV     TBBUF,R1
006520  005201                                    INC     R1              ;LEAVE ROOM IN BUFFER FOR SOM.
006522  012703  000120                            MOV     #80.,R3         ;CREATE A LINE OF 80 INCREMENTING
006526  004037  017076                            JSR     R0,BLDINC       ;CHAR. ON THE SCREEN.
006532  105011                                    CLRB    (R1)
006534  013703  015136                            MOV     TBBUF,R3
006540  004037  016076                            JSR     R0,LDXMIT
006544  005005                                    CLR     R5              ;USE R5 AS TEST INDEXER.
006546  012737  000002  016020            INAG:   MOV     #2,RECITT       ;SET UP TO RECEIVE 2 CHAR.
006554  012737  030530  016024                    MOV     #TCRLB+200,BYSTOR       ;SET UP STORAGE AREA.
006562  013701  015136                            MOV     TBBUF,R1
006566  004037  017564                            JSR     R0,LDBUF        ;LOAD THE BUFFER WITH:
006572     033     110     172                    .BYTE   .ESC,.CHOM,172, ESC,.CHOM,.ESC,.0,.TCUCH
006575     033     110     033
006600     117     127
006602     033     102     033                    .BYTE   .ESC,.CDWN,.ESC,.0,.TCUCH,0
006605     117     127     000
006610  012737  000015  015144                    MOV     #13.,XMCNT      ;SET UP TO XMIT 13 CAHR.
006616  004037  015552                            JSR     R0,XMREC
006622  000402                                    BR      1$              ;NORMAL EXIT
006624  104011                                    ERROR   11              ;LAST XMIT CAUSED UNIT TO HANG.
006626  000433                                    BR      INRXT           ;EXIT TEST.
006630  026537  006706  030530            1$:     CMP     TDATA(R5),TCRLB+200     ;COMPARE GOOD TO REC.DATA.
006636  001407                                    BEQ     2$              ;GOOD-LOOP OR EXIT.
006640  016537  006700  001126                    MOV     TFUNCT(R5),$BDDAT
006646  013737  002116  001124                    MOV     ESCP,$GDDAT     ;LOAD ESCAPE SEQ. ERROR.
006654  104001                                    ERROR   1
006656  005725                            2$:     TST     (R5)+           ;INCREMENT INDEXER.
006660  020527  000004                            CMP     R5,#4           ;THRU WITH TEST?
006664  001414                                    BEQ     INRXT           ;YES-EXIT.
006666  012703  006712                            MOV     #ENSRT,R3       ;NO-SECOND PASS- ENTER
006672  004037  016076                            JSR     R0,LDXMIT       ;INSERT MODE AND DO AGAIN.
006676  000723                                    BR      INAG

006700  000151  000111  177777            TFUNCT: .WORD   .ERPL,.EINST,-1
006706     172     000     172            TDATA:  .BYTE   172,0,172,160
006711     160
006712     033     120     111            ENSRT:  .BYTE   .ESC,.P,.EINST,0
006715     000
006716  000240                            INRXT:  NOP
```

```

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

```
                                                ;ROUTINE TO INSURE VT61 WILL SCROLL IF A LINE FEED
                                                ;IS ISSUED FORM ROW 23 OR A CURSOR RIGHT FROM ROW23,COL. 79.
                                                ;IN SUBTEST A, ROW 0 IS INTIIALLY WRITTEN TO A 0 AND ROW 1
                                                ;A 1. AFTER COMPLETION OF A LINE FEED(AND RESULTING SCROLL;
                                                ;ROW 00,COL.00 IS EXPECTED TO CONTAIN A 1.
                                                ;IN SUBTEST B, THE CURSOR IS PLACED AT ROW23,COL.79
                                                ;AND A DATA CHARACTER "A" IS ENTERED. THE CURSOR
                                                ;POSITION IS THEN READ AND SHOULD BE ROW23,COL.00. THE
                                                ;CHARACTER AT HOME IS VERIFIED TO BE A NULL.
                                        ;***************************************************************


                                        ;***************************************************************
006720  000004                          TST14:  SCOPE
006722  012737  000005  001156                  MOV     #5,$TIMES       ;;DO 5 ITERATIONS
006730  012737  006736  001106                  MOV     #CKSCRA,$LPADR  ;;SET SCOPE LOOP ADDRESS

006736  004037  015146                  CKSCRA: JSR     R0,RESETV       ;RESET THE UNIT.
006742  013701  015136                          MOV     TBBUF,R1
006746  004037  017564                          JSR     R0,LDBUF        ;LOAD THE XMIT BUFFER WITH:
006752     060     033     102                  .BYTE   60,.ESC,.CDWN,.ESC,.CLFT,61,.ESC,.Y,.R23,.C00
006755     033     104     061
006760     033     131     067
006763     040
006764     012     033     110                  .BYTE   .LNFED,.ESC,.CHOM,.ESC,.O,.TCUCH,.BEL,0
006767     033     117     127
006772     007     000
006774  012737  000020  015144                  MOV     #16.,XMCNT      ;SET UP TO XMIT 16 BYTES.
007002  004037  015552                          JSR     R0,XMREC
007006  000402                                  BR      1$              ;NORMAL EXIT
007010  104011                                  ERROR   11              ;LAST XMIT CAUSED UNIT TO HANG.
007012  000452                                  BR      GDSCRL          ;EXIT TEST.
007014  127727  005610  000061          1$:     CMPB    @RBBUF,#61      ;CHARACTER AT HOME A 1?
007022  001401                                  BEQ     CKSCRB          ;YES-NEXT TEST
007024  104023                                  ERROR   23              ;NO-ISSUE NO SCROLL ERROR.
007026  012737  000002  016020          CKSCRB: MOV     #2,RECITT       ;SET UP FOR TWO REC. LOOPS.
007034  012737  030530  016022                  MOV     #TCRLB+200,WDSTOR       ;SET UP CURSOR POSITION STROAGE.
007042  013701  015136                          MOV     TBBUF,R1
007046  004037  017564                          JSR     R0,LDBUF        ;LOAD XMIT BUFFER WITH:
007052     033     131     067                  .BYTE   .ESC,.Y,.R23,.C79,101,.ESC,.O,.RDCUR
007055     157     101     033
007060     117     131
007062     033     110     033                  .BYTE   .ESC,.CHOM,.ESC,.O,.TCUCH,0
007065     117     127     000
007070  012737  000015  015144                  MOV     #13.,XMCNT      ;SET UP TO XMIT 13 BYTES.
007076  004037  015552                          JSR     R0,XMREC        ;XMIT AND WAIT FOR RECEIVED DONE.
007102  000402                                  BR      1$
007104  104011                                  ERROR   11              ;LAST XMIT CAUSED VT61 TO HANG.
007106  000414                                  BR      GDSCRL          ;ERROR EXIT
007110  127737  005514  002166          1$:     CMPB    @RBBUF,ZERO     ;NULL RECEIVED?
007116  001410                                  BEQ     GDSCRL          ;YES-EXIT TEST
007120  104023                                  ERROR   23              ;NO-ISSUE NO SCROLL ERROR.
007122  013777  030530  005500                  MOV     TCRLB+200,@RBBUF        ;LOAD RECEIVED CURSOR POSITION.
007130  013746  002142                          MOV     R23C00,-(SP)    ;;PUSH R23C00 ON STACK
007134  004037  016216                          JSR     R0,CURER        ;GO ISSUE CURSOR ERROR.
```

```
007140  000240                          GDSCRL: NOP



                                        ;***************************************************************
                                                ;THIS TEST INSURES THAT THE VT61 CURSOR CAN BE
                                                ;POSITIONED TO VERY POSSIBLE ROW/COLUMN POSITON
                                                ;ON THE SCREEN.  THIS IS TESTED BY FILLING THE
                                                ;COMPLETE SCREEN WITH A CHARACTER(A) AND THEN
                                                ;POSITONING THE CURSOR (VIA DCA) TO EVERY POSITION
                                                ;AND THE "A" AT THAT POSITION IS REPLACED WITH A SPACE.
                                                ;THE SCREEN IS THEN READ TO VERIFY THAT ONLY SPACES
                                                ;EXIST ON THE SCREEN.  ALL POSITIONS CONTAINING
                                                ;NON-SPACES ARE REPORTED.

                                        ;*******************************************************************

                                        ;*******************************************************************
007142  000004                          TST15:  SCOPE
007144  012737  000001  001156                  MOV     #1,$TIMES               ;;DO 1 ITERATION
007152  012737  007160  001106                  MOV     #CURS2,$LPADR           ;;SET SCOPE LOOP ADDRESS

007160  042737  077577  002222  CURS2:  BIC     #77577,VSTAT            ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
007166  004037  015146                  JSR     R0,RESETV               ;RESET THE UNIT AND WAIT FOR XON.
007172  012702  003600                  MOV     #TOTCH,R2               ;LOAD A COUNT OF SCREEN(1920).
007176  112777  000002  005736          MOVB    #SOM,@TBUFP             ;ISSUE START OF MESSAGE.
007204  004037  016026                  JSR     R0,XMIT1
007210  005302                  1$:     DEC     R2                      ;DECREMENT XMIT COUNT
007212  001413                          BEQ     10$                     ;COUNT = ZERO?

007214  112777  000101  005720          MOVB    #101,@TBUFP             ;LOAD THE CHARACTER(A).
007222  004037  016026                  JSR     R0,XMIT1                ;NO-XMIT ANOTHER BYTE.
007226  023737  002176  002200          CMP     FTLCNT,ALWCNT           ;EXCEEDED FATAL ERROR COUNT?
007234  103765                          BLO     1$                      ;NO-CHECK IF DONE NOW
007236  000137  007640                  JMP     C2XT                    ;YES-ABORT TESTING THIS UNIT.
007242  112777  000004  005672  10$:    MOVB    #EOM,@TBUFP             ;ISSUE END OF MESSAGE.
007250  004037  016026                  JSR     R0,XMIT1
007254  004037  016136                  JSR     R0,RESPTR               ;RESET INTERRUPT POINTERS.
007260  013737  002156  016420          MOV     R23C78,LNRW             ;SET UP 1ST ADDRESS
007266  013701  015136                  MOV     TBBUF,R1                ;LOAD XMIT BUFFER WITH
007272  013721  002042                  MOV     DCRAD,(R1)+
007276  010102                          MOV     R1,R2                   ;R2 POINTS TO CURSOR ADD. IN BUFFER
007300  013721  002156                  MOV     R23C78,(R1)+            ;CURSOR TO LOWER RIGHT -1.
007304  112721  000040                  MOVB    #40,(R1)+               ;SPACE
007310  012737  000005  015144  2$:     MOV     #5,XMCNT                ;SET UP TO XMIT 5 CHARACTERS
007316  042737  077577  002222          BIC     #77577,VSTAT            ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
007324  052777  000100  172402          BIS     #TENA,@VXCSR            ;XMIT INTERRUPTS.
007332  012746  000001                  MOV     #XMDNE,-(SP)            ;;PUSH #XMDNE ON STACK
007336  012746  000002                  MOV     #2,-(SP)                ;;PUSH #2 ON STACK
007342  004037  020470                  JSR     R0,WTBGND               ;LOOK FOR  XMIT DONE
007346  000534                          BR      C2XT                    ;NOT FOUND-ERROR EXIT
007350  021237  002152                  CMP     (R2),CUHME              ;DELETED TO HOME?
007354  001405                          BEQ     3$                      ;YES
007356  004037  016314                  JSR     R0,CMPOS                ;NO-GET NEXT POSITION TO BE DELETED
007362  013712  016420                  MOV     LNRW,(R2)               ;LOAD IT IN XMIT BUFFER
007366  000750                          BR      2$                      ;AND DELETE IT.
007370  004037  016136          3$:     JSR     R0,RESPTR               ;RESET INTERRUPT POINTERS
```

```
007374  013737  002152  016420          MOV     CUHME,LNRW              ;LOAD INITIAL CHECK POSITION(HOME)
007402  012737  001001  002224          MOV     #1001,BLKM              ;SET UP TO XMIT A SOM -DATA- EOM.
007410  013701  015136                  MOV     TBBUF,R1                ;LOAD XMIT BUFFER WITH
007414  010102                          MOV     R1,R2                   ;STORE ERRORS IN XMIT BUFFER
007416  042737  077577  002222          BIC     #77577,VSTAT            ;CLEAR ALL FLAGS BUT XOFF AND XMKIL
007424  013721  002130                  MOV     ESCN,(R1)+
007430  013721  001760                  MOV     CHOM,(R1)+              ;CURSOR HOME
007434  013721  002056                  MOV     ESCO,(R1)+
007440  013721  002060                  MOV     XMTAL,(R1)+             ;TRANSMIT ALL
007444  012737  000005  015144          MOV     #5,XMCNT
007452  052777  000100  172254          BIS     #TENA,@VXCSR            ;SET XMIT ENABLE
007460  012746  000001                  MOV     #XMDNE,-(SP)            ;;PUSH #XMDNE ON STACK
```

```
007464  012746  000003                  MOV     #3,-(SP)        ;;PUSH #3 ON STACK
007470  004037  020470                  JSR     R0,WTBGND       ;LOOK FOR SOM OR XMIT DONE
007474  000461                          BR      C2XT            ;NOT FOUND-ERROR EXIT
007476  013701  014634          4$:     MOV     RBUFP,R1        ;SET UP RECEIVE FLAG
007502  005037  002216                  CLR     DLAY            ;SET UP TIME OUT DELAY
007506  020137  014634          40$:    CMP     R1,RBUFP        ;CHARACTER RECEIVED?
007512  103411                          BLO     41$             ;YES-GO CHECK IT.
007514  032737  020000  002222          BIT     #REOM,VSTAT     ;LOOK FOR END OF MESSAGE
007522  001025                          BNE     C2CK            ;FOUND IT, EXIT TEST
007524  005337  002216                  DEC     DLAY            ;RUN TIME OUT DELAY.
007530  001366                          BNE     40$             ;AND LOOK FOR RECEIVED CH.
007532  104011                          ERROR   11              ;LAST XMIT CAUSED VT61 TO HANG.
007534  000420                          BR      C2CK            ;GO SEE IF ANY ERRORS STORED.
007536  013737  014630  014634  41$:    MOV     RBBJF,RBUFP     ;RESET RECEIVE POINTER
007544  127727  005060  000040          CMPB    @RBBUF,#40      ;CHAR EQUAL A SPACE?
007552  001003                          BNE     6$              ;NOT A SPACE-MUST BE ERROR-STORE IT
007554  004037  016356          5$:     JSR     R0,CPPOS        ;UPDATE CURSOR POSITION
007560  000746                          BR      4$
007562  022702  030354          6$:     CMP     #TCRLB+20.,R2   ;STORED 10 ERRORS?
007566  101772                          BLOS    5$              ;YES-IGNORE ANY FURTHER ERRORS.
007570  013722  016420                  MOV     LNRW,(R2)+      ;STORE FAILING CURSOR POSITION
007574  000767                          BR      5$

007576  020237  015136          C2CK:   CMP     R2,TBBUF        ;ANY EPRORS STORED?
007602  001416                          BEQ     C2XT            ;NO EXIT TEST
007604  013701  015136                  MOV     TBBUF,R1        ;USE R1 AS ERROR POINTER
007610  021137  002044          1$:     CMP     (R1),R23C79     ;CURSOR TO LOWER RIGHT?
007614  001411                          BEQ     C2XT            ;YES-NOT AN ERROR.
007616  104012                          ERROR   12              ;NO-ISSUE ERROR MESSAGES
007620  012746  020040                  MOV     #20040,-(SP)    ;;PUSH #20040 ON STACK
007624  012177  005000                  MOV     (R1)+,@RBBUF    ;LOAD FAILING POS.
007630  004037  016216                  JSR     R0,CURER        ;ISSUE CURSOR ERROR
007634  020102                          CMP     R1,R2           ;DONE WITH ERRORS?
007636  103764                          BLO     1$              ;NO, DUMP ANOTHER.
007640  000240                  C2XT:   NOP                     ;EXIT TEST


                                ;*************************************************************
                                        ;ROUTINE TO INSURE PROPER OPERATION OF CARRIAGE RETURN
                                        ;AND LINE FEED DURING NORMAL MODE. INITIALLY THE CURSOR IS
                                        ;SET(VIA D.C.A.) TO ROW0,COL 20 AND A LINE FEEL IS ISSUED
                                        ;THE CURSOR POSITION IS THEN READ AND MUST BE ROW1,COL20.
                                        ;A CARRIAGE RETURN IS THEN ISSUED AND CURSOR POSITION VERIFIED
                                        ;TO BE ROW1,COL0.

```

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

                                ;*************************************************************
007642  000004                  TST16:  SCOPE
007644  012737  000005  001156          MOV     #5,$TIMES       ;;DO 5 ITERATIONS
007652  012737  007660  001106          MOV     #NWLN,$LPADR    ;;SET SCOPE LOOP ADDRESS

007660  004037  015146          NWLN:   JSR     R0,RESETV       ;RESET THE UNIT AND ENTER MAINT.MODE
007664  013701  015136                  MOV     TBBUF,R1
007670  004037  017564                  JSR     R0,LDBUF        ;LOAD XMIT BUFFER WITH-
007674     033     131     040          .BYTE   .ESC,.Y,.R00,.C20
007677     064
007700     012     033     117          .BYTE   .LNFED,.ESC,.O,.RDCUR,.BEL,0
007703     131     007     000
007706  012737  000011  015144          MOV     #9.,XMCNT       ;SETUP TO XMIT 9 CHARACTERS
007714  004037  015552                  JSR     R0,XMREC        ;GO DO IT
007720  000402                          BR      30$             ;NORMAL EXIT.
```

```
007722  104011                              ERROR   11              ;TRANSMISSION CAUSED VT61 TO FAIL/HANG
007724  000454                              BR      4$              ;EXIT TEST
007726  023777  002134  004674      30$:    CMP     R01C20,@RBBUF   ;CHECK CURSOR POS. S/B ROW 1, COL 20.
007734  001412                              BEQ     3$
007736  005037  001124                      CLR     $GDDAT
007742  013737  001752  001126              MOV     LNFED,$BDDAT
007750  104001                              ERROR   1               ;ISSUE IT
007752  013746  002134                      MOV     R01C20,-(SP)    ;;PUSH R01C20 ON STACK
007756  004037  016216                      JSR     R0,CURER        ;SETUP AND ISSUE CURSOR ERROR
007762  013701  015136              3$:     MOV     TBBUF,R1
007766  013721  001750                      MOV     CARRT,(R1)+     ;LOAD XMIT BUFFER WITH
007772  013721  002056                      MOV     ESCOI,(R1)+     ;CARRIAGE RETURN, READ CURSOR
007776  013721  002052                      MOV     RDCUR,(R1)+     ;POSITION
010002  012737  000004  015144              MOV     #4,XMCNT        ;SET UP TO TRANSMIT 4 CHARACTERS
010010  004037  015552                      JSR     R0,XMREC        ;GO DO IT
010014  000402                              BR      40$             ;NORMAL EXIT.
010016  104011                              ERROR   11              ;TRANSMISSION CAUSED VT61 TO FAIL/HANG
010020  000416                              BR      4$              ;EXIT TEST
010022  023777  002132  004600      40$:    CMP     R01C00,@RBBUF   ;CHECK CURSOR POS. S/B ROW1, COL 0.
010030  001412                              BEQ     4$              ;EXIT TEST IF GOOD.
010032  005037  001124                      CLR     $GDDAT
010036  013737  001750  001126              MOV     CARRT,$BDDAT
010044  104001                              ERROR   1               ;ISSUE IT
010046  013746  002132                      MOV     R01C00,-(SP)    ;;PUSH R01C00 ON STACK
010052  004037  016216                      JSR     R0,CURER        ;SET UP AND ISSUE CURSOR ERROR
010056  000240  .                   4$:     NOP



                                    ;*****************************************************************

                                            ;ROUTINE TO VERIFY PROPER OPERATION OF ERASE TO END-OF-
                                            ;SCREEN. SCREEN IS WRITTEN TO 1920 INCREMENTING CHAR.
                                            ;ERASE TO END OF SCREEN IS THEN ISSUED AND THE
                                            ;ENTIRE SCREEN IS READ VERIFYING THAT IT IS ALL NULLS.
                                    ;*****************************************************************

                                    ;*****************************************************************
010060  000004                      TST17:  SCOPE
010062  012737  000003  001156              MOV     #3,$TIMES       ;;DO 3 ITERATIONS
```

```
010070  012737  010076  001106              MOV     #ERSE,$LPADR    ;;SET SCOPE LOOP ADDRESS


010076  004037  015146              ERSE:   JSR     R0,RESETV       ;RESET THE UNIT -SET MAINT. MODE.
010102  005077  004522                      CLR     @RBBUF          ;CLEAR THE CHECK LOCATION.
010106  004037  017124                      JSR     R0,DATSC        ;FILL THE SCREEN.
010112  013701  015136                      MOV     TBBUF,R1
010116  004037  017564                      JSR     R0,LDBUF        ;LOAD XMIT BUFFER WITH:
010122     033     110     033              .BYTE   .ESC,.CHOM,.ESC,.EOS,.ESC,.0,.XMTAL,0
010125     112     033     117
010130     126     000
010132  113737  002130  001125              MOVB    ESCN,$GDDAT+1
010140  113737  001772  001124              MOVB    EOS,$GDDAT      ;LOAD ERROR WITH ERASE TO EOS
010146  005037  001126                      CLR     $BDDAT
010152  005077  004452                      CLR     @RBBUF
010156  012737  000007  015144              MOV     #7,XMCNT        ;SET UP TO XMIT 7 BYTES
010164  004037  015552                      JSR     R0,XMREC        ;XMIT AND WAIT FOR REC. DONE
010170  000402                              BR      5$
010172  104011                              ERROR   11              ;ESC ERROR
010174  000413                              BR      ERSXT           ;EXIT TEST
010176  127737  004426  002166      5$:     CMPB    @RBBUF,ZERO     ;VT61 XMITTED SOM/EOM ONLY?
010204  001407                              BEQ     ERSXT           ;YES-EXIT TEST.
```

F04

```
010206  104001                                ERROR   1                 ;NO-ERASE TO END OF SCREEN
010210  004037  015530                        JSR     R0,CLREG          ;GO CLEAR ERROR STORAGE
010214  117737  004410  001126                MOVB    @RBBUF,$BDDAT
010222  104004                                ERROR   4                 ;ISSUE DATA ERROR
010224  000240                        ERSXT:  NOP



                                      ;*******************************************************************
                                              ;ROUTINE TO SET UP END OF PASS INDICATION.
                                              ;SELF TEST(ESC P T) IS ISSUED TO THE UNIT UNDER TEST
                                              ;AND AN ERROR IS ISSUED IF THE UNIT CANNOT RESPOND AFTER
                                              ;SELF TEST IS COMPLETE. IF SELF TEST IS SUCCESSFUL THE
                                              ;SCREEN IS WRITTEN TO 23 LINES OF INCREMENTING CHARACTERS
                                              ;AND 23 LINES OF INCREMENTING CHAR. IN REVERSE VIDEO.
                                              ;THE IDENT IS THEN CHECKED AND IF A COPIER IS PRESENT A
                                              ;COPY SCREEN  COMMAND IS ISSUED(NOTE: THIS COMMAND WILL CAUSE
                                              ;THE UNIT TO BE "BUSY" AND NOT RESPOND TO ANY FURTHER COMMANDS
                                              ;UNTIL THE SCREEN HAS BEEN COMPLETELY COPIED.)
                                      ;*******************************************************************

                                      ;*******************************************************************
010226  000004                        TST20:  SCOPE
010230  012737  000001  001156                MOV     #1,$TIMES         ;;DO 1 ITERATION
010236  012737  010244  001106                MOV     #LSTST,$LPADR     ;;SET SCOPE LOOP ADDRESS

010244                                LSTST:
010244  013746  002166                        MOV     ZERO,-(SP)        ;;PUSH ZERO ON STACK
010250  013746  002120                        MOV     TSTER,-(SP)       ;;PUSH TSTER ON STACK
010254  013746  002056                        MOV     ESCO,-(SP)        ;;PUSH ESCO ON STACK
010260  004037  013322                        JSR     R0,TESC           ;TRANSMIT IT.
010264  004037  015256                        JSR     R0,GETON          ;GO LOOK FOR A XON.
010270  000407                                BR      1$                ;VT61RESPONDED-NOT HUNG
010272  013737  001730  001124                MOV     VRCSR,$GDDAT      ;LOAD THE ADDRESS
```

```
010300  013737  001740  001126                MOV     VECT,$BDDAT       ;LOAD THE VECTOR
010306  104010                                ERROR   10                ;REPORT SELF TEST FAILURE
010310  004037  015146                1$:     JSR     R0,RESETV         ;RESET AND SET MAINT. MODE.
010314  005037  002206                        CLR     BUBCT             ;SET UP HALF-SCREEN FLAG.
010320  042737  077577  002222        2$:     BIC     #77577,VSTAT      ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
010326  012737  001001  002224                MOV     #1001,BLKM        ;SET UP TO XMIT A SOM -DATA- EOM.
010334  013701  015136                        MOV     TBBUF,R1          ;SET UP BEG. OF XMIT BUFFER
010340  012703  000500                        MOV     #320.,R3          ;FILL BUFFER WITH INCREMENTING CHAR.
010344  004037  017076                        JSR     R0,BLDINC
010350  012737  001700  015144                MOV     #960.,XMCNT       ;SEND 12 LINE TO VT61
010356  052777  000100  171350                BIS     #TENA,@VXCSR      ;ENABLE XMIT INTERRUPTS
010364  012746  000001                        MOV     #XMDNE,-(SP)      ;;PUSH #XMDNE ON STACK
010370  012746  000012                        MOV     #10.,-(SP)        ;;PUSH #10. ON STACK
010374  004037  020470                        JSR     R0,WTBGND         ;LOOK FOR XMDNE.
010400  000430                                BR      ENDSEL            ;NOT FOUND-EXIT.
010402  005737  002206                        TST     BUBCT             ;DONE WITH SCREEN?
010406  001007                                BNE     3$                ;YES-EXIT
010410  012703  005416                        MOV     #SETREV,R3        ;NO-ISSUE ENTER REVERSE VIDEO
010414  004037  016076                        JSR     R0,LDXMIT         ;ESCAPE SEQUENCE.
010420  005237  002206                        INC     BUBCT             ;INCREMENT SCREEN HALF FLAG.
010424  000735                                BR      2$                ;AND ISSUE SECOND HALF IN REV. VIDEO.
010426  032737  000001  002122        3$:     BIT     #BIT00,IDENT      ;IDENT = COPIER?
010434  001412                                BEQ     ENDSEL            ;NO
010436  013746  002166                        MOV     ZERO,-(SP)        ;;PUSH ZERO ON STACK
010442  012746  000135                        MOV     #.CPYSC,-(SP)     ;;PUSH #.CPYSC ON STACK
010446  013746  002120                        MOV     ESCN,-(SP)        ;;PUSH ESCN ON STACK
010452  004037  013322                        JSR     R0,TESC
```

G04

```
010456  004037  017572                  JSR     R0,CKABRT       ;CHECK FOR A PERIPHERAL ABORT.
010462  105737  002174          ENDSEL: TSTB    MODE            ;IF IN MAN MODE DO NOT ENTER EOP.
010466  001402                          BEQ     ENDPS
010470  000137  003062                  JMP     ASTRT
010474  042777  000100  171226  ENDPS:  BIC     #RDENA,@VRCSR   ;CLEAR REC.INT. BEFORE NEXT UNIT SELECT.
                                ;*******************************************************************

                                .SBTTL  END OF PASS ROUTINE

                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO MODCA

010502                          $EOP:
010502  000004                          SCOPE
010504  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
010510  005037  001156                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
010514  005237  001100                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
010520  042737  100000  001100          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
010526  005327                          DEC     (PC)+           ;;LOOP?
010530  000001                  $EOPCT: .WORD   1
010532  003022                          BGT     $DOAGN          ;;YES
010534  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
010536  000001                  $ENDCT: .WORD   1
010540  010530                          $EOPCT
010542  104400  010604                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
010546  013746  001100                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
```

```
010552  104404                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
010554  104400  010621                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
010560                          $GET42:

010560  013700  000042                  MOV     @#42,R0         ;;GET MONITOR ADDRESS
010564  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
010566  000005                          RESET                   ;;CLEAR THE WORLD
010570  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
010572  000240                          NOP                     ;;SAVE ROOM
010574  000240                          NOP                     ;;FOR
010576  000240                          NOP                     ;;ACT11
010600                          $DOAGN:
010600  000137  002516                  JMP     @#MODCA         ;;RETURN
010604  005015  047105  020104  $ENDMG: .ASCIZ  <15><12>/END PASS #/
010612  040520  051523  021440
010620     000
010621     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                                ;*******************************************************************
                                        ;ROUTINE TO ECHO THE KEYBOARD. KEYS FOR TAB,BELL,CARRIAGE
                                        ;AND LINE FEED ECHO A MNEMONIC,NON-DISPLAY CHAR. ECHO OCTAL
                                        ;EQUIVALENTS AND DISPLAY CHAR. ECHO THEMSELVES.
                                        ;(EXAMPLES-CHAR.,SPACE, ESC, SPACE OR 037, SPACE.) A
                                        ;CONTROL C (003) WILL CAUSE A TEST EXIT.

                                ;*******************************************************************

                                ;*******************************************************************
010624  000004                  TST21:  SCOPE
010626  012737  000001  001156          MOV     #1,$TIMES       ;;DO 1 ITERATION
010634  012737  010642  001106          MOV     #KYBD,$LPADR    ;;SET SCOPE LOOP ADDRESS

010642  004037  016136          KYBD:   JSR     R0,RESPTR
```

```
010646  012702  025634                 MOV     #DKYBD,R2       ;LOAD MESSAGE ADDRESS INR2
010652  004037  017172                 JSR     R0,DSMES        ;DISPLAY KEYBOARD MESSAGE
010656  012703  026222                 MOV     #DCNTZ,R3       ;ISSUE CONTROL C EXIT MESSAGE
010662  004037  016076                 JSR     R0,LDXMIT
010666  012703  011124                 MOV     #EXMAIN,R3
010672  004037  016076                 JSR     R0,LDXMIT       ;ISSUE EXIT MAINTENANCE MODE.
010676  042737  077577  002222 KYSTRT: BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
010704  105777  020120                 TSTB    @ABUFP          ;SEE IF A CHAR. RECEIVED
010710  001001                         BNE     11$             ;YES-GO PROCESS IT
010712  000001                         WAIT                    ;WAIT FOR A CH.
010714  117701  020110         11$:    MOVB    @ABUFP,R1       ;GET RAW RECEIVED DATA
010720  004037  020414                 JSR     R0,EXTST        ;CHECK FOR EXIT CONDITIONS
010724  000402                         BR      10$             ;NO EXIT -CONTINUE.
010726  000137  003062                 JMP     ASTRT           ;EXIT TEST 4
010732  105077  020072         10$:    CLRB    @ABUFP          ;CLEAR CHAR FROM BUFFER
010736  032737  000400  002222         BIT     #ESC,VSTAT      ;CHAR.=ESC(033)?
010744  001405                         BEQ     12$             ;NO
010746  005037  014636                 CLR     ESAMB           ;YES - RESET ESC ASSEMBLY FLAG
010752  012703  025627                 MOV     #DESC,R3        ;LOAD ESC MESSAGE ADDRESS
010756  000454                         BR      KYBXMT
010760  120127  000041         12$:    CMPB    R1,#41          ;CHAR. LESS THAN 41 OR
010764  103415                         BLO     2$              ;HIGHER THAN 176, GO ECHO
010766  120127  000176                 CMPB    R1,#176         ;OCTAL EQUIVALENT
010772  101012                         BHI     2$
```

```
010774  110177  004142                 MOVB    R1,@TBUFP       ;LOAD CHAR. IN XMIT BUFF.
011000  004037  016026                 JSR     R0,XMIT1        ;GO XMIT IT
011004  112777  000040  004130         MOVB    #40,@TBUFP      ;LOAD A SPACE
011012  004037  016026                 JSR     R0,XMIT1        ;AND XMIT IT.
011016  000727                         BR      KYSTRT
011020  120137  001746         2$:     CMPB    R1,BEL          ;CHAR.=BELL?
011024  001003                         BNE     3$
011026  012703  026103                 MOV     #DBELL,R3       ;LOAD BELL MESSAGE ADDRESS
011032  000426                         BR      KYBXMT
011034  120137  001754         3$:     CMPB    R1,TAB          ;CHAR. =TAB?
011040  001003                         BNE     4$
011042  012703  026064                 MOV     #DTAB,R3        ;YES-ECHO 'TAB'
011046  000420                         BR      KYBXMT
011050  123701  001750         4$:     CMPB    CARRT,R1        ;CHAR.=CARRIAGE RETURN?
011054  001003                         BNE     5$
011056  012703  026071                 MOV     #DCR,R3         ;YES - ECHO 'C/R'.
011062  000412                         BR      KYBXMT
011064  120137  001752         5$:     CMPB    R1,LNFED        ;CHAR.=LINE FEED?
011070  001003                         BNE     6$              ;NO CHECK FOR CONTROL Z
011072  012703  026076                 MOV     #DLF,R3         ;YES - ECHO 'L/F'.
011076  000404                         BR      KYBXMT
011100  004037  017266         6$:     JSR     R0,BINOCT       ;CONVERT BINARY TO OCTAL
011104  012703  002162                 MOV     #SVER1,R3
011110  042737  077577  002222 KYBXMT: BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
011116  004037  016076                 JSR     R0,LDXMIT       ;GO XMIT BUFFER
011122  000665                         BR      KYSTRT          ;WAIT FOR NEXT CHAR.

                                       ;SEQUENCE TO EXIT MAINTENANCE MODE.
011124     033     117     141 EXMAIN: .BYTE   .ESC,.O,.DMAIN,0
011127     000

                               ;*************************************************************
                                       ;ROUTINE TO UTILIZE THE VT61 AS A PRINTER CONTROLLER.
                                       ;AFTER TEST MESSAGE IS DISPLAYED, THE TEST WAITS
                                       ;FOR A C/R BEFORE ACTUALLY ENTERING TEST. A PATTERN
                                       ;OF INCREMENTING, ROLLING CHAR. WILL BE OUTPUTTED UNTIL A
                                       ;CONTROL C(003) IS RECEIVED.
```

```

                                        ;****************************************************

                                        ;****************************************************
011130  000004                          TST22:  SCOPE
011132  012737  000001  001156                  MOV     #1,$TIMES       ;;DO 1 ITERATION
011140  012737  011146  001106                  MOV     #TPRNT,$LPADR   ;;SET SCOPE LOOP ADDRESS

011146  012702  026266                  TPRNT:  MOV     #DPRNT,R2       ;LOAD PRINTER MESSAGE ADDRESS
011152  004037  017172                          JSR     R0,DSMES        ;AND ISSUE IT
011156  012703  011124                          MOV     #EXMAIN,R3
011162  004037  016076                          JSR     R0,LDXMIT       ;ISSUE EXIT MAINTENANCE MODE.
011166  004037  017364                          JSR     R0,GTCR         ;GO SET CARRIAGE RETURN
011172                                  3$:
011172  013746  002166                          MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
011176  013746  001774                          MOV     EPNT,-(SP)      ;;PUSH EPNT ON STACK
011202  013746  002130                          MOV     ESCN,-(SP)      ;;PUSH ESCN ON STACK
011206  004037  013322                          JSR     R0,TESC
011212  013701  015136                          MOV     TBBUF,R1        ;LOAD R1 WITH XMIT BUFFER
```

```
011216  012705  000041                  4$:     MOV     #41,R5          ;R5=1ST CHAR
011222  042737  077577  002222          5$:     BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
011230  013701  015136                          MOV     TBBUF,R1
011234  012703  000132                          MOV     #132,R3         ;R3= LINE WIDTH
011240  004037  017102                          JSR     R0,BLDINA       ;GO BUILD A SLIDING PATTERN.
011244  013721  001750                          MOV     CARRT,(R1)+     ;LOAD A C/R AND L/F
011250  013721  001752                          MOV     LNFED,(R1)+
011254  012737  000134  015144                  MOV     #134,XMCNT      ;SET UP TO XMIT BY BYTES.
011262  052777  000100  170444                  BIS     #TENA,@VXCSR
011270  032737  000001  002222                  BIT     #XMDNE,VSTAT    ;WAIT FOR XMIT DONE
011276  001774                                  BEQ     .-6
011300  004037  017572                          JSR     R0,CKABRT       ;CHECK FOR A PERIPHERAL ABORT.
011304  004037  020414                          JSR     R0,EXTST        ;CHECK FOR EXIT REQUEST.
011310  000402                                  BR      6$              ;NO-CONTINUE
011312  000137  003062                          JMP     ASTRT           ;YES-EXIT TEST!!
011316  122705  000177                  6$:     CMPB    #177,R5         ;EXCEEDED PATT. LIMIT?
011322  001337                                  BNE     5$              ;NO
011324  000734                                  BR      4$              ;YES RESET IT

                                        ;****************************************************

                                                ;ROUTINE TO LOOP DATA/COMMANDS FROM THE VT61 BACK TO
                                                ;THE VT61. DATA TRANSMISSIONS RESULTING FROM A ESC
                                                ;SEQUENCE WILL ALSO BE LOOPED AND WILL ENTER THE SCREEN
                                                ;AT THE CURSOR POSITION.THIS TEST CAN BE USED TO SIMULATE,
                                                ;OR CREATE, SPECIFIC SCREEN PATTERNS AND OPERATIONS.
                                        ;****************************************************

                                        ;****************************************************
011326  000004                          TST23:  SCOPE
011330  012737  000001  001156                  MOV     #1,$TIMES       ;;DO 1 ITERATION
011336  012737  011344  001106                  MOV     #LPTST,$LPADR   ;;SET SCOPE LOOP ADDRESS

011344  004037  016136                  LPTST:  JSR     R0,RESPTR       ;RESET POINTERS
011350  012702  026111                          MOV     #DLOOP,R2       ;LOAD LOOP MESSAGE ADDRESS
011354  004037  017172                          JSR     R0,DSMES        ;DISPLAY IT
011360  012703  011124                          MOV     #EXMAIN,R3
011364  004037  016076                          JSR     R0,LDXMIT       ;ISSUE EXIT MAINTENANCE MODE.
011370  004037  020164                          JSR     R0,LOOP         ;GO LOOP VT61
011374  000137  003062                          JMP     ASTRT           ;ENTER MAN MODE VIA SCOPE ROUTINE.

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

```

                                                    ;PRODUCTION KEYBOARD TEST.  ALL KEYS MUST BE DEPRESSED
                                                    ;IN THE SEQUENCE INDICATED ON THE SCREEN.  ALL ERRORS
                                                    ;OR MISTAKES ARE DISPLAYED IN OCTAL POSITIONAL FORMAT AND THE
                                                    ;CORRECT KEY POSITION IN THE ROW  IS DISPLAYED IN DECIMAL.
                                                    ;THIS TEST IS RUN IN MAINTENANCE MODE, THEREFORE THE KEYS
                                                    ;WILL ECHO THEIR POSITION ,NOT THEIR INDICATED MNEMONIC. 10
                                                    ;ERRORS WILL CAUSE AN AUTOMATIC EXIT FROM TEST.
                                            ;*************************************************************

                                            ;*************************************************************
011400  000004                              TST24:  SCOPE
011402  012737  000001  001156                      MOV     #1,$TIMES       ;;DO 1 ITERATION
011410  012737  011416  001106                      MOV     #PDKBD,$LPADR   ;;SET SCOPE LOOP ADDRESS
```

```


011416  012702  026502                      PDKBD:  MOV     #DKBD   ,R2
011422  004037  017172                              JSR     R0      ,DSMES  ;DISPLAY KEYBOARD TEST MESSAGE.
011426  005037  002206                              CLR     BUBCT           ;CLEAR ERROR COUNT LOCATION.
011432  005005                                      CLR     R5

011434  016504  011560                      DOAROW: MOV     DTTBL(R5),R4    ;SET UP 'GOOD' CHAR. POINTER
011440  016503  011532                              MOV     MSTBL(R5),R3
011444  001414                                      BEQ     FEXIT           ;MESSAGE WAS ZERO-EXIT.
011446  100421                                      BMI     CLMAIN          ;IF MESSAGE IS -1,CLEAR MAINT. MODE.
011450  004037  016076                              JSR     R0,LDXMIT       ;ISSUE 'ROW OR FUNCTION' MESSAGE.
011454  004037  017716                              JSR     R0,CKKBD        ;GO CHECK IT.
011460  123727  002206  000012                      CMPB    BUBCT,#10.      ;TEN ERROR EXIT?
011466  103401                                      BLO     1$              ;NO-CONTINUE.
011470  000402                                      BR      FEXIT           ;YES-EXIT TEST.
011472  005725                              1$:     TST     (R5)+           ;INCREMENT OFFSET.
011474  000757                                      BR      DOAROW          ;NO-DO NEXT ROW/FUNCTION.
011476  012702  026253                      FEXIT:  MOV     #DEXT,R2
011502  004037  017172                              JSR     R0,DSMES        ;ISSUE EXIT MESSAGE
011506  000137  003062                              JMP     ASTRT
011512  012703  011526                      CLMAIN: MOV     #RSMAIN,R3      ;SET UP TO EXIT MAINT. MODE.
011516  004037  016076                              JSR     R0,LDXMIT
011522  005725                                      TST     (R5)+           ;INCREMENT OFFSET.
011524  000743                                      BR      DOAROW          ;NOW TEST CONTROL AND SHIFT FUNCTIONS.
011526     033     117     141              RSMAIN: .BYTE   .ESC,.O,.DMAIN,0
011531     000



                                                    ;TABLE OF MESSAGE ADDRESSES.

011532  026705  027012  027047              MSTBL:  .WORD   DTOP,DSEC,DTHRD,DBOT
011540  027176
011542  027254  027300  177777                      .WORD   DSPCE,DKPD,-1,DCONT,DLSHFT,DRSHFT,0
011550  027126  026633  026737
011556  000000

011560  027501  027522  027542              DTTBL:  .WORD   ROW1,ROW2,ROW3,ROW4,SPCB
011566  027560  027602
011572  027604  000000  027576                      .WORD   KYPD,0,CNTRA,SHFTA,SHFTA
011600  027600  027600

                                            ;*************************************************************
                                                    ;SUBROUTINE TO ALLOW SETUP FROM MULTIPLE ENTRIES
                                            ;*************************************************************

011604                                      SETA:
```

K04

```
011604  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
011610  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
011612  022706  001126                  CMP     #$BDDAT,R6      ;;DONE?
011616  001374                          BNE     .-6             ;;LOOP BACK IF NO
011620  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
011624  012737  020604  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
011632  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
011640  012737  021060  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
011646  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
```

```
011654  012737  022436  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
011662  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
011670  012737  022276  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
011676  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
011704  013737  010536  010530          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
011712  005037  001156                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
011716  005037  001160                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
011722  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
011730  012737  011730  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
011736  012737  011736  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
011744  013746  000004                  MOV     @#4,-(SP)       ;;SAVE ERROR VECTOR
011750  013746  000006                  MOV     @#6,-(SP)
011754  012737  011770  000004          MOV     #1$,4           ;;SET UP TIMEOUT VECTOR
011762  005777  167150                  TST     @SWR            ;;TRY TO REFERENCE HARDWARE SWR
011766  000407                          BR      2$              ;;BRANCH IF NO TIMEOUT TRAP OCCURS
011770  012737  000176  001136  1$:     MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWITCH REG.
011776  012737  000174  001140          MOV     #DISPREG,DISPLAY ;;POINT TOSOFTWARE DISPLAY REG.
012004  022626                          CMP     (SP)+,(SP)+     ;;RESTORE STACK
012006  012637  000006          2$:     MOV     (SP)+,@#6       ;;RESTORE ERROR VECTOR
012012  012637  000004                  MOV     (SP)+,@#4
012016  104400  022542                  TYPE    ,STUPM  ;ISSUE SET-UP MESSAGE.
012022  012737  012042  000010          MOV     #TRPA,@#10      ;AND VECTOR
012030  000230                          SPL     0               ;PROCESSOR IS 11/45?
012032  012737  000004  017072          MOV     #4,PMULT        ;YES-DELAY MULTIPLIER = 4
012040  000416                          BR      RTRP

012042  022626                  TRPA:   POP2SP                  ;NO
012044  012737  012066  000010          MOV     #TRPB,@#10      ;RELOAD TRAP ADDRESS
012052  006737  002160                  SXT     CHRD            ;PROCESSOR IS 11/40 OR 35?
012056  012737  000002  017072          MOV     #2,PMULT        ;YES-DELAY MULTIPLIER=2
012064  000404                          BR      RTRP

012066  022626                  TRPB:   POP2SP
012070  012737  000001  017072          MOV     #1,PMULT        ;PROCESSOR MUST BE 11/05
012076  012737  000012  000010  RTRP:   MOV     #12,@#10        ;RESTORE TRAP CATCHER
012104  105737  002174                  TSTB    MODE            ;CHECK MODE FOR CORRECT EXIT.
012110  001402                          BEQ     70$
012112  000137  002274                  JMP     MANSA           ;EXIT TO MANUAL SELECT
012116  000137  002240          70$:    JMP     AUTOA           ;EXIT TO AUTO MODE.
                                ;*****************************************************************
                                        ;THIS ROUTINE MAPS ALL POSSIBL.. DL11 ADDRESSES AND STORES
                                        ;THEM IN A TABLE (INTAB). ALL ADDRESSES WHICH DO NOT
                                        ;RESULT IN TIMEOUTS ARE STORED.
                                ;*****************************************************************

012122  012701  000300          TRPVEC: MOV     #300,R1         ;START AT BEG. OF FLOATING VECTORS
012126  012702  000302                  MOV     #302,R2         ;
012132  012703  000004                  MOV     #4,R3           ;R3 CONTAINS IOT TRAP INST.
012136  010221                  1$:     MOV     R2,(R1)+        ;START LOADING ADDRESSES
012140  010321                          MOV     R3,(R1)+        ;LOAD THE TRAP
012142  062702  000004                  ADD     #4,R2           ;ASSUME 4 REGISTERS PER INTERFACE
012146  020127  001000                  CMP     R1,#1000        ;DONE?
```

```
012152  002771                                   BLT     1$              ;NO CONTINUE LOADING TRAPS
012154  012737  000340  000006                   MOV     #340,@#6        ;SET TIMEOUT TRAP TO A PSW OF 7.
012162  012737  012222  000004                   MOV     #TPENT,@#4      ;SET UP TIME-OUT TRAP ADDRESS
012170  005001                                   CLR     R1              ;CLEAR THE TABLE POINTER
```

```
012172  012705  001650                           MOV     #INTAB,R5       ;R5=DESTINATION TABLE
012176  016102  001714                   FADD:   MOV     STRTAB(R1),R2   ;PUT THE ADDRESS TO BE TESTED IN R2
012202  026102  001722                   TRPE:   CMP     ENDTAB(R1),R2   ;HAVE WE EXCEEDED END OF TABLE ADDRESS?
012206  103407                                   BLO     TBLCK           ;YES GET NEXT BASE ADDRESS.
012210  005712                                   TST     @R2             ;ADDRESS THE DEVICE IF POSSIBLE
012212  010225                                   MOV     R2,(R5)+        ;IF WE GOT THIS FAR THERE IS A DEVICE THERE-SAVE IT
012214  062702  000010                   FADD1:  ADD     #10,R2          ;INCREMENT TO THE NEXT POSSIBLE ADDRESS
012220  000770                                   BR      TRPE            ;GO TEST THE NEXT ADDRESS
012222  022626                           TPENT:  POP2SP                  ;RESTORE THE STACK AND TEST
012224  000773                                   BR      FADD1           ;NEXT ADDRESS
012226  005721                           TBLCK:  TST     (R1)+           ;BUMP AREA COUNTER BY 2.
012230  032701  000004                           BIT     #BIT02,R1       ;SEE IF BOTH DL11 AREAS CHECKED.
012234  001760                                   BEQ     FADD            ;NO-GO CHECK THE OTHER AREA
012236  005015                                   CLR     (R5)            ;SET UP TABLE TERMINATOR OF  ZEROS.
012240  000200                                   RTS     R0
                                         ;*************************************************************
                                                 ;THIS ROUTINE WILL INSURE THAT THE DEVICE(DL11)
                                                 ;WILL INTERRUPT WHEN XMIT INT. ENABLE BIT IS SET.
                                         ;*************************************************************

012242  005046                           CDEV:   CLR     -(SP)                   ;CLEAR THE PSW,LSI11 STYLE.
012244  012746  012252                           MOV     #100$,-(SP)
012250  000002                                   RTI
012252  012737  000004  000004           100$:   MOV     #4,@#4                  ;INSTALL IOT TRAP INST. AT LOCATION 4.
012260  012737  012356  000020                   MOV     #TDEV,@#IOTVEC          ;SET UP IOT TRAP EXIT ADDRESS
012266  012737  000340  000022                   MOV     #340,@#IOTVEC+2         ;SET PSW TO 7-ALLOW NO OTHER INTERRUPTS
012274  000005                                   RESET                           ;INSURE ALL XMIT FLAGS HIGH.
012276  012703  001550                           MOV     #VVECT,R3               ;VECTOR STORAGE ADDRESS SET
012302  012702  001650                           MOV     #INTAB,R2               ;PRIMARY DEVICE TABLE ADDRESS SET
012306  012705  001610                           MOV     #DLTBL,R5               ;FIN DEVICE TABLE ADDRESS SET.
012312  012701  001730                   CDEVA:  MOV     #VRCSR,R1               ;VT61 DEVICE ADDRESS SET.
012316  005712                                   TST     (R2)                    ;CHECKED ALL DEVICES?
012320  001506                                   BEQ     AOUT                    ;YES-EXIT
012322  100403                                   BMI     1$                      ;INSURE ADDRESS IS IN PROPER RANGE(17XXXX)

012324  062702  000002                           ADD     #2,R2                   ;ADDRESS IS DEFINITELY NOT GOOD -PURGE
012330  000770                                   BR      CDEVA                   ;AND LOOK FOR ANOTHER.
012332  004037  013040                   1$:     JSR     R0,LDADD                ;LOAD NEXT ADDRESSES TO BE CHECKED
012336  012701  001200                           MOV     #1200,R1                ;NOW USE R1 AS FAILSAFE COUNTER
012342  052777  000100  167364                   BIS     #TENA,@VXCSR            ;SET XMIT ENABLE
012350  005301                                   DEC     R1                      ;IF DEVICE DOES NOT INTERRUPT WITHIN
012352  001376                                   BNE     .-2                     ;APPROX. 200US IT IS NOT A DL11.
012354  000756                                   BR      CDEVA                   ;THEREFORE, GO TRY ANOTHER DEVICE.
012356  042777  000100  167350           TDEV:   BIC     #TENA,@VXCSR            ;CLEAR XMIT ENABLE.
012364  162716  000010                           SUB     #10,(R6)                ;RESET TO RECEIVER VECTOR ADDRESS
012370  012613                                   MOV     (R6)+,(R3)              ;STORE IT IN VECTOR TABLE(VVECT).
012372  005726                                   TST     (R6)+                   ;POP THE OLD PSW AND DISCARD
012374  022626                                   POP2SP                          ;POP THE ADD. AND PSW PRIOR TO INTERRUPT.
                                         ;************************************************************
                                                 ;THIS ROUTINE IS A QUICK TEST OF ANY DL11 ENCOUNTERED
                                                 ;A DATA PATTERN WILL BE RUN ON ALL ENTRIES IN INTAB
                                         ;************************************************************
012376  005046                                   CLR     -(SP)                   ;CLEAR THE PSW,LSI11 STYLE.
012400  012746  012406                           MOV     #100$,-(SP)
012404  000002                                   RTI
012406  012301                           100$:   MOV     (R3)+,R1                ;GET THE RECEIVE VECTOR ADDRESS
```

```
012410  012721  013650                  MOV     #RECAD,(R1)+        ;AND STORE SAME.
012414  012721  000340                  MOV     #340,(R1)+          ;SET RECEIVE PSW TO 7.
012420  012721  013732                  MOV     #TSMAD,(R1)+        ;STORE THE XMIT VECTOR ADDRESS
012424  012711  000340                  MOV     #340,(R1)           ;SET XMIT PSW TO 7.
012430  012704  000001                  MOV     #BIT00,R4           ;R4 IS NOW DATA PATTERN OF 1.
012434  005001                          CLR     R1                  ;SET UP FAILSAFE DELAY.
012436  052777  000100  167264          BIS     #RDENA,@VRCSR       ;SET RECEIVE ENABLE.
012444  052777  000104  167262          BIS     #TCOMB,@VXCSR       ;ENABLE XMIT INT. AND MAINTENACE .
012452  105704                  1$:     TSTB    R4                  ;XMIT PATTERN COMPLETE?
012454  001423                          BEQ     GDAD                ;YES GO STORE THIS ADDRESS
012456  005301                          DEC     R1                  ;CYCLE TIMEOUT DELAY
012460  001374                          BNE     1$                  ;NOT YET 'TIMEOUT',KEEP CYCLING.
012462  162703  000002                  SUB     #2,R3               ;RESET VECTOR POINTER
012466  042777  000104  167240          BIC     #TCOMB,@VXCSR       ;CLEAR XMIT AND RECEIVE INT. ENABLES.
012474  042777  000100  167226          BIC     #RDENA,@VRCSR
012502  104400  023020                  TYPE    ,DLERR              ;ISSUE DL11 FAILURE MESSAGE.
012506  013746  001730                  MOV     VRCSR,-(SP)         ;;SAVE VRCSR FOR TYPEOUT
                                                                    ;;TYPE BD. ADDRESS
012512  104402                          TYPOS                       ;;GO TYPE--OCTAL ASCII
012514     006                          .BYTE   6                   ;;TYPE 6 DIGIT(S)
012515     001                          .BYTE   1                   ;;TYPE LEADING ZEROS
012516  104400  001167                  TYPE    ,$CRLF
012522  000673                          BR      CDEVA               ;GO TRY ANOTHER SET OF ADDRESSES.
012524  013725  001730          GDAD:   MOV     VRCSR,(R5)+         ;SAVE GOOD ADDRESS IN DL TABLE
012530  005077  167176                  CLR     @VRBUF              ;CLEAR ANY RECEIVE FLAG STILL SET.
012534  000666                          BR      CDEVA               ;;CHECK ANOTHER DL11
012536  005015                  AOUT:   CLR     (R5)                ;SET A ZERO TABLE TERMINTATOR.
012540  012737  000006  000004          MOV     #6,@#4              ;RESTORE LOCATION 4 TO HALT CONDITION
012546  005037  000006                  CLR     @#6                 ;TO CATCH ERRORS AND ILLEGAL INTERRUPTS.
012552  012737  020604  000020          MOV     #$SCOPE,@#IOTVEC            ;RELOAD IOT VECTOR FOR SCOPE
012560  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;LOOP.
012566  012701  000300                  MOV     #300,R1
012572  012702  000302                  MOV     #302,R2
012576  010221                  1$:     MOV     R2,(R1)+
012600  005021                          CLR     (R1)+               ;RESTORE HALTS TO ALL LOCATIONS CONTAINING IOTS
012602  062702  000004                  ADD     #4,R2
012606  020127  001000                  CMP     R1,#1000            ;TO LOCATION 1000
012612  103771                          BLO     1$
012614  000005                          RESET                       ;CLEAR ALL FLAGS
012616  000200                          RTS     R0

                                ;*************************************************************
                                        ;INITIALIZATION ROUTINE FOR AUTO SELECTION. THIS ROUTINE
                                        ;WILL INSURE THAT ALL DL11S IN DLTBL HAVE A VT61 CONNECTED
                                        ;ALL UNITS WHICH CANNOT CORRECTLY RESPOND WILL BE PURGED.
                                ;*************************************************************

012620  012702  001610          INITA:  MOV     #DLTBL,R2           ;R2 POINTS TO DL11 ADDRESS TABLE
012624  012703  001550                  MOV     #VVECT,R3           ;R3 POINTS TO DL11 VECTOR
012630  012701  001730          11$:    MOV     #VRCSR,R1           ;POINTER TO VT61 DL11
012634  005712                          TST     (R2)                ;SEE IF ALL CHECKED
012636  001447                          BEQ     INTXT               ;YES-EXIT
012640  004037  013040                  JSR     R0,LDADD            ;NO-GO LOAD THE ADDRESSES
012644  062703  000002                  ADD     #2,R3               ;UPDATE VECTOR COUNT
012650  004037  015326                  JSR     R0,ZFLAG            ;ISSUE ESCZ AND LOOK FOR RESPONSE.
012654                          2$:
```

```
012654  012637  002160                  MOV     (SP)+,CHRD          ;;POP STACK INTO CHRD
012660  100414                          BMI     5$                  ;TIMEOUT OCCURRED NO CHARACTER
012662  123727  002160  000140          CMPB    CHRD,#140           ;CHECK IDENT FOR VT61 IDENTIFIERS
```

```
012670  103410                          BLO     5$              ;NOT A VT61-SET UP TO PURGE ADDRESS
012672  123727  002160  000172          CMPB    CHRD,#172       ;IDENTS ARE SMALL A THRU Z
012700  101004                          BHI     5$              ;NOT A VT61-PURGE
012702                          4$:
012702  012637  002160                  MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
012706  001375                          BNE     4$
012710  000747                          BR      11$             ;TEST ANOTHER ADDRESS
012712                          5$:
012712  012637  002160                  MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
012716  001375                          BNE     5$
012720  162702  000002                  SUB     #2,R2           ;RESET ADDRESS AND VECTOR FOINTERS
012724  162703  000002                  SUB     #2,R3
012730  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
012732  012746  000001                  MOV     #1,-(SP)        ;;PUSH #1 ON STACK
012736  004037  013250                  JSR     R0,BBLUP
012742  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
012744  012746  000001                  MOV     #1,-(SP)        ;;PUSH #1 ON STACK
012750  004037  013250                  JSR     R0,BBLUP
012754  000725                          BR      11$             ;TRY ANOTHER DL11 ADDRESS.

012756  005737  001610          INTXT:  TST     DLTBL           ;CHECK TO INSURE GOOD ADDRESSES
012762  001012                          BNE     EXINT           ;YES-GO TO NEXT TEST
012764  104400  022727                  TYPE    ,NOVT           ;NO-ISSUE NO VT61 MESSAGE.
012770  012737  005670  017074          MOV     #3000.,DCOUNT   ;SET DELAY TO 30 SEC.
012776  004037  017032                  JSR     R0,DELAY        ;AND DO IT.
013002  062700  000004                  ADD     #4,R0           ;SET UP 'NO VT61 FOUND' EXIT
013006  000200                          RTS     R0
013010  012702  001610          EXINT:  MOV     #DLTBL,R2       ;LOAD AND ISSUE GOOD ADDRESSES
013014  104400  022656                  TYPE    ,DUNTST         ;OF RESPONSIVE VT61S.
013020                          1$:
013020  012246                          MOV     (R2)+,-(SP)     ;;SAVE (R2)+ FOR TYPEOUT
                                                                ;;TYPE AN ADDRESS
013022  104402                          TYPOS                   ;;GO TYPE--OCTAL ASCII
013024     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
013025     001                          .BYTE   1               ;;TYPE LEADING ZEROS
013026  104400  001167                  TYPE    ,$CRLF
013032  005712                          TST     (R2)            ;AT END OF GOOD UNITS?
013034  001371                          BNE     1$              ;NO PRINT ANOTHER ADDRESS.
013036  000200                          RTS     R0
                                ;*************************************************************
```

```
                                          ;SUBOUTINE TO LOAD 4 ADRESSES FROM THE LOCATION AT (R2).
                                          ;TO 4 LOCATION POINTED TO BY R1(TO VXBUFF+2).ROUTINE USES R4 AS
                                          ;WORK REG AND EXITSD WITH R2 INCREMENTED BY 2.
                                   ;*************************************************************

        013040  012204             LOADD:  MOV     (R2)+,R4        ;LOAD THE ADDRESS
        013042  010421             1$:     MOV     R4,(R1)+        ;STORE AN ADDRESS
        013044  062704  000002             ADD     #2,R4           ;INCREMENT ADDRESS
        013050  020127  001740             CMP     R1,#VXBUF+2     ;LOADED 4?
        013054  002772                     BLT     1$              ;NO LOAD ANOTHER
        013056  000200                     RTS     R0              ;YES-EXIT

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

```
                                          ;ROUTINE TO RECEIVE CHARACTER(S).  ENTERED WITH
                                          ;NUMBER OF CHARACTERS TO RECEIVE ON THE STACK.
                                          ;ROUTINE EXITS WITH CARACTER(S) ON STACK.  IF A
                                          ;PROGRAM TIME-OUT (100 M.S.) OCCURS BEFORE A CHARACTER
                                          ;IS RECEIVED ROUTINE EXITS WITH -1 ON STACK. FORMAT
                                          ;FOR DATA IS (BYTE2, BYTE1) ETC.  A WORD OF ZEROS TERMINATES
                                          ;DATA STRING ON THE STACK. SOM/EOM, IF SENT, ARE RECEIVED
                                          ;BUT NOT STORED.

                                   ;*************************************************************

        013060                             RECTM:
        013060  012637  002220                     MOV     (SP)+,ROSVE     ;;POP STACK INTO ROSVE
        013064  012637  002206                     MOV     (SP)+,@#BUBCT   ;;POP STACK INTO @#BUBCT
        013070  013746  002166                     MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
        013074  005037  002160             1$:     CLR     CHRD            ;CLEAR CHARACTER STORAGE LOCATION.
        013100  005037  002216                     CLR     DLAY            ;SET UP FAILSAFE DELAY
        013104  032777  000200  166616     3$:     BIT     #RECDN,@VRCSR   ;SEE IF DONE FLAG SET
        013112  001007                             BNE     4$
        013114  005337  002216                     DEC     DLAY            ;DECREMENT FAILSAFE CNTR.
        013120  001371                             BNE     3$              ;NOT AT ZERO-CONTINUE WAITING.
        013122  012737  177777  002160     31$:    MOV     #-1,CHRD        ;SET UP FOR FAILSAFE EXIT.
        013130  000442                             BR      RECEX           ;EXIT ROUTINE.
        013132  117737  166574  002160     4$:     MOVB    @VRBUF,CHRD     ;STORE THIS CHARACTER.
        013140  042737  000200  002160             BIC     #200,CHRD       ;STRIP PARITY BIT.
        013146  122737  000057  002160             CMPB    #SLSH,CHRD      ;RECEIVED A IDENT SLASH(57)?
        013154  001007                             BNE     41$             ;NO-STORE A CHARACTER.
        013156  105337  002207                     DECB    BUBCT+1         ;DECREMENT ALLOWABLE SLASH COUNT.
        013162  001757                             BEQ     31$             ;COUNT EQUAL ZERO-SET UP ERROR EXIT.
        013164  123727  002207  000213             CMPB    BUBCT+1,#139.   ;RECEIVED FIRST SLASH?
        013172  103740                             BLO     1$              ;YES-IGNORE THIS ONE.
        013174  122737  000002  002160     41$:    CMPB    #SOM,CHRD       ;IS CHAR. ACTUALLY SOM?
        013202  001003                             BNE     5$              ;NO
        013204  105237  002206                     INCB    BUBCT           ;YES -SET UP TO RECEIVE EOM ALSO
        013210  000731                             BR      1$              ;AND RECEIVE NEXT CHAR.
        013212  122737  000004  002160     5$:     CMPB    #EOM,CHRD       ;CHAR. = EOM?
        013220  001410                             BEQ     RECEXA          ;YES- DO NOT PUSH IT ON STACK
        013222  105337  002206                     DECB    BUBCT           ;DECREMENT CHARACTER COUNT.
        013226  001403                             BEQ     RECEX           ;COUNT=0. EXIT WERE DONE.
        013230  013746  002160                     MOV     CHRD,-(SP)      ;;PUSH CHRD ON STACK
        013234  000717                             BR      1$              ;GO READ AGAIN.

        013236                             RECEX:
        013236  013746  002160                     MOV     CHRD,-(SP)      ;;PUSH CHRD ON STACK
        013242                             RECEXA:
        013242  013746  002220                     MOV     ROSVE,-(SP)     ;;PUSH ROSVE ON STACK
        013246  000200                             RTS     R0
```

```
                                                  ;*********************************************************************
                                                          ;THIS POUTINE WILL 'BUBBLE UP' XX WORDS TO
                                                          ;ELIMINATE NON-RESPONSIVE ADDRESSES.  ENTERED
                                                          ;WITH ADDRESS TO BE 'BUBBLED' TO ON THE STACK. LOCATIONS
                                                          ;ELIMINATED WILL BE FILLED WITH ZEROS.  THE STACK MUST ALSO
                                                          ;BE LOADED WITH THE NUMBER OF POSITIONS TO BUBBLE.
                                                  ;*********************************************************************
```

```

 013250                                           BBLUP:
 013250  012637  002220                                   MOV     (SP)+,ROSVE       ;;POP STACK INTO ROSVE
 013254  012637  002206                                   MOV     (SP)+,BUBCT       ;;POP STACK INTO BUBCT
 013260  012637  002204                                   MOV     (SP)+,TOADD       ;;POP STACK INTO TOADD
 013264  010446                                           MOV     R4,-(SP)          ;;PUSH R4 ON STACK
 013266  013704  002204                           2$:     MOV     @#TOADD,R4        ;PUT LAST GOOD DL11 ADDRESS IN R4
 013272  012437  002160                                   MOV     (R4)+,CHRD        ;MOVE NEXT WORD TO CHRD FOR STORAGE
 013276  012464  177774                           1$:     MOV     (R4)+,-4(R4)      ;BUBBLE UP DATA.
 013302  001375                                           BNE     1$                ;BUBBLE UNTIL ZERO BYTE MOVED.
 013304  005337  002206                                   DEC     BUBCT             ;SUBTRACT ONE FROM BUBBLE COUNT.
 013310  001366                                           BNE     2$                ;IF BUBBLE COUNT NOT ZERO - DO AGAIN.
 013312                                           3$:
 013312  012604                                           MOV     (SP)+,R4          ;;POP STACK INTO R4
 013314  013746  002220                                   MOV     ROSVE,-(SP)       ;;PUSH ROSVE ON STACK
 013320  000200                                           RTS     RO                ;YES-EXIT


                                                  ;*********************************************************************
                                                          ;THIS ROUTINE OUTPUTS THE ESC SEQNENCE FOUND ON
                                                          ;THE STACK. A WORD OF ZEROS MUST TERMINATE THE SEQUENCE.
                                                          ;FORMAT FOR STACK WORD IS SEQ-ESC, IE-XXX033.
                                                  ;*********************************************************************

 013322                                           TESC:
 013322  012637  002220                                   MOV     (SF.- ROSVE       ;;POP STACK INTO ROSVE
 013326  010437  002206                                   MOV     R4,BUBCT          ;SAVE R4.
 013332  112777  000002  166376                           MOVB    #SOM,@VXBUF       ;SEND A START OF MESSAGE.
 013340  012705  177777                           1$:     MOV     #-1,R5            ;ALL ONES THO CHECK LOCATION.
 013344  012604                                           MOV     (R6)+,R4          ;GET COMMAND FROM STACK.
 013346  001415                                           BEQ     3$                ;IF ZERO TERMINATOR FOUND-EXIT.
 013350  110405                                           MOVB    R4,R5             ;LOAD CHECK BYTE.
 013352  105704                                   2$:     TSTB    R4                ;CHECK BYTE FOR A ZERO.
 013354  001406                                           BEQ     20$               ;IF ZERO-DO NOT XMIT IT.
 013356  032777  000200  166350                           BIT     #TRDY,@VXCSR
 013364  001774                                           BEQ     .-6               ;WAIT FOR XMIT READY BIT
 013366  110477  166344                                   MOVB    R4,@VXBUF         ;XMMIT A BYTE.
 013372  000304                                   20$:    SWAB    R4                ;GET THE OTHER BYTE .
 013374  120405                                           CMPB    R4,R5             ;IF GOOD COMPARE WE HAVE CHECKED BOTH
 013376  001760                                           BEQ     1$                ;BYTES SO POP ANOTHER WORD.
 013400  000764                                           BR      2$                ;GO XMIT ANOTHER BYTE
 013402  032777  000200  166324                   3$:     BIT     #TRDY,@VXCSR      ;SEE IF READY SET
 013410  001774                                           BEQ     .-6
 013412  012777  000004  166316                           MOV     #EOM,@VXBUF       ;SEND A EOM.
 013420  032777  000200  166306                           BIT     #TRDY,@VXCSR      ;SEE IF READY SET
 013426  001774                                           BEQ     .-6
 013430  013704  002206                                   MOV     BUBCT,R4          ;RESTORE R4.
 013434  013746  002220                                   MOV     ROSVE,-(SP)       ;;PUSH ROSVE ON STACK
 013440  000200                                           RTS     RO


                                                  ;*********************************************************************
                                                          ;ROUTINE TO READ A CHARACTER FROM THE CONSOLE.
                                                          ;EXITS WITH CHARACTER ON THE STACK.
```

D05

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

013442                                     CONRD:
013442  012637  002220                             MOV     (SP)+,ROSVE       ;;POP STACK INTO ROSVE
```

```
013446  032777  000200  165466             BIT     #RECDN,@$TKS      ;LOOK FOR DONE BIT
013454  001774                             BEQ     .-6               ;WAIT FOR IT
013456  117746  165462                     MOVB    @$TKB,-(R6)       ;PUSH CHARACTER TO STACK
013462  042716  000200                     BIC     #200,(R6)         ;STRIP ANY PARITY BIT.
013466  013746  002220                     MOV     ROSVE,-(SP)       ;;PUSH ROSVE ON STACK
013472  000200                             RTS     R0


                                   ;*************************************************************
                                           ;MANUAL TEST SELECT MONITOR
                                           ;SELECTS TESTS TO BE EXECUTED FROM THOSE ENTERED IN
                                           ;INITIAL DIALOGE. IF TEST 377 WAS REQUESTED THE TESTS WILL
                                           ;REPEAT INFINITELY.
                                   ;*************************************************************

013474  105737  002174             MONIT:  TSTB    MODE              ;TEST MODE SWITCH
013500  001012                             BNE     1$                ;MANUAL MODE
013502  023737  002176  002200             CMP     FTLCNT,ALWCNT     ;COMPARE FATAL XMITS WITH ALLOWED.
013510  103405                             BLO     100$              ;FATALS LESS THAN ALLOWED-CONTINUE.
013512  104400  024461                     TYPE    ,DABRT            ;ISSUE ABORT MESSAGE.
013516  000005                     200$:   RESET                     ;CLEAR ALL INTERFACE FLAGS.
013520  000137  011604                     JMP     SETA              ;SET UP TO RESTART TEST.
013524  000200                     100$:   RTS     R0                ;AUTO MODE
013526  005726                     1$:     TST     (R6)+             ;POP THE STACK
013530  022626                             POP2SP                    ;POP SCOPE RETURN AND VECTOR
013532  005037  002176                     CLR     FTLCNT            ;DO NOT INC. FATAL COUNT IN MANUAL MODE.
013536  032777  000200  165376     10$:    BIT     #RECDN,@$TKS      ;CONSOLE ACTIVE?
013544  001407                             BEQ     11$
013546  117701  165372                     MOVB    @$TKB,R1          ;STORE INPUT BUFFER
013552  042701  000200                     BIC     #200,R1           ;CLEAR THE PARITY BIT
013556  122701  000003                     CMPB    #3,R1             ;CHAR. EQUAL ESC. C?
013562  001755                             BEQ     200$              ;YES-EXIT
013564  117701  166402             11$:    MOVB    @TSTPTR,R1        ;GET THE NEXT TEST #
013570  003005                             BGT     2$                ;NOT AT END OF LIST
013572  042777  000100  166130             BIC     #RDENA,@VRCSR     ;CLEAR REC. INTERRUPTS BEFORE NEXT UNIT SELECT.
013600  000137  002516                     JMP     MODCA             ;END OF LIST-GO SET UP NEXT 61
013604  005301                     2$:     DEC     R1                ;ADJUST OFFSET
013606  006301                             ASL     R1                ;USE TEST # TO FORM ADDRESS OFFSET
013610  016137  022472  013646             MOV     TSTADD(R1),JMPADD+2       ;LOAD NEW ADDRESS
013616  062737  000002  013646             ADD     #2,JMPADD+2       ;BYPASS INITIAL SCOPE LOOP
013624  005237  002172                     INC     TSTPTR            ;INCREMENT TEST OPINTER
013630  005037  177776                     CLR     PSW               ;SET NON-INT. PRIORITY TO ZERO
013634  005046                             CLR     -(SP)             ;CLEAR THE PSW,LSI 11 STYLE.
013636  012746  013644                     MOV     #JMPADD,-(SP)
013642  000002                             RTI
013644  000137  013644             JMPADD: JMP     JMPADD            ;EXIT TO NEXT SELECTED TEST
                                   ;*************************************************************
                                   ;*************************************************************
                                                   ;FOLLOWING ROUTINES ARE INTERRUPT HANDLERS FOR THE
                                                   ;DL11 QUICK-TEST.
                                   ;*************************************************************
                                   ;*************************************************************
013650  117737  166056  002160     RECAD:  MOVB    @VRBUF,CHRD       ;GET THE RECEIVED CHAR.
013656  042737  000200  002160             BIC     #200,CHRD         ;CLEAR ANY PARITY.
013664  120437  002160                     CMPB    R4,CHRD           ;COMPARE RECEIVED TO XMITTED
013670  001407                             BEQ     UPD4              ;AND UPDATE PATTERN IF OK.
013672  042777  000104  166034     TOFF:   BIC     #TCOMB,@VXCSR     ;DATA ERROR OCCURED OR WE ARE DONE
```

```
013700  042777  000100  166022            BIC     #RDENA,@VRCSR     ;EITHER WAY-EXIT.
013706  000002                   REEX:    RTI
013710  052777  000100  166016   UPD4:    BIS     #TENA,@VXCSR      ;ENABLE XMIT INT.
013716  106304                            ASLB    R4                ;UPDATE DATA PATTERN.
013720  032704  000200                    BIT     #BIT07,R4         ;ROTATED TO PARITY BIT?
013724  001770                            BEQ     REEX              ;NO-CONTINUE TESTING
013726  005004                            CLR     R4                ;YES-SET UP COMPLETE FLAG
013730  000760                            BR      TOFF              ;AND EXIT.
013732  110477  166000           TSMAD:   MOVB    R4,@VXBUF         ;XMIT DATA
013736  042777  000100  165770            BIC     #TENA,@VXCSR      ;CLEAR XMIT INT. UNTIL LAST BIT REC.
013744  000002                            RTI

                                 ;*****************************************************************
                                          ;RECEIVE INTERRUPT ROUTINE.AFTER EACH RECEIVE
                                          ;CYCLE BUFFER POINTER (RBUFP) WILL BE SET TO (RBBUF).
                                          ;MAX. EXECUTION TIME IS APPROX 200US, AVERAGE =100US.
                                          ;UPON RECEIPT OF XON, XMTKIL BIT IS CHECKED IN VSTAT
                                          ;AND IF SET, WILL BE CLEARED AND XMIT INT. ENABLE SET.
                                          ;LOCATION ESAMB IS USED FOR ESC ASSEMBLY FLAGS. IE. BIT
                                          ;00 SET MEANS A033 WAS RECEIVED, BIT 01 SET MEANS AN ESCP
                                          ;SEQUENCE IS BEING ASSEMBLED. BIT 03
                                          ;SET INDICATES AND ESCAPE O SEQUENCE IS BEING ASSEMBLED.
                                          ;LOCATIONS STRO AND STRP ARE USED TO STORE ESCAPE
                                          ;O AND ESCAPE P SEQUENCES DETECTED,BUT NOT UTILIZED IN TEST.
                                 ;*****************************************************************

013746                           INTRC:
013746  010146                            MOV     R1,-(SP)          ;;PUSH R1 ON STACK
013750  017701  165756                    MOV     @VRBUF,R1         ;USE R1 FOR  STORAGE OF STATUS AND CH.
013754  042701  000200                    BIC     #200,R1           ;STRIP PARITY BIT.
013760  032737  000100  002222            BIT     #TXSUM,VSTAT      ;CHECKSUM CALCULATION REQUESTED?
013766  001403                            BEQ     11$               ;NO
013770  010105                            MOV     R1,R5             ;YES-STORE CHAR. AND
013772  004037  017516                    JSR     R0,CALCK          ;CALCULATE THE CHECKSUM.
013776  005237  031030           11$:     INC     ABUFP             ;INCREMINT THE RAW DATA POINTER
014002  023727  031030  031114            CMP     ABUFP,#ABBUF+50.        ;AT THE END OF BUFFER?
014010  001003                            BNE     12$               ;NO
014012  012737  031032  031030            MOV     #ABBUF,ABUFP      ;YES-RESET IT
014020  110177  015004           12$:     MOVB    R1,@ABUFP         ;STORE THE RAW DATA
014024  001505                            BEQ     6$                ;IF CHAR. IS NULL-GO STORE IT
014026  032737  000013  014636            BIT     #BIT00+BIT01+BIT03,ESAMB      ;ESC OR ESC O?
014034  001150                            BNE     ASESC             ;YES-KEEP ASSEMBLING
014036  120137  002130                    CMPB    R1,ESCN           ;BYTE = ESCN?
014042  101076                            BHI     6$                ;NO-PROBABLY A DISPLAY CH.-STORE IT.
014044  001007                            BNE     1$                ;NO-DECODE FOR XON,XOFF,SOM,EOM
014046  012737  000001  014636            MOV     #1,ESAMB          ;YES SET ESC ASSEMBLY FLAG.
014054  052737  000400  002222            BIS     #ESC,VSTAT        ;SET ESC RECEIVED FLAG
014062  000515                            BP      RSTER             ;AND EXIT
014064  120127  000023           1$:      CMPB    R1,#XOFF          ;SEE IF RECEIVED BYTE WAS XOFF
014070  001004                            BNE     2$                ;NO
014072  052737  100000  002222            BIS     #RXOFF,VSTAT      ;YES, SET XOFF IN STATUS REG.
014100  000506                            BR      RSTER             ;EXIT
014102  120127  000021           2$:      CMPB    R1,#XON           ;SEE IF BYTE WAS XON
014106  001016                            BNE     3$                ;NO
014110  042737  100000  002222            BIC     #RXOFF,VSTAT      ;YES, CLEAR XOFF IN VSTAT.
014116  032737  000200  002222            BIT     #XMKIL,VSTAT      ;CHECK XMIT KILL BIT.
```

```
014124  001474                            BEQ     RSTER             ;NOT SET, EXIT
014126  052777  000100  165600            BIS     #TENA,@VXCSR      ;SET XMIT INT. ENABLE.
014134  042737  000200  002222            BIC     #XMKIL,VSTAT      ;CLEAR THE XMIT KILLED FLAG
```

```
014142  000465                          BR      RSTER           ;EXIT
014144  120127  000002          3$:     CMPB    R1,#SOM         ;SEE IF BYTE WAS SOM
014150  001004                          BNE     4$              ;NO
014152  052737  040000  002222  31$:    BIS     #RSOM,VSTAT     ;YES, SET SOM IN VSTAT.
014160  000456                          BR      RSTER           ;EXIT

014162  120127  000004          4$:     CMPB    R1,#EOM         ;WAS BYTE EOM?
014166  001012                          BNE     5$              ;NO
014170  052737  020000  002222          BIS     #REOM,VSTAT     ;NOW SET EOM IN VSTAT.
014176  013737  014630  014634          MOV     RBBUF,RBUFP     ;RESET THE BUFFER POINTER.
014204  042737  000100  002222          BIC     #TXSUM,VSTAT    ;CLEAR CHECKSUM REQUEST BIT.
014212  000441                          BR      RSTER           ;AND EXIT
014214  123701  001750          5$:     CMPB    CARRT,R1        ;CHAR. =CARRIAGE RETURN?
014220  001403                          BEQ     51$             ;YES-GO SET END OF LINE FLAG
014222  123701  001752                  CMPB    LNFED,R1        ;CHAR.= LINEFEED?
014226  001004                          BNE     6$              ;NO- GO STORE IT
014230  052737  001000  002222  51$:    BIS     #EPL,VSTAT      ;SET END OF LINE INDICATOR
014236  000427                          BR      RSTER

014240  023737  014634  014632  6$:     CMP     RBUFP,REBUF     ;IS CIRCULAR BUFFER FILLED?
014246  001003                          BNE     61$             ;NO
014250  013737  014630  014634          MOV     RBBUF,RBUFP     ;YES, RESET POINTER TO BEGINNING
014256  032737  000020  002222  61$:    BIT     #COMGP,VSTAT    ;RECEIVING GRAPHICS CHAR.?
014264  001402                          BEQ     7$              ;NO
014266  162701  000137                  SUB     #137,R1         ;YES-SUBTRACT 137 FROM RECEIVED CHAR.

014272  032737  000040  002222  7$:     BIT     #REVID,VSTAT    ;REVERSE VIDEO MODE?
014300  001402                          BEQ     70$             ;NO STORE RECEIVED BYTE.
014302  052701  000200                  BIS     #200,R1         ;YES-FORCE BIT7 AS REV. VIDEO IND.
014306  110177  000322          70$:    MOVB    R1,@RBUFP       ;STORE BYTE AND
014312  005237  014634                  INC     RBUFP           ;INCREMENT POINTER.

014316  005701                  RSTER:  TST     R1              ;CHECK FOR STATUS ERROR
014320  100014                          BPL     RECXT           ;NO, EXIT ROUTINE

014322  052737  004000  002222          BIS     #RSTT,VSTAT     ;SET STATUS ERROR FLAG IN VSTAT
014330  027727  000326  177777          CMP     @STTEP,#-1      ;IS ERROR TABLE FULL?
014336  001405                          BEQ     RECXT           ;YES, EXIT ROUTINE
014340  010177  000316                  MOV     R1,@STTEP       ;NO, STORE STATUS ERR. AND CHECK
014344  062737  000002  014662          ADD     #2,STTEP        ;INCREMENT STATUS ERR. POINTER

014352                          RECXT:
014352  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
014354  000002                          RTI                     ;EXIT
014356  032737  000002  014636  ASESC:  BIT     #2,ESAMB        ;ASSEMBLING ESC P?
014364  001063                          BNE     AESCP           ;YES-GO GET LAST CH,
014366  032737  000010  014636          BIT     #BIT03,ESAMB    ;ASSEMBLING ESC O?
014374  001062                          BNE     AESCO           ;YES
014376  122701  000120                  CMPB    #120,R1         ;CH.= A P?
014402  001004                          BNE     10$             ;NO KEEP CHECKING
014404  052737  000002  014636          BIS     #BIT01,ESAMB    ;YES-SET ESCP ASSEMBLY FLAG
014412  000741                          BR      RSTER           ;AND EXIT
014414  122701  000077          10$:    CMPB    #77,R1          ;CHAR.IS AN ESC ? ?
```

```
014420  001403                          BEQ     110$            ;YES-FAKE AN ESC O.
014422  122701  000117                  CMPB    #117,R1         ;CHAR = O?
014426  001004                          BNE     11$             ;NO
014430  052737  000010  014636  110$:   BIS     #BIT03,ESAMB    ;YES SET ESC O ASSEMBLY FLAG
014436  000727                          BR      RSTER           ;AND EXIT
014440  123701  002052          11$:    CMPB    RDCUR,R1        ;BYTE= CURSOR POSITION?
014444  001004                          BNE     1$              ;NO-
014446  052737  000004  002222          BIS     #CURPOS,VSTAT   ;YES-SET RECEIVED CURSOR POSITION.
```

```
014454  000424                          BR      CESAM
014456  122701  000057          1$:     CMPB    #SLSH,R1                ;BYTE=TERMINAL ID ESC?
014462  001004                          BNE     2$                      ;NO-CHECK FOR GRAPHICS SEQUENCE.
014464  052737  000002  002222          BIS     #TRMID,VSTAT            ;YES-SET TERM. IDENT FLAG IN VSTAT
014472  000415                          BR      CESAM
014474  122701  000106          2$:     CMPB    #CKGP,R1                ;RECEIVED GRAPHICS CHAR. SEQUENCE?
014500  001004                          BNE     3$                      ;NO
014502  052737  000020  002222          BIS     #COMGP,VSTAT            ;YES-SET GRAPHICS DATA FLAG.
014510  000406                          BR      CESAM
014512  122701  000107          3$:     CMPB    #NCKGP,R1               ;RECEIVED RESET GRAPHICS SEQ.?
014516  001003                          BNE     CESAM                   ;NO
014520  042737  000020  002222          BIC     #COMGP,VSTAT            ;YES-SET NORMAL CHAR. RECEIVE.
014526  005037  014636          CESAM:  CLR     ESAMB                   ;CLEAR ASSEMBLY FLAG.
014532  000671                          BR      RSTER                   ;AND EXIT.

014534  110137  014666          AESCP:  MOVB    R1,STRP                 ;STORE ANY UNCHECKED FOR  ESC. P
014540  000772                          BR      CESAM

014542  123701  002020          AESCO:  CMPB    EEMP,R1                 ;BYTE=ESC O  -REV. VIDEO- ?
014546  001004                          BNE     1$                      ;NO
014550  052737  000040  002222          BIS     #REVID,VSTAT            ;YES-SET REVERSE VIDEO MODE IN VSTAT.
014556  000763                          BR      CESAM

014560  123701  002022          1$:     CMPB    DEMP,R1                 ;BYTE=ESC O DISABLE REV. VIDEO MODE?
014564  001004                          BNE     2$                      ;NO
014566  042737  000040  002222          BIC     #REVID,VSTAT            ;YES-CLEAR REVERSE VIDEO MODE IN VSTAT.
014574  000754                          BR      CESAM
014576  122701  000171          2$:     CMPB    #CPABRT,R1              ;COPIER ABORT?
014602  001403                          BEQ     3$                      ;YES-SET ABORT FLAG IN VSTAT
014604  122701  000172                  CMPB    #PRABRT,R1              ;PRINTER ABORT?
014610  001004                          BNE     4$                      ;NO
014612  052737  010000  002222  3$:     BIS     #PABRT,VSTAT            ;YES-SET THE ABORT FLAG.
014620  000742                          BR      CESAM                   ;AND EXIT.
014622  110137  014664          4$:     MOVB    R1,STRO                 ;STORE ESCAPE O COMMAND
014626  000737                          BR      CESAM

014630  000000                  RBBUF:  .WORD                           ;ADDRESS OF STAT OF BUFFER
014632  000000                  REBUF:  .WORD                           ;ADDRESS OF END OF BUFFER.
014634  000000                  RBUFP:  .WORD                           ;READ BUFFER POINTER.
014636  000000                  ESAMB:  .WORD   0                       ;ESCAPE SEQ.ASSEMBLY AREA

014640                          STTER:
014640  000000                          0
014642  000000                          0
014644  000000                          0
014646  000000                          0
014650  000000                          0
014652  000000                          0
```

```
014654  000000                          0
014656  000000                          0
014660  177777                          .WORD   -1                      ;STATUS REGISTER DELIMITER.
014662  000000                  STTEP:  .WORD                           ;STATUS ERROR POINTER.
014664  000000                  STRO:   .WORD   0                       ;ESCAPE O STORAGE
014666  000000                  STRP:   .WORD                           ;ESCAPE P STORAGE


                                ;*****************************************************************
                                        ;TRANSMIT INTERRUPT ROUTINE.
                                        ;  IF XOFF BIT IS SET IN VSTAT , TRANSMISSION WILL NOT OCCUR
                                        ;AND XMIT INT. ENABLE BIT WILL BE CLEARED AND THE ROUTINE
                                        ;WILL BE EXITED IMMEDIATELY. IF AFTER THE TRANSMISSION
                                        ;OF THE CHARACTER DURING THIS INTERRUPT CYCLE, THE
```

```
                                                ;XMIT COUNT (XMCNT) IS EQUAL TO ZERO.
                                                ;THE XMIT DONE BIT WILL BE SET IN VSTAT AND XMIT
                                                ;INT ENABLE BIT WILL CLEARED. TRANSMIT COUNT(XMCNT) MUST BE
                                                ;SET TO THE NUMBER OF BYTE/CHARACTER TO TRANSMIT.
                                                ;IF LOCATION BLKM IS SET TO 1001,A SOM WILL PRECEED THE
                                                ;DATA AND A EOM WILL FOLLOW IT. IF XMZER IS SET TO NON-
                                                ;ZERO, ALL DATA(INCLUDING ZEROS) WILL BE XMITTED.
                                        ;**************************************************************
014670  005737  002222                  INTXM:  TST     VSTAT           ;HAS 61 TRANSMITTED XOFF?
014674  100004                                  BPL     NOKIL           ;NO XMIT ANOTHER
014676  052737  000200  002222                  BIS     #XMKIL,VSTAT    ;SET XMIT KILLED BIT IN VSTAT
014704  000510                                  BR      KIENA           ;GO KILL XMIT ENABLE

014706  105737  002225                  NOKIL:  TSTB    BLKM+1          ;SOM/EOM TRANSMIT?
014712  001406                                  BEQ     NOSOM           ;NO
014714  112777  000002  165014                  MOVB    #SOM,@VXBUF     ;YES-ISSUE START OF MESSAGE.
014722  105037  002225                          CLRB    BLKM+1          ;AND CLEAR SOM FLAG.
014726  000002                                  RTI
014730  005737  015144                  NOSOM:  TST     XMCNT           ;XMITTED THE BUFFER?
014734  001006                                  BNE     100$            ;NO-XMIT A NORMAL CHAR.
014736  112777  000004  164772                  MOVB    #EOM,@VXBUF     ;YES SEND EOM AND EXIT
014744  105037  002224                          CLRB    BLKM
014750  000452                                  BR      2$
014752  105777  000164                  100$:   TSTB    @TBUFP          ;CHECK FOR CH.= ZERO. IF SO DO NOT XMIT
014756  001016                                  BNE     1$              ;OR COUNT BYTE. OR ARE WE
014760  005737  020466                          TST     XMZER           ;XMITTING ZEROS?
014764  001023                                  BNE     22$             ;YES-XMIT NEXT BYTE
014766  023737  015142  015140                  CMP     TBUFP,TEBUF     ;AT END OF BUFFER?
014774  001004                                  BNE     10$             ;NO
014776  013737  015136  015142                  MOV     TBBUF,TBUFP     ;YES-RESET BUFFER POINTER
015004  000740                                  BR      NOKIL
015006  005237  015142                  10$:    INC     TBUFP
015012  000735                                  BR      NOKIL           ;LOOK FOR NON-ZERO BYTE TO TRANSMIT.

015014  032737  002000  002222          1$:     BIT     #CKSUM,VSTAT    ;CHECKSUM REQUESTED?
015022  001404                                  BEQ     22$
015024  117705  000112                          MOVB    @TBUFP,R5       ;YES,LOAD THE BYTE
015030  004037  017516                          JSR     R0,CALCK        ;AND CALCULATE THE NEW CHECKSUM.
015034  117777  000102  164674          22$:    MOVB    @TBUFP,@VXBUF   ;TRAMSMIT A CHARACTER
015042  023737  015142  015140                  CMP     TBUFP,TEBUF     ;AT END OF CIRCULAR BUFFER?
015050  001004                                  BNE     11$             ;NO
015052  013737  015136  015142                  MOV     TBBUF,TBUFP     ;YES, RESET IT TO START.
015060  000402                                  BR      12$             ;BY-PASS INCREMENT BUFF. POINTER
```

```
015062  005237  015142                  11$:    INC     TBUFP           ;INCREMENT BUFFER POINTER.

015066  005337  015144                  12$:    DEC     XMCNT           ;DECREMENT THE TRANSMIT COUNT
015072  001401                                  BEQ     2$              ;YES,CLEANUP,REQUEST ERRORS AND EXIT.
015074  000002                                  RTI                     ;NO, CONTINUE
015076  105737  002224                  2$:     TSTB    BLKM            ;SOM/EOM XMIT?
015102  001014                                  BNE     TXEX            ;YES-DO NOT SET XMDNE UNTIL EOM SENT.
015104  052737  000001  002222                  BIS     #XMDNE,VSTAT    ;SET THE DONE BIT IN VSTAT.
015112  042737  002000  002222                  BIC     #CKSUM,VSTAT    ;CLEAR THE CHECKSUM FLAG WHEN DONE.
015120  013737  015136  015142                  MOV     TBBUF,TBUFP     ;RESET BUFFER POINTER.
015126  042777  000100  164600          KIENA:  BIC     #TENA,@VXCSR    ;CLEAR XMIT. INT. ENABLE
015134  000002                          TXEX:   RTI


015136  000000                          TBBUF:  .WORD                   ;CONTAINS INITIAL ADDRESS
015140  000000                          TEBUF:  .WORD                   ;CONTAIN LAST ADDRESS
015142  000000                          TBUFP:  .WORD                   ;CONTAINS CURRENT LOCATION
```

```
015144  000000                          XMCNT:  .WORD   0       I05 ;LOADED WITH NUMBER OF XMITS.
                                        ;*************************************************************



                                                ;SUBROUTINE TO ISSUE RESET TO THE VT61, ENTERS MAINTENANCE MODE
                                                ;AND FORCES LINEAR ADDRESSING.
                                        ;*************************************************************

015146  113737  001102  002226          RESETV: MOVB    $TSTNM,TSTNM    ;LOAD THE TEST NUMBER IN ERROR PRINT AREA.
015154  013746  002166                          MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
015160  013746  002126                          MOV     RESET,-(SP)     ;;PUSH RESET ON STACK
015164  013746  002056                          MOV     ESCO,-(SP)      ;;PUSH ESCO ON STACK
015170  004037  013322                          JSR     RO,TESC         ;GO XIMT IT
015174  004037  015256                          JSR     RO,GETON        ;GO LOOK FOR XON.
015200  000405                                  BR      1$              ;FOUND IT.
015202  005237  002176                          INC     FTLCNT          ;ADD 1 TO FATAL XMIT COUNT.
015206  010037  001120                          MOV     RO,$GDADR       ;NO XON ISSUE XON ERROR
015212  104017                                  ERROR   17
015214                                  1$:
015214  013746  002166                          MOV     ZERO,-(SP)      ;;PUSH ZERO ON STACK
015220  013746  002002                          MOV     EMAIN,-(SP)     ;;PUSH EMAIN ON STACK
015224  013746  002056                          MOV     ESCO,-(SP)      ;;PUSH ESCO ON STACK
015230  013746  002012                          MOV     DRECT,-(SP)     ;;PUSH DRECT ON STACK
015234  013746  002056                          MOV     ESCO,-(SP)      ;;PUSH ESCO ON STACK
015240  004037  013322                  2$:     JSR     RO,TESC
015244  005037  002222                          CLR     VSTAT           ;CLEAR INT. FLAGS AFTER TERMINAL RESET
015250  005037  016746                          CLR     HDFLG           ;CLEAR PRINT HEADER FLAG.
015254  000200                                  RTS     RO

                                        ;*************************************************************
                                                ;SUBROUTINE TO WAIT FOR AN XON. NO XON EXIT IS PC +2.
                                        ;*************************************************************

015256  012737  000454  002206          GETON:  MOV     #300.,BUBCT     ;SET UP TO LOOK FOR 3 SEC.
015264  105077  013540                          CLRB    @ABUFP
015270  127727  013534  000021          1$:     CMPB    @ABUFP,#XON     ;RECEIVED A XON?
015276  001412                                  BEQ     GOTON           ;YES-EXIT.
015300  012737  000001  017074                  MOV     #1,DCOUNT       ;NO-DELAY 10 M.S.
```

```
015306  004037  017032                          JSR     RO,DELAY
015312  005337  002206                          DEC     BUBCT           ;AT END OF DELAY?
015316  001364                                  BNE     1$              ;NO
015320  062700  000002                          ADD     #2,RO           ;YES-SET UP ERROR EXIT.
015324  000200                          GOTON:  RTS     RO


                                        ;*************************************************************

                                                ;SUBROUTINE TO ISSUE ESCZ AND LOOK FOR A RESPONSE-EITHER
                                                ;A -1 OR THE RETURNED IDENT. THE -1 INDICATES NO
                                                ;RESPONSE FROM THE UNIT UNDER TEST.
                                        ;*************************************************************

015326                                  ZFLAG:
015326  012637  015364                          MOV     (SP)+,ROSV1     ;;POP STACK INTO ROSV1
015332  013746  002166                          MOV     @#ZERO,-(SP)    ;;PUSH @#ZERO ON STACK
015336  013746  002124                          MOV     @#ESCZ,-(SP)    ;;PUSH @#ESCZ ON STACK
015342  004037  013322                          JSR     RO,TESC         ;GO ISSUE ESZ SEQUENCE
015346  012746  106003                          MOV     #106003,-(SP)   ;;PUSH #106003 ON STACK
015352  004037  013060                          JSR     RO,RECTM        ;GO READ THE CHARACTER
015356  013746  015364                          MOV     ROSV1,-(SP)     ;;PUSH ROSV1 ON STACK
015362  000200                                  RTS     RO
015364  000000                          ROSV1:  .WORD   0
```

```
                                          ;*****************************************************************
                                                  ;ROUTINE TO CHECK SOFTWARE S'ATUS REGISTER (VSTAT)
                                                  ;RECEIVE FLAGS ONLY.  ENTERED WITH ANTICIPATED
                                                  ;STATUS WORD ON THE STACK.
                                          ;*****************************************************************

 015366                                   CKSFT:
 015366  012637  002220                           MOV     (SP)+,ROSVE          ;;POP STACK INTO ROSVE
 015372  010137  002162                           MOV     R1,SVER1             ;SAVE R1
 015376  010237  002164                           MOV     R2,SVER2             ;SAVE R2

 015402  012601                                   MOV     (SP)+,R1             ;;POP STACK INTO R1
 015404  013702  002222                           MOV     VSTAT,R2             ;SET R2 EQUAL TO VSTAT

 015410  042702  003576                           BIC     #003576,R2           ;CLEAR NON-ERROR BITS
 015414  020102                                   CMP     R1,R2                ;COMPARE ANTICIPATED TO ACTUAL.
 015416  001432                                   BEQ     NOER                 ;NO UNUSAL BITS EXIT

 015420  010137  001124                           MOV     R1,$GDDAT            ;MOVE GOOD STATUS TO MESSAGE
 015424  013737  002222  001126                   MOV     VSTAT,$BDDAT         ;MOVE BAD STATUS TO MESSAGE
 015432  104003                                   ERROR   3                    ;ISSUE ERROR MESSAGE.


                                          ;*****************************************************************
                                                  ;ROUTINE TO PRINT THE STATUS REGISTER IN THE FOLLOWING
                                                  ;FORMAT: STATUS BITS (XXX 000), CHARACTER TRANSFERRED (000  X X)
                                          ;*****************************************************************

 015434  012701  014640                           MOV     #STTER,R1            ;SET R1 EQUAL TO FIRST ENTRY
 015440  013702  014662                           MOV     STTEP,R2             ;SET R2 EQUAL LAST ENTRY
 015444  020102                           1$:     CMP     R1,R2                ;ARE THEY EQUAL
 015446  001416                                   BEQ     NOER                 ;YES-RESET POINTERS AND EXIT.
```

```
 015450  004037  015530                           JSR     R0,CLREG             ;CLEAR ERROR PRINT LOC.
 015454  013737  001730  001120                   MOV     VRCSR,$GDADR         ;LOAD ADDRESS
 015462  017737  164242  001124                   MOV     @VRCSR,$GDDAT        ;LOAD CSR
 015470  112137  001126                   2$:     MOVB    (R1)+,$BDDAT         ;MOVE CHARACTER AND
 015474  112137  001123                           MOVB    (R1)+,$BDADR+1       ;STATUS BITS TO ERROR REGISTERS.
 015500  104002                                   ERROR   2                    ;ISSUE ERROR MESSAGE
 015502  000760                                   BR      1$                   ;DO AGAIN
 015504  013701  002162                   NOER:   MOV     SVER1,R1             ;RESTORE R1 AND
 015510  013702  002164                           MOV     SVER2,R2             ;R2.
 015514  012737  014640  014662                   MOV     #STTER,STTEP         ;RESET STATUS ERROR POINTER.
 015522  013746  002220                           MOV     ROSVE,-(SP)          ;;PUSH ROSVE ON STACK
 015526  000200                                   RTS     R0                   ;EXIT

                                          ;*****************************************************************
                                                  ;SUBROUTINE TO CLEAR ERROR/DATA OUTPUT LOCATIONS. NEEDED
                                                  ;ONLY WHEN DISPLAYING BYTES IN WORD LOCATIONS.
                                          ;*****************************************************************

 015530  005037  001120                   CLREG:  CLR     $GDADR
 015534  005037  001122                           CLR     $BDADR
 015540  005037  001124                           CLR     $GDDAT
 015544  005037  001126                           CLR     $BDDAT
 015550  000200                                   RTS     R0
                                          ;*****************************************************************
                                                  ;SUBROUTINE TO TRANSMIT THE BUFFER AND WAIT FOR XMIT DONE
                                                  ;AND END OF RECEIVE MESSAGE. SUBROUTINE WILL LOOP IF LOCATION
                                                  ;RECITT IS PRE-LOADED WITH A NUMBER HIGHER THAN1(IE. MULTIPLE
                                                  ;RECEIVES CAN BE ACCOMPLISHED WITH ONLY ONE ENTRY TO SUB-
```

```
                                                   ;ROUTINE).WDSTOR AND BYSTOR ARE THE WORD(CURSOR POS.) AND BYTE
                                                   ;STORAGE LOCATIONS,RESPECTIVELY.DEFAULT STORAGE IS THE REC. BUFFER.

                                           ;*************************************************************

015552                                     XMREC:
015552  010546                                     MOV     R5,-(SP)            ;;PUSH R5 ON STACK
015554  012737  001001  002224                     MOV     #1001,BLKM          ;SET UP FOR A SOM/EOM TRANSMIT.
015562  042737  077577  002222                     BIC     #77577,VSTAT        ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
015570  013701  016024                             MOV     BYSTOR,R1           ;LOAD THE STORAGE POINTERS
015574  013702  016022                             MOV     WDSTOR,R2
015600  052777  000100  164126                     BIS     #TENA,@VXCSR        ;SET INTERRUPT ENABLES
015606  042737  061466  002222     XMITT:  BIC     #61466,VSTAT        ;CLEAR SOM,EOM,EPL,ESC,REV.VID., PARA. DELIM.,IDENT,CUR.
015614  005037  002216             1$:     CLR     DLAY                ;SET UP TIME OUT DELAY.
015620  032737  000001  002222             BIT     #XMDNE,VSTAT        ;IS XMIT DONE?
015626  001015                             BNE     3$                  ;YES-LOOK FOR RECEIVE DONE.
015630  032737  020000  002222     2$:     BIT     #REOM,VSTAT         ;RECEIVED AN EOM?
015636  001401                             BEQ     20$                 ;NO
015640  000435                             BR      CKSTR               ;YES-GO HANDLE DATA
015642  032737  100000  002222     20$:    BIT     #RXOFF,VSTAT        ;NO- IS XOFF SET?
015650  001761                             BEQ     1$                  ;NO-STILL TRANSMITTONG.
015652  005337  002216                     DEC     DLAY                ;YES- RUN DELAY
015656  001364                             BNE     2$                  ;WAITING FOR XON
015660  000416                             BR      XMAD2               ;NO XON-REPORT VT61 FAILURE.

015662  013705  031030             3$:     MOV     ABUFP,R5            ;LOAD CH. RECEIVED FLAG.
015666  005037  002216                     CLR     DLAY                ;SET UP RECEIVE DELAY.
015672  032737  020000  002222     4$:     BIT     #REOM,VSTAT         ;RECEIVE END OF MESSAGE?
```

```
015700  001015                             BNE     CKSTR               ;YES-CHECK DATA STORAGE POINTERS
015702  020537  031030                     CMP     R5,ABUFP            ;RECEIVED ANOTHER CHARACTER?
015706  001365                             BNE     3$                  ;YES-RESET CH. FLAG AND DELAY
015710  005337  002216             5$:     DEC     DLAY                ;RUN DELAY
015714  001366                             BNE     4$                  ;AND KEEP LOOKING FOR EOM.
015716  062700  000002             XMAD2:  ADD     #2,R0               ;TIME OUT OCCURRED-SET UP ERROR EXIT.
015722  005237  002176                     INC     FTLCNT              ;INCREMENT FATAL XMIT COUNT.
015726  004037  016136                     JSR     R0,RESPTR           ;AND REST ALL INTERRUPT POINTERS.
015732  000422                             BR      CKVST
015734  020102                     CKSTR:  CMP     R1,R2               ;STORAGE POINTERS CLEARED?
015736  001413                             BEQ     CHKITT              ;YES--LEAVE DATA IN REC. BUFFER.
015740  032737  000004  002222             BIT     #CURPOS,VSTAT       ;RECEIVED A CURSOR POSITION?
015746  001403                             BEQ     STRBYT              ;NO-GO STORE A BYTE.
015750  017722  176654                     MOV     @RBBUF,(R2)+        ;YES,STORE IT.
015754  000404                             BR      CHKITT              ;AND CHECK ITERATION COUNT.
015756  005701                     STRBYT: TST     R1                  ;STORING A CHAR?
015760  001402                             BEQ     CHKITT              ;NO
015762  117721  176642                     MOVB    @RBBUF,(R1)+        ;STORE A RECEIVED BYTE
015766  005337  016020             CHKITT: DEC     RECITT              ;DONE RECEIVING?
015772  001305                             BNE     XMITT               ;NO-LOOP SUBROUTINE
015774  004037  020560                     JSR     R0,CKOFF            ;SEE IS XOFF IS UP.
016000                             CKVST:
016000  012746  060001                     MOV     #60001,-(SP)        ;;PUSH #60001 ON STACK
016004  004037  015366                     JSR     R0,CKSFT
016010  004037  016136                     JSR     R0,RESPTR           ;RESET INTERRUPT POINTERS.
016014  012605                             MOV     (SP)+,R5            ;;POP STACK INTO R5
016016  000200                     XMXT:   RTS     R0                  ;EXIT SUBROUTINE.
016020  000000                     RECITT: .WORD   0                   ;RECEIVE ITERATION COUNT.
016022  000000                     WDSTOR: .WORD   0                   ;WORD STORAGE POINTER
016024  000000                     BYSTOR: .WORD   0                   ;BYTE STORAGE POINTER

                                   ;*************************************************************
                                           ;SUBROUTINE TO  XMIT THE BYTE AT TBUFP.
```

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

016026  042737  000001  002222  XMIT1:  BIC     #1,VSTAT        ;CLEAR XMIT DONE FLAG
016034  012737  000001  015144          MOV     #1,XMCNT        ;SET UP TO XMIT 1 BYTE
016042  052777  000100  163664          BIS     #TENA,@VXCSR
016050                          1$:
016050  012746  000001                  MOV     #XMDNE,-(SP)    ;;PUSH #XMDNE ON STACK
016054  012746  000001                  MOV     #1,-(SP)        ;;PUSH #1 ON STACK
016060  004037  020470                  JSR     R0,WTBGND       ;LOOK FOR XMIT DONE
016064  000401                          BR      FTLEXT          ;HUNG TRANSMIT-CLEAR FLAGS AND EXIT
016066  000402                          BR      NORXT           ;NORMAL EXIT.
016070  005037  002222          FTLEXT: CLR     VSTAT           ;CLEAR ANY FLAGS
016074  000200                  NORXT:  RTS     R0              ;AND EXIT

                                        ;***********************************************************
                                                ;SUBROUTINE TO ISSUE A BYTE AT A TIME UNTIL A ZERO
                                                ;BYTE IS ENCOUNTERED.
                                        ;***********************************************************

016076  112777  000002  177036  LDXMIT: MOVB    #SOM,@TBUFP     ;SEND THE START OF MESSAGE.
016104  000403                          BR      2$
016106  112377  177030          1$:     MOVB    (R3)+,@TBUFP    ;MOVE A BYTE TO XMIT BUFFER
016112  001403                          BEQ     LDOUT           ;IF A ZERO BYTE-EXIT
```

```
016114  004037  016026          2$:     JSR     R0,XMIT1        ;GO XMIT A BYTE
016120  000772                          BR      1$              ;XMIT AGAIN.
016122  112777  000004  177012  LDOUT:  MOVB    #EOM,@TBUFP     ;SEND THE END OF MESSAGE.
016130  004037  016026                  JSR     R0,XMIT1
016134  000200                          RTS     R0

                                        ;***********************************************************
                                                ;ROUTINE TO RESET ALL INTERRUPT POINTERS.
                                        ;***********************************************************

016136  042777  000100  163570  RESPTR: BIC     #TENA,@VXCSR    ;CLEAR INTERRUPT ENABLES
016144  013737  014630  014634          MOV     RBBUF,RBUFP     ;RESET RECEIVE BUF POINTER
016152  013737  015136  015142          MOV     TBBUF,TBUFP     ;RESET XMMT BUF POINTER
016160  012737  014640  014662          MOV     #STTER,STTEP    ;RESET RECEIVE STATUS ERR POINTER
016166  005037  015144                  CLR     XMCNT           ;CLEAR TRANSMIT COUNT
016172  005037  014636                  CLR     ESAMB           ;CLEAR ESC ASSEMBLY FLAGS
016176  012737  000001  016020          MOV     #1,RECITT       ;RESET REC. ITERATION COUNT
016204  005037  016022                  CLR     WDSTOR          ;CLEAR STORAGE POINTERS
016210  005037  016024                  CLR     BYSTOR
016214  000200                          RTS     R0


                                        ;***********************************************************
                                                ;SUBROUTINE TO ISSUE CURSOR POSITION ERROR. GOOD
                                                ;LINE/COLUMN MUST BE A WORD ON STACK. ERROR
                                                ;POSITION IS EXPECTED TO BE @ RBBUF.
                                        ;***********************************************************

016216                          CURER:
016216  012637  002220                  MOV     (SP)+,R0SVE     ;;POP STACK INTO R0SVE
016222  012637  002160                  MOV     (SP)+,CHRD      ;;POP STACK INTO CHRD
016226  162737  020040  002160          SUB     #20040,CHRD     ;EXTRACT MOD 40 FROM GOOD POSITION
016234  004037  015530                  JSR     R0,CLREG
016240  113737  002161  001124          MOVB    CHRD+1,$GDDAT   ;LOAD MESSAGE WITH GOOD
016246  113737  002160  001120          MOVB    CHRD,$GDADR     ;LINE AND COLUMN
016254  017737  176350  002160          MOV     @RBBUF,CHRD     ;LINE AND COLUMN.
016262  162737  020040  002160          SUB     #20040,CHRD     ;EXTRACT MOD 40 FROM BAD POSITION.
016270  113737  002161  001126          MOVB    CHRD+1,$BDDAT   ;LOAD MESSAGE WITH BAD
```

```
016276  113737  002160  001122            MOVB    CHRD,$BDADR         ;LINE AND COLUMN.
016304  104006                            ERROR   6                   ;ISSUE ERROR
016306  013746  002220                    MOV     ROSVE,-(SP)         ;;PUSH ROSVE ON STACK
016312  000200                            RTS     RO

                                  ;*************************************************************

                                  ;*************************************************************
                                          ;SUBROUTINE TO DECREMENT CUSROR POSITION IN A
                                          ;LINEAR SEQUENCE. (IE. ROW 20, COL 1 ;ROW 20 COL0 ;ROW 17, COL 157).
                                  ;*************************************************************

016314  123727  016421  000040    CMPOS:  CMPB    LNRW+1,#40          ;AT LEFT EDGE OF ROW?
016322  001403                            BEQ     1$                  ;YES, GO ADJUST COL., ROW.
016324  105337  016421                    DECB    LNRW+1              ;NO, DECREMENT COL. AND EXIT
016330  000200                            RTS     RO
016332  123727  016420  000040    1$:     CMPB    LNRW,#40            ;AT ROW 0?
016340  001405                            BEQ     2$                  ;YES, NO DECREMENT POSSIBLE-EXIT.
```

```
016342  105337  016420                    DECB    LNRW                ;NO, DECREMENT ROW AND
C16346  112737  000157  016421            MOVB    #157,LNRW+1         ;SET COL. TO RIGHT EDGE.
016354  000200                    2$:     RTS     RO

                                  ;*************************************************************
                                          ;SUBROUTINE TO INCREMENT CURSOR POSITION IN A LINEAR
                                          ;SEQUENCE (IE. ROW 10, COL 78, ROW 10. COL 79, ROW 11 , COL 0).
                                  ;*************************************************************

016356  123727  016421  000157    CPPOS:  CMPB    LNRW+1,#157         ;AT RIGHT EDGE OF ROW
016364  001403                            BEQ     1$                  ;YES, ADJUST ROW AND COLUMN.
016366  105237  016421                    INCB    LNRW+1              ;NO, INCREMENT COL. COUNT
016372  000200                            RTS     RO                  ;AND EXIT
016374  123727  016420  000067    1$:     CMPB    LNRW,#67            ;AT BOTTOM ROW?
016402  001405                            BEQ     2$                  ;YES, NO INCREMENT POSSIBLE-EXIT.
016404  105237  016420                    INCB    LNRW                ;NO, INCREMENT ROW COUNT AND
016410  112737  000040  016421            MOVB    #40,LNRW+1          ;SET COL. TO LEFT EDGE.
016416  000200                    2$:     RTS     RO

016420  000000                    LNRW:   .WORD   0                   ;CONTAINS UPDATED CURSOR POSITION.
                                  ;*************************************************************


                                          ;SUBROUTINE TO XMIT, RECEIVE AND COMPARE. DATA ERRORS
                                          ;ARE REPORTED FROM SUBROUTINE. IF THE TRANSMIT OR
                                          ;RECEIVE LOOPS 'TIME OUT', EXIT FROM SUBROUTINE WILL
                                          ;BE NORMAL EXIT +2. SUBROUTINE ENTERED WITH (R1)=
                                          ;GOOD DATA BUFFER, (R2)=RECEIVE DATA BUFFER AND
                                          ;R3=COMPARE COUNT. IF THE VT61 DOES NOT HANG,THE ROUTINE
                                          ;WILL WAIT FOR END OF REC. MESSAGE(EOM).



                                  ;*************************************************************

016422                            XRCMP:
016422  010446                            MOV     R4,-(SP)            ;;PUSH R4 ON STACK
016424  005004                            CLR     R4                  ;USE R4 A RECEIVE COUNTER.
016426  012737  001001  002224            MOV     #1001,BLKM          ;SET UP FOR A SOM/EOM TRANSMIT.
016434  042737  077577  002222            BIC     #77577,VSTAT        ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
016442  052777  000100  163264            BIS     #TENA,@VXCSR        ;SET INTERRUPT ENABLES.
016450  005037  C16746                    CLR     HDFLG               ;CLEAR ERROR 13 PRINT FLAG
016454  012705  031000                    MOV     #TCRLB+450,R5       ;R5 IS ERROR STORAGE POINTER
016460  005037  002216            1$:     CLR     DLAY                ;SET UP TIME OUT DELAY
016464  032737  000001  002222            BIT     #XMONE,VSTAT        ;XMIT DONE?
```

```
016472  001014                              BNE     XREC            ;YES-GO RECEIVE
016474  023737  014630  014634  2$:         CMP     RBBUF,RBUFP     ;HAS RECEIVE OPERATION BEGUN?
016502  103410                              BLO     XREC            ;YES-GO RECEIVE
016504  032737  100000  002222              BIT     #RXOFF,VSTAT    ;XMIT XOFF SET?
016512  001762                              BEQ     1$              ;NO-KEEP LOOKING FOR XMIT DONE?
016514  005337  002216                      DEC     DLAY            ;YES RUN DELAY AND LOOK
016520  001365                              BNE     2$              ;FOR XON OR RECEIVED CH.
016522  000432                              BR      XRERR           ;TRANSMIT TIMEOUT-SET UP ERROR EXIT

016524  005037  002216          XREC:       CLR     DLAY            ;SET UP TIME OUT DELAY
016530  020237  014634          1$:         CMP     R2,RBUFP        ;INSURE COMPARE POINTER
016534  103410                              BLO     2$              ;LESS THAN RECEIVE POINTER
016536  032737  020000  002222              BIT     #REOM,VSTAT     ;RECEIVE EOM?
```

```
016544  001030                              BNE     XREXT           ;YES-SET UP TO EXIT
016546  005337  002216                      DEC     DLAY            ;RUN TIMEOUT DELAY
016552  001416                              BEQ     XRERR           ;TIME OUT OCCURRED-ERROR EXIT
016554  000765                              BR      1$              ;RETURN TO CHECK RECEIVE COUNT
016556  005204                  2$:         INC     R4              ;ADD 1 TO RECEIVE COUNTER.
016560  122122                              CMPB    (R1)+,(R2)+     ;COMPARE CHARACTERS
016562  001407                              BEQ     4$              ;EQUAL-COMPARE AGAIN
016564  020527  031030                      CMP     R5,#TCRLB+500   ;ALLREADY STORED 50 ERRORS?
016570  103004                              BHIS    4$              ;YES-BYPASS STORAGE
016572  114125                              MOVB    -(R1),(R5)+     ;STORE GOOD DATA
016574  114225                              MOVB    -(R2),(R5)+     ;STORE BAD DATA
016576  010425                              MOV     R4,(R5)+        ;LOAD RECEIVE COUNT
016600  132122                              BITB    (R1)+,(R2)+     ;RESET POINTERS AND
016602  005303                  4$:         DEC     R3              ;CHECK COMPARE COUNT
016604  001410                              BEQ     XREXT           ;ALL DONE-EXIT
016606  000746                              BR      XREC            ;COMPARE ANOTHER
016610  062700  000002          XRERR:      ADD     #2,R0           ;SET UP ERROR EXIT
016614  005237  002176                      INC     FTLCNT          ;INCREMENT FATAL XMIT COUNT.
016620  004037  016136                      JSR     R0,RESPTR       ;RESET INTERRUPT POINTERS.
016624  000440                              BR      XROUT
016626                          XREXT:
016626  012746  020000                      MOV     #REOM,-(SP)     ;;PUSH #REOM ON STACK
016632  012746  000004                      MOV     #4,-(SP)        ;;PUSH #4 ON STACK
016636  004037  020470                      JSR     R0,WTBGND
016642  000431                              BR      XROUT           ;NO EOM-ISSUE ERROR AND EXIT.
016644  162705  031000                      SUB     #TCRLB+450,R5   ;NOW EXTRACT ERROR COUNT-IF ANY.
016650  010501                              MOV     R5,R1           ;AND STORE IT IN R1
016652  012705  031000                      MOV     #TCRLB+450,R5   ;RELOAD ERROR POINTER
016656  005701                              TST     R1              ;TEST FOR ERRORS
016660  001422                              BEQ     XROUT           ;NO-CHECK STATUS AND EXIT
016662  005737  016746                      TST     HDFLG           ;;DATA ERROR HEADER PRINTED?
016666  001003                              BNE     1$              ;YES-BYPASS HEADER PRINT
016670  104012                              ERROR   12              ;PRINT DATA ERROR HEADER
016672  005237  016746                      INC     HDFLG           ;SET HEADER PRINT FLAG
016676  004037  015530          1$:         JSR     R0,CLREG        ;ERROR WAS LEGTIMATE. LOAD
016702  112537  001124                      MOVB    (R5)+,$GDDAT    ;ERROR MESSAGE AND ISSUE
016706  112537  001126                      MOVB    (R5)+,$BDDAT    ;IT.
016712  012537  001120                      MOV     (R5)+,$GDADR    ;LOAD RECEIVE COUNT
016716  104004                              ERROR   4               ;ISSUE DATA COMPARE ERROR
016720  162701  000004                      SUB     #4,R1           ;DECREMENT ERROR COUNT
016724  001364                              BNE     1$              ;PRINT ANOTHER IF NOT AT ZERO
016726  004037  020560          XROUT:      JSR     R0,CKOFF        ;SEE IS XOFF IS UP
016732  012746  060001                      MOV     #60001,-(SP)    ;;PUSH #60001 ON STACK
016736  004037  015366                      JSR     R0,CKSFT        ;CHECK FOR VSTAT /STATUS ERR.
016742  012604                              MOV     (SP)+,R4        ;;POP STACK INTO R4
016744  000200                              RTS     R0              ;EXIT SUBROUTINE

016746  000000                  HDFLG:      0                       ;INHIBIT PRINT FLAG.
```

B06

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

                                          ;SUBROUTINE TO CREATE A 'RULER' IN LOCATIONS 200
                                          ;TO 317.

                                  ;*****************************************************************

```

```
 016750  012701  030530           CRRUL:  MOV     #TCRLB+200,R1     ;LOAD STARTING ADDRESS
 016754  012702  130461                   MOV     #130461,R2        ;LOAD INITIAL RULER ASCII CODES.
 016760  110221                   1$:     MOVB    R2,(R1)+          ;STORE A RULER BYTE IN XMIT BUF.
 016762  022701  030650                   CMP     #TCRLB+320,R1     ;RULER COMPLETE?
 016766  103001                           BHIS    2$                ;NO
 016770  000200                           RTS     R0                ;AND EXIT.
 016772  105202                   2$:     INCB    R2                ;INCREMENT ASCII BYTE
 016774  122702  000272                   CMPB    #272,R2           ;END OF REVERSE VIDEO?
 017000  001003                           BNE     3$                ;NO-SEE IF END OF NORMAL.
 017002  012702  030660                   MOV     #030660,R2        ;SET UP TO ISSUE REVERSE 0.
 017006  000405                           BR      5$
 017010  122702  000072           3$:     CMPB    #72,R2            ;END OF NORMAL VIDEO?
 017014  001361                           BNE     1$                ;NOT AT END OF A VIDEO STRING.
 017016  012702  130460                   MOV     #130460,R2                ;YES-SET UP TO ISSUE NORMAL 0.
 017022  110221                   5$:     MOVB    R2,(R1)+          ;DO IT
 017024  105202                           INCB    R2                ;SET BYTE TO NEXT ASCII CODE
 017026  000302                           SWAB    R2                ;REVERSE VIDEO MODE.
 017030  000753                           BR      1$                ;BEGIN NEXT STRING

                                  ;*****************************************************************
                                          ;SUBROUTINE TO DELAY 10 M.S. TIME THE NUMBER INLOCATION
                                          ;DCOUNT. THE PROCESSOR TYPE PRE-DETERMINES THE # OF LOOPS
                                          ;REQUIRED TO DELAY 10 M.S. FOR ONE ITERATION. LOCATION
                                          ;PMULT IS PRE-LOADED WITH : 11/45 = 4, 11/40 = 2
                                          ;AND 11/10 =1.
                                  ;*****************************************************************

 017032                           DELAY:
 017032  010146                           MOV     R1,-(SP)          ;;PUSH R1 ON STACK
 017034  010246                           MOV     R2,-(SP)          ;;PUSH R2 ON STACK
 017036  013702  017072           1$:     MOV     PMULT,R2          ;LOAD PROCESSOR MULTIPLIER
 017042  012701  002570           2$:     MOV     #1400.,R1         ;LOAD 10 M.S. DELAY
 017046  005301                           DEC     R1                ;RUN BASIC DELAY
 017050  001376                           BNE     .-2
 017052  005302                           DEC     R2                ;RUN MULTIPLIER DELAY
 017054  001372                           BNE     2$
 017056  005337  017074                   DEC     DCOUNT            ;RUN ITERATION COUNT
 017062  001365                           BNE     1$
 017064  012602                           MOV     (SP)+,R2          ;;POP STACK INTO R2
 017066  012601                           MOV     (SP)+,R1          ;;POP STACK INTO R1
 017070  000200                           RTS     R0

 017072  000000                   PMULT:  0                         ;PROCESSOR MULTIPLIER
 017074  000000                   DCOUNT: 0                         ;ITERATION COUNT

                                  ;*****************************************************************

                                          ;SUBROUTINE TO GENERATE A INCREMENTING PATTERN AT
                                          ;(R1)+.  ENTER WITH R3 EQUAL TO # OF CH. TO CREATE.
                                          ;R5 IS UTILIZED AS A WORK REGISTER.

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

```
017076  012705  000041          BLDINC: MOV     #41,R5                  ;LOAD R5 WITH INITIAL CH.
017102  110521                  BLDINA: MOVB    R5,(R1)+                ;MOVE A CH. TO BUFFER
017104  005303                          DEC     R3                      ;DECREMENT BYTE COUNT
```

```
017106  001001                          BNE     2$                      ;NOT DONE-UPDATE PATTERN
017110  000200                          RTS     R0                      ;EXIT-DONE.
017112  105205                  2$:     INCB    R5                      ;UPDATE CH. PATTERN
017114  122705  000177                  CMPB    #177,R5                 ;PATTERN EXCEEDED MAX?
017120  001766                          BEQ     BLDINC                  ;YES-RESET IT.
017122  000767                          BR      BLDINA                  ;NO-ISSUE CURRENT PATTERN.


                                ;*****************************************************************

                                        ;SUBROUTINE TO FILL THE SCREEN WITH INCREMENTING DATA
                                ;*****************************************************************

017124  042737  077577  002222  DATSC:  BIC     #77577,VSTAT            ;CLEAR INTERRUPT FLAGS.
017132  013701  015136                  MOV     TBBUF,R1
017136  012703  000500                  MOV     #320.,R3                ;FILL XMIT BUFFER WITH INCRE-
017142  004037  017076                  JSR     R0,BLDINC               ;MENTING PATTERN
017146  012737  003600  015144  10$:    MOV     #TOTCH,XMCNT            ;SET UP TO XMIT 1920 BYTES
017154  052777  000100  162552          BIS     #TENA,@VXCSR

017162  032737  000001  002222  1$:     BIT     #XMDONE,VSTAT           ;XMIT DONE?
017170  001774                          BEQ     .-6                     ;NO

                                ;*****************************************************************

                                        ;SUBROUTINE TO RESET VT61 AND DISPLAY MESSAGE
                                        ;POINTED TO BY R2.

                                ;*****************************************************************

017172  004037  015146          DSMES:  JSR     R0,RESETV               ;RESET THE UNIT AND WAIT FOR XON.
017176  042737  077577  002222          BIC     #77577,VSTAT            ;CLEAR ALL FLAGS EXCEPT XOFF AND XMKIL.
017204  012737  000005  015144          MOV     #5,XMCNT                ;PRE-LOAD XMIT COUNT.
017212  013701  015136                  MOV     TBBUF,R1                ;LOAD XMIT BUFFER WITH:
017216  012721  000002                  MOV     #SOM,(R1)+              ;START OF MESSAGE
017222  013721  002056                  MOV     ESCO,(R1)+
017226  013721  002012                  MOV     DRECT,(R1)+             ;DISABLE RECTANGULAR MODE
017232  005237  015144          1$:     INC     XMCNT                   ;INCREMENT TRANSMIT COUNT
017236  112221                          MOVB    (R2)+,(R1)+             ;DISPLAY MESSAGE
017240  001374                          BNE     1$
017242  112711  000004                  MOVB    #EOM,(R1)               ;TERMINATE WITH END OF MESSAGE.
017246  052777  000100  162460          BIS     #TENA,@VXCSR            ;XMIT IT AND WAIT FOR
017254  032737  000001  002222  2$:     BIT     #XMDONE,VSTAT           ;DONE
017262  001774                          BEQ     2$
017264  000200                          RTS     R0

                                ;*****************************************************************

                                        ;SUBROUTINE TO CONVERT A BINARY CHARACTER
                                        ;TO 3 OCTAL CHARACTERS. R1 CONTAINS BINARY
                                        ;NUMBER. RESULT IS STORED IN LOCATIONS SVER1,
                                        ;SVER2

                                ;*****************************************************************
017266                          BINOCT:
017266  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
```

```
017270  012705  000002                  MOV     #2,R5             ;LOAD ITERATION COUNT
017274  000403                          BR      2$                ;BYPASSS SHIFTS FOR 1ST CONVERSION
017276  106201                  1$:     ASRB    R1
017300  106201                          ASRB    R1                ;SHIFT A CHAR INTO POSITION
017302  106201                          ASRB    R1
017304  110165  002162          2$:     MOVB    R1,SVER1(R5)      ;STORE THE BINARY OFFSET
017310  142765  000370  002162          BICB    #370,SVER1(R5)    ;CLEAR NON ESSENTIAL BITS
017316  152765  000060  002162          BISB    #60,SVER1(R5)     ;CONVERT OFFSET TO OCTAL
017324  005305                          DEC     R5                ;DECREMENT CONVERSION COUNT
017326  100363                          BPL     1$                ;NOT DONE CONVERT ANOTHER
017330  112737  000040  002165          MOVB    #40,SVER2+1       ;LOAD A SPACE
017336  012605                          MOV     (SP)+,R5          ;;POP STACK INTO R5
017340  000200                          RTS     R0

                                ;*****************************************************************
                                        ;SUBROUTINE TO CONVERT AN OCTAL CHAR. TO BINARY.  REG
                                        ;R1 CONTAINS OCTAL AND REG R2 IS BINARY ASSEMBLY AREA.
                                ;*****************************************************************

017342  042701  177770          OCTBIN: BIC     #177770,R1        ;EXTRACT OCTAL COMPONENT
017346  005702                          TST     R2                ;FIRST CONVERSION?
017350  001403                          BEQ     NOSHFT            ;YES - DO NOT SHIFT
017352  006302                          ASL     R2                ;NO - SHIFT PREVIOUS CHAR.
017354  006302                          ASL     R2
017356  006302                          ASL     R2
017360  060102                  NOSHFT: ADD     R1,R2             ;ADD CURRENT CHAR.
017362  000200                          RTS     R0

                                ;*****************************************************************
                                        ;ROUTINE TO WAIT FOR C/R FROM VT61 UNDER TEST
                                ;*****************************************************************

017364  032777  000200  162336  GTCR:   BIT     #RECDN,@VRCSR     ;WAIT FOR REVEIVE DONE
017372  001774                          BEQ     .-6
017374  127737  162332  001750          CMPB    @VRBUF,CARRT      ;CHAR = CARRIAGE RETURN?
017402  001370                          BNE     GTCR              ;NO-KEEP LOOKING
017404  000200                          RTS     R0                ;YES-EXIT
```

```
                                                ;*******************************************************************
                                                        ;SUBROUTINE TO GET A CHARACTER (NUMERIC) FROM THE
                                                        ;CONSOLE.  IF OTHER THAN A NUMERIC IS TYPED A
                                                        ;"?" WILL BE ECHOED.
                                                ;*******************************************************************

 017406  004037  013442                 GTNUM:  JSR     R0,CONRD        ;GET A CHAR
 017412  012601                                 MOV     (SP)+,R1        ;;POP STACK INTO R1
 017414  122701  000054                         CMPB    #54,R1          ;CHAR. =COMMA?
 017420  001411                                 BEQ     1$              ;YES-GO PRINT IT
 017422  123701  001750                         CMPB    CARRT,R1        ;CHAR. = CARRIAGE RETURN?
 017426  001406                                 BEQ     1$
 017430  120127  000060                         CMPB    R1,#60
 017434  103421                                 BLO     QUST            ;IF CHAR. IS LESS THAN 60
 017436  120127  000067                         CMPB    R1,#67          ;OR MORE THAN 67, TYPE
 017442  101016                                 BHI     QUST            ;A QUESTION MARK
 017444  110137  017514                 1$:     MOVB    R1,TYPNUM
```

```
 017450  104400  017514                         TYPE    ,TYPNUM
 017454  123701  001752                         CMPB    LNFED,R1
 017460  001406                                 BEQ     GTEXT
 017462  123701  001750                         CMPB    CARRT,R1        ;IF CHAR. - C/R SET UP TO ISSUE
 017466  001003                                 BNE     GTEXT           ;LINE FEED BEFORE EXITING.
 017470  113701  001752                         MOVB    LNFED,R1
 017474  000763                                 BR      1$
 017476  000200                         GTEXT:  RTS     R0              ;GOOD CHAR., EXIT
 017500  112737  000077  017514         QUST:   MOVB    #77,TYPNUM
 017506  104400  017514                         TYPE    ,TYPNUM         ;TYPE QUESTION MARK AND
 017512  000735                                 BR      GTNUM           ;KEEP LOOKING.
 017514     000                         TYPNUM: .BYTE   0
 017515     000                                 .BYTE   0


                                                ;*******************************************************************
                                                        ;SUBROUTINE TO CALCULATE CHECKSUM ON THE LOWER
                                                        ;BYTE OF R5.  R4 IS STORAGE FOR THE CHECKSUM
                                                        ;CHARACTER. ALGORITHM FOR CHECKSUM IS ROTATE
                                                        ;CURRENT ONE PLACE LEFT AND XOR NEW CHAR. CHECKSUM
                                                        ;IS THE LOWER 7 BITS OF R4

                                                ;*******************************************************************

 017516  042705  177400                 CALCK:  BIC     #177400,R5      ;CLEAR UPPER BYTE OF R5
 017522  120527  000021                         CMPB    R5,#XON         ;CHAR. =XON?
 017526  001415                                 BEQ     NOCALC          ;YES DO NOT CALCULATE CHECKSUM
 017530  120527  000023                         CMPB    R5,#XOFF        ;CHAR =XOFF?
 017534  001412                                 BEQ     NOCALC          ;YES DO NOT CALCULATE CHECKSUM

 017536  000241                                 CLC                     ;INSURE CARRY BIT INITIALLLY CLEAR
 017540  105704                                 TSTB    R4              ;SET UP TO ROTATE R4
 017542  100001                                 BPL     1$              ;A FULL 8 BYTES

 017544  000261                                 SEC                     ;R4 WAS NEG. SO ROTATE A ONE
 017546  106104                         1$:     ROLB    R4              ;INTO LOW ORDER BIT.
 017550  010403                                 MOV     R4,R3
 017552  040503                                 BIC     R5,R3           ;NOT A AND B
 017554  040405                                 BIC     R4,R5           ;NOT B AND A
 017556  050305                                 BIS     R3,R5           ;ORED
 017560  010504                                 MOV     R5,R4           ;EQUAL NEW CHECKSUM
 017562  000200                         NOCALC: RTS     R0
                                                ;*******************************************************************
```

```
                                                        ;SUBROUTINE TO LOAD XMIT BUFFER FROM R0 THRU R1
                                              ;************************************************************

017564  112021                                LDBUF:  MOVB    (R0)+,(R1)+             ;LOAD A BYTE
017566  001376                                        BNE     .-2                     ; UNTIL ZERO BYTE FOUND.
017570  000200                                        RTS     R0
                                              ;************************************************************
                                                        ;SUBROUTINE TO CHECK THE VT61 FOR A PERIPHERAL ABORT.
                                              ;************************************************************

017572  032737  010000  002222                CKABRT: BIT     #PABRT,VSTAT            ;ABORT FLAG RECEIVED?
017600  001445                                        BEQ     2$                      ;NO-EXIT
```

```
017602  010037  001124                                MOV     R0,$GDDAT
017606  162737  000004  001124                        SUB     #4,$GDDAT               ;POINT ERR PC TO MAIN ROUTINE.
017614  013737  002222  001126                        MOV     VSTAT,$BDDAT
017622  104020                                        ERROR   20                      ;ISSUE PERIPHERAL ABORT ERROR

017624  013701  015136                                MOV     TBBUF,R1
017630  004037  017564                                JSR     R0,LDBUF                ;LOAD THE XMIT BUFFER WITH:
017634     033     117     137                        .BYTE   .ESC,.O,.IABT,.ESC,.O,.RABT
017637     033     117     140
017642     033     117     145                        .BYTE   .ESC,.O,.UNLKKB,0
017645     000
017646  012737  000011  015144                        MOV     #9.,XMCNT               ;SET UP TO XMIT 9 BYTES.
017654  004037  015552                                JSR     R0,XMREC                ;XMIT AND RECEIVE.
017660  000240                                        NOP
017662  123727  014664  000170                        CMPB    STR0,#NABRT             ;ABORT FLAG CLEARED?
017670  001411                                        BEQ     2$                      ;YES-EXIT
017672  010037  001124                                MOV     R0,$GDDAT               ;NO-SET UP AND ISSUE A CANT
017676  162737  000004  001124                        SUB     #4,$GDDAT               ;CLEAR ABORT FLAG ERROR MESSAGE.
017704  013737  002222  001126                        MOV     VSTAT,$BDDAT
017712  104021                                        ERROR   21
017714  000200                                2$:     RTS     R0

                                              ;************************************************************
                                                        ;SUBROUTINE TO COMPARE RECEIVED KEYBOARD POSITION WITH
                                                        ;EXPECTED KEYBOARD POSITION. ERRORS ARE REPORTED
                                                        ;AS POSITIONAL ERRORS AND NOT DATA COMPARE ERRORS.
                                              ;************************************************************


017716  105077  011106                        CKKBD:  CLRB    @ABUFP                  ;CLEAR RECEIVE BYTE
017722  005037  002160                                CLR     CHRD                    ;CLEAR INPUT STORAGE.
017726  105777  011076                        KBDLP:  TSTB    @ABUFP                  ;WAIT FOR A INPUT.
017732  001775                                        BEQ     .-4

017734  117737  011070  002160                        MOVB    @ABUFP  ,CHRD           ;STORE IT AND
017742  105077  011062                                CLRB    @ABUFP                  ;CLEAR THE INPUT AREA.
017746  123714  002160                        1$:     CMPB    CHRD    ,(R4)           ;RECEIVED EQUAL EXPECTED?
017752  001500                                        BEQ     GDSTRK                  ;NO-UPDATE POINTERS.
017754  005237  002206                                INC     BUBCT                   ;INCREMENT ERROR COUNT.
017760  023727  002206  000012                        CMP     BUBCT,#10.              ;COUNT = 10?
017766  103075                                        BHIS    CNTF                    ;YES-EXIT SUBROUTINE.
017770  010401                                        MOV     R4,R1
017772  166501  011560                                SUB     DTTBL(R5),R1            ;EXTRACT KEY POSITION FROM ROW LOC.
017776  005201                                        INC     R1                      ;CONVERT LOGICAL POS  TO ACTUAL.
020000  004037  017266                                JSR     R0,BINOCT               ;GET KEY POSITION IN OCTAL.
020004  113737  002164  002162                        MOVB    SVER2,SVER1             ;RE-ASSEMBLE OCTAL BYTES.
020012  123727  002163  000060                        CMPB    SVER1+1,#60             ;POSITION LESS THAN 8?
020020  001413                                        BEQ     LDPOS                   ;YES-GO LOAD IT.
```

```
020022  123727  002162  000062          CMPB    SVER1,#62       ;POSITION GREATER THAN 8 AND LESS THAN12?
020030  103404                          BLO     BOROW           ;YES-SET UP TO BORROW.
020032  162737  000002  002162          SUB     #2,SVER1        ;NO-JUST SUBTRACT 2.
020040  000403                          BR      LDPOS
020042  162737  000370  002162  BOROW:  SUB     #370,SVER1      ;SUBTRACT AND BORROW.
020050  113737  002162  027441  LDPOS:  MOVB    SVER1,KYSTRK+1  ;LOAD THE CONVERTED DECIMAL #.
020056  113737  002163  027440          MOVB    SVER1+1,KYSTRK
020064  012703  027363          DMPOCT: MOV     #DKBERR ,R3
```

```
020070  004037  016076                  JSR     R0      ,LDXMIT ;ISSUE BODY OF KEYBOARD ERROR.
020074  111401                          MOVB    (R4)    ,R1
020076  004037  017266                  JSR     R0      ,BINOCT
020102  012703  002162                  MOV     #SVER1  ,R3
020106  004037  016076                  JSR     R0      ,LDXMIT ;CONVERT AND ISSUE GOOD CHAR.
020112  012703  027472                  MOV     #DSPC6,R3
020116  004037  016076                  JSR     R0,LDXMIT       ;INSERT 6 SPACES IN MESSAGE.
020122  113701  002160                  MOVB    CHRD    ,R1
020126  004037  017266                  JSR     R0      ,BINOCT
020132  012703  002162                  MOV     #SVER1  ,R3
020136  004037  016076                  JSR     R0      ,LDXMIT ;CONVERT AND ISSUE RECEIVED CHAR.
020142  012703  001167                  MOV     #$CRLF  ,R3
020146  004037  016076                  JSR     R0      ,LDXMIT ;ISSUE C/R AND L/F.
020152  000665                          BR      KBDLP           ;LOOK FOR SAME KEY AGAIN.

020154  005204                  GDSTRK: INC     R4              ;INCREMENT KEYBOARD ROW COUNTER.
020156  105714                          TSTB    (R4)            ;REACHED END OF ROW?
020160  001262                          BNE     KBDLP           ;NO-LOOK FOR NEXT INPUT
020162  000200                  CNTF:   RTS     R0              ;YES-EXIT.

                                ;**************************************************************

                                        ;SUBROUTINE TO LOOP DATA THROUGH HOST COMPUTER. ALL
                                        ;FUNCTIONS ARE ALLOWED, BUT BLOCK TRANSMITS WHICH
                                        ;EXCEED 552 BYTES WILL RESULT IN THE TERMINATION
                                        ;OF THE OPERATION AFTER 552 RECEIVED BYTES.

                                ;**************************************************************

020164  005237  020466          LOOP:   INC     XMZER           ;SET UP TO XMIT NULLS.
020170  012737  031030  014632          MOV     #TCRLB+500,REBUF        ;RESET BUFFER POINTERS
020176  012737  027630  015136          MOV     #RCRLB,TBBUF
020204  004037  016136                  JSR     R0,RESPTR       ;RELOAD ALL INTERRUPT POINTERS
020210  042737  077577  002222          BIC     #77577,VSTAT    ;CLEAR ALL FLAGS BUT XOFF AND XMKIL.
020216  013704  014634          LOOPT:  MOV     RBUFP,R4        ;SET UP RECEIVE FLAG
020222  032737  000001  002222  LOOPTA: BIT     #XMDNE,VSTAT    ;XMIT COMPLETE?
020230  001407                          BEQ     LOOPR           ;NO
020232  042737  000001  002222          BIC     #XMDNE,VSTAT    ;YES RESET FLAG
020240  013737  014630  014634          MOV     RBBUF,RBUFP     ;RESET THE REC. BUFFER POINTER
020246  000763                          BR      LOOPT
020250  032737  001400  002222  LOOPR:  BIT     #EPL+ESC,VSTAT  ;RECEIVED AN ESC OR EPL?
020256  001004                          BNE     LPSTR           ;YES-GO CHECK IT
020260  023704  014634                  CMP     RBUFP,R4        ;RECEIVED A DISPLAY CHAR?
020264  001756                          BEQ     LOOPTA          ;NO-LOOP
020266  000426                          BR      BUMPCT
020270  117777  010534  174336  LPSTR:  MOVB    @RBUFP,@RBUFP   ;YES LOAD IT IN THE BUFFER
020276  005237  014634                  INC     RBUFP           ;AND INCREMENT BUFFER POINTER
020302  005037  014636                  CLR     ESAMB           ;CLEAR ESC ASSEMBLY WORD
020306  042737  001400  002222          BIC     #EPL+ESC,VSTAT  ;CLEAR THE FLAGS
020314  005237  015144                  INC     XMCNT           ;INCREMENT XMIT COUNT
020320  123777  002130  010502          CMPB    ESCN,@RBUFP     ;CHAR. A ESC(033)?
020326  001733                          BEQ     LOOPT           ;YES WAIT FOR NEXT PART OF FUNCTION
020330  113777  001752  174276          MOVB    LNFED,@RBUFP    ;CHAR. WAS EPL ADD A LINE FEED.
```

```
 020336  005237  014634                          INC     RBUFP
 020342  000407                                  BR      FRCECT                  ;AND ISSUE THEM.
 020344  023727  015144  000764         BUMPCT:  CMP     XMCNT,#500.             ;BUFFER ABOUT FILLED?
```

```
 020352  103403                                  BLO     FRCECT                  ;NO
 020354  005337  014634                 1$:      DEC     RBUFP                   ;YES-RESET THE RECEIVE POINTER
 020360  000716                                  BR      LOOPT
 020362  005237  015144                 FRCECT:  INC     XMCNT                   ;INCREMENT THE XMIT COUNT
 020366  023727  015144  000002                  CMP     XMCNT,#2                ;FIRST CHAR TO XMIT?
 020374  101003                                  BHI     XMWT                    ;NO
 020376  052777  000100  161330                  BIS     #TENA,@VXCSR            ;YES-SET THE XMIT ENABLE
 020404  004037  020414                 XMWT:    JSR     R0,EXTST                ;LOOK FOR END OF TEST COMMAND.
 020410  000702                                  BR      LOOPT                   ;NONE FOUND.
 020412  000200                                  RTS     R0                      ;AND EXIT

                                        ;************************************************************
                                                 ;SUBROUTINE TO CHECK FOR END OF TEST COMMAND. THE CONTROL
                                                 ;C KEY EXITS ALL TESTS EXCEPT THE BLOCK MODE TEST
                                                 ;WHICH IS EXITED ON A @ KEY.
                                        ;************************************************************

 020414  127727  010410  000003         EXTST:   CMPB    @ABUFP,#3               ;LOOK FOR CONTROL C.
 020422  001020                                  BNE     NOROUT

 020424  012737  030327  014632         ABSXT:   MOV     #RCRLB+477,REBUF                ;RESET THE BUFFERS
 020432  012737  030330  015136                  MOV     #TCRLB,TBBUF
 020440  004037  016136                          JSR     R0,RESPTR               ;RESET ALL POINTERS
 020444  012702  026253                          MOV     #DEXT,R2
 020450  004037  017172                          JSR     R0,DSMES                ;ISSUE EXIT MESSAGE
 020454  005037  020466                          CLR     XMZER                   ;CLEAR THE ZERO TRANSMIT FLAG.
 020460  062700  000002                          ADD     #2,R0                   ;SET UP TEST EXIT.
 020464  000200                         NOROUT:  RTS     R0                      ;EXIT SUBROUTINE.

 020466  000000                         XMZER:   .WORD   0
                                        ;************************************************************
                                                 ;SUB-ROUTINE TO LOOK FOR VSTAT BIT ON THE STACK
                                                 ;DELAY FACTOR IS FIRST WORD ON THE STACK AND VSTAT BIT
                                                 ;IS THE SECOND. MIN. DELAY IS 4 U.S FOR A MOS 11/45.
                                        ;************************************************************

 020470                                 WTBGND:
 020470  012637  002220                          MOV     (SP)+,R0SVE             ;;POP STACK INTO R0SVE
 020474  012637  020556                          MOV     (SP)+,VDLAY             ;;POP STACK INTO VDLAY
 020500  012637  020554                          MOV     (SP)+,VBIT              ;;POP STACK INTO VBIT
 020504  005037  002216                 1$:      CLR     DLAY
 020510  033737  020554  002222         2$:      BIT     VBIT,VSTAT              ;SENSED THE CONDITION?
 020516  001012                                  BNE     FNDBT                   ;YES-EXIT.
 020520  005337  002216                          DEC     DLAY                    ;NO-RUN DELAY.
 020524  001371                                  BNE     2$
 020526  005337  020556                          DEC     VDLAY                   ;DELAY FACTOR EXPIRED?
 020532  001364                                  BNE     1$                      ;NO-LOOP
 020534  104011                                  ERROR   11                      ;DELAY EXPIRED-ISSUE HUNG NIT
 020536  005237  002176                          INC     FTLCNT                  ;INCREMENT FATAL XMIT COUNT.
 020542  000401                                  BR      TIMEXT
 020544  005720                         FNDBT:   TST     (R0)+                   ;SET UP FOR NORMAL EXIT
 020546                                 TIMEXT:
 020546  013746  002220                          MOV     R0SVE,-(SP)             ;;PUSH R0SVE ON STACK
 020552  000200                                  RTS     R0
 020554  000000                         VBIT:    0
 020556  000000                         VDLAY:   0
```

```
                                          ;****************************************************************
                                                  ;SUBROUTINE TO LOOK FOR XOFF BEFORE EXITING A RECEIVE ROUTINE.
                                          ;****************************************************************

020560  005037  002216                    CKOFF:  CLR     DLAY
020564  032737  100000  002222            1$:     BIT     #RXOFF,VSTAT     ;IS XOFF SET?
020572  001403                                    BEQ     2$               ;NO-EXIT
020574  005337  002216                            DEC     DLAY             ;RUN DELAY.
020600  001371                                    BNE     1$
020602  000200                            2$:     RTS     R0

                                          ;****************************************************************

                                          .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
                                          ;*SW09=1          LOOP ON ERROR
                                          ;*SW08=1          LOOP ON TEST IN SWR<7:0>
                                          ;*CALL
                                          ;*       SCOPE            ;;SCOPE=IOT

020604                                    $SCOPE:
020604  004037  013474                            JSR     R0,MONIT
020610  032777  040000  160320            1$:     BIT     #BIT14,@SWR      ;;LOOP ON PRESENT TEST?
020616  001111                                    BNE     $OVER            ;;YES IF SW14=1
                                          ;#####START OF CODE FOR THE XOR TESTER#####
020620  000416                            $XTSTR: BR      6$               ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                           ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
020622  013746  000004                            MOV     @#ERRVEC,-(SP)   ;;SAVE THE CONTENTS OF THE ERROR VECTOR
020626  012737  020646  000004                    MOV     #5$,@#ERRVEC     ;;SET FOR TIMEOUT
020634  005737  177060                            TST     @#177060         ;;TIME OUT ON XOR?
020640  012637  000004                            MOV     (SP)+,@#ERRVEC   ;;RESTORE THE ERROR VECTOR
020644  000463                                    BR      $SVLAD           ;;GO TO THE NEXT TEST
020646  022626                            5$:     CMP     (SP)+,(SP)+      ;;CLEAR THE STACK AFTER A TIME OUT
020650  012637  000004                            MOV     (SP)+,@#ERRVEC   ;;RESTORE THE ERROR VECTOR
020654  000423                                    BR      7$               ;;LOOP ON THE PRESENT TEST
020656                                    6$:;#####END OF CODE FOR THE XOR TESTER#####
020656  032777  000400  160252                    BIT     #BIT08,@SWR      ;;LOOP ON SPEC. TEST?
020664  001404                                    BEQ     2$               ;;BR IF NO
020666  127737  160244  001102                    CMPB    @SWR,$TSTNM      ;;ON THE RIGHT TEST?    SWR<7:0>
020674  001462                                    BEQ     $OVER            ;;BR IF YES
020676  105737  001103                    2$:     TSTB    $ERFLG           ;;HAS AN ERROR OCCURRED?
020702  001421                                    BEQ     3$               ;;BR IF NO
020704  123737  001115  001103                    CMPB    $ERMAX,$ERFLG    ;;MAX. ERRORS FOR THIS TEST OCCURRED?
020712  101015                                    BHI     3$               ;;BR IF NO
020714  032777  001000  160214                    BIT     #BIT09,@SWR      ;;LOOP ON ERROR?
020722  001404                                    BEQ     4$               ;;BR IF NO
020724  013737  001110  001106            7$:     MOV     $LPERR,$LPADR    ;;SET LOOP ADDRESS TO LAST SCOPE
020732  000443                                    BR      $OVER
020734  105037  001103                    4$:     CLRB    $ERFLG           ;;ZERO THE ERROR FLAG
020740  005037  001156                            CLR     $TIMES           ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
```

```
020744  000415                                    BR      1$               ;;ESCAPE TO THE NEXT TEST
020746  032777  004000  160162            3$:     BIT     #BIT11,@SWR      ;;INHIBIT ITERATIONS?
020754  001011                                    BNE     1$               ;;BR IF YES
020756  005737  001100                            TST     $PASS            ;;IF FIRST PASS OF PROGRAM
```

```
020762  001406                              BEQ     1$                ;;      INHIBIT ITERATIONS
020764  005237  001104                      INC     $ICNT             ;;INCREMENT ITERATION COUNT
020770  023737  001156  001104              CMP     $TIMES,$ICNT      ;;CHECK THE NUMBER OF ITERATIONS MADE
020776  002021                              BGE     $OVER             ;;BR IF MORE ITERATION REQUIRED
021000  012737  000001  001104  1$:         MOV     #1,$ICNT          ;;REINITIALIZE THE ITERATION COUNTER
021006  013737  021056  001156              MOV     $MXCNT,$TIMES     ;;SET NUMBER OF ITERATIONS TO DO
021014  105237  001102          $SVLAD:     INCB    $TSTNM            ;;COUNT TEST NUMBERS
021020  011637  001106                      MOV     (SP),$LPADR       ;;SAVE SCOPE LOOP ADDRESS
021024  011637  001110                      MOV     (SP),$LPERR       ;;SAVE ERROR LOOP ADDRESS
021030  005037  001160                      CLR     $ESCAPE           ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
021034  112737  000001  001115              MOVB    #1,$ERMAX         ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
021042  013777  001102  160070  $OVER:      MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER
021050  013716  001106                      MOV     $LPADR,(SP)       ;;FUDGE RETURN ADDRESS
021054  000002                              RTI                       ;;FIXES PS
021056  000005                  $MXCNT:     5                         ;;MAX. NUMBER OF ITERATIONS
                                ;*************************************************************

                                .SBTTL  ERROR HANDLER ROUTINE

                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;*AND GO TO $ERRTYP ON ERROR
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1        HALT ON ERROR
                                ;*SW13=1        INHIBIT ERROR TYPEOUTS
                                ;*SW10=1        BELL ON ERROR
                                ;*SW09=1        LOOP ON ERROR
                                ;*CALL
                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

021060                          $ERROR:
021060  105237  001103          7$:         INCB    $ERFLG            ;;SET THE ERROR FLAG
021064  001775                              BEQ     7$                ;;DON'T LET THE FLAG GO TO ZERO
021066  013777  001102  160044              MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER AND ERROR FLAG
021074  032777  002000  160034              BIT     #BIT10,@SWR       ;;BELL ON ERROR?
021102  001402                              BEQ     1$                ;;NO - SKIP
021104  104400  001162                      TYPE    ,$BELL            ;;RING BELL
021110  005237  001112          1$:         INC     $ERTTL            ;;COUNT THE NUMBER OF ERRORS
021114  011637  001116                      MOV     (SP),$ERRPC       ;;GET ADDRESS OF ERROR INSTRUCTION
021120  162737  000002  001116              SUB     #2,$ERRPC
021126  117737  157764  001114              MOVB    @$ERRPC,$ITEMB    ;;STRIP AND SAVE THE ERROR ITEM CODE
021134  032777  020000  157774              BIT     #BIT13,@SWR       ;;SKIP TYPEOUT IF SET
021142  001004                              BNE     20$               ;;SKIP TYPEOUTS
021144  004737  021450                      JSR     PC,$ERRTYP        ;;GO TO USER ERROR ROUTINE
021150  104400  001167                      TYPE    ,$CRLF
021154                          20$:
021154  005777  157756          2$:         TST     @SWR              ;;HALT ON ERROR
021160  100006                              BPL     3$                ;;SKIP IF CONTINUE
021162  000000                              HALT                      ;;HALT ON ERROR!
021164  022737  010570  000042              CMP     #$ENDAD,@#42      ;;ACT-11 AUTO-ACCEPT?
021172  001001                              BNE     3$                ;;BRANCH IF NO
021174  000000                              HALT                      ;;YES
```

```
021176  032777  001000  157732  3$:         BIT     #BIT09,@SWR       ;;LOOP ON ERROR SWITCH SET?
021204  001402                              BEQ     4$                ;;BR IF NO
021206  013716  001110                      MOV     $LPERR,(SP)       ;;FUDGE RETURN FOR LOOPING
021212  005737  001160          4$:         TST     $ESCAPE           ;;CHECK FOR AN ESCAPE ADDRESS
021216  001402                              BEQ     5$                ;;BR IF NONE
021220  013716  001160                      MOV     $ESCAPE,(SP)      ;;FUDGE RETURN ADDRESS FOR ESCAPE
021224                          5$:
021224  000002                              RTI                       ;;RETURN
                                ;*************************************************************
```

```
                                   .SBTTL  TYPE ROUTINE

                                   ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                   ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                   ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                   ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                   ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                   ;*
                                   ;*CALL:
                                   ;*1) USING A TRAP INSTRUCTION
                                   ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                   ;*OR
                                   ;*      TYPE
                                   ;*      MESADR
                                   ;*
                                   ;*2) USING A JSR INSTRUCTION
                                   ;*      MOV     PS,-(SP)        ;;PUSH PROCESSOR STATUS WORD ON THE STACK
                                   ;*      JSR     PC,$TYPE        ;;CALL TYPE ROUTINE
                                   ;*      MESADDR                 ;;FIRST ADRESS OF MESSAGE

 021226  105737  001155            $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 021232  100002                            BPL     1$              ;;BR IF YES
 021234  000000                            HALT                    ;;HALT HERE IF NO TERMINAL
 021236  000407                            BR      3$              ;;LEAVE
 021240  010046                    1$:     MOV     R0,-(SP)        ;;SAVE R0
 021242  017600  000002                    MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 021246  112046                    2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 021250  001005                            BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 021252  005726                            TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 021254  012600                    60$:    MOV     (SP)+,R0        ;;RESTORE R0
 021256  062716  000002            3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 021262  000002                            RTI                     ;;RETURN
 021264  122716  000011            4$:     CMPB    #THT,(SP)       ;;BRANCH IF <HT>
 021270  001431                            BEQ     8$
 021272  122716  000200                    CMPB    #TCRLF,(SP)     ;;BRANCH IF NOT <CRLF>
 021276  001007                            BNE     5$
 021300  005726                            TST     (SP)+           ;;POP  <CR><LF> EQUIV
 021302  013746  177776                    MOV     PS,-(SP)        ;;TYPE CR AND LF
 021306  004737  021226                    JSR     PC,$TYPE
 021312  001167                            $CRLF
 021314  000754                            BR      2$              ;;GET NEXT CHARACTER
 021316  004737  021400            5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 021322  123726  001154            6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 021326  001347                            BNE     2$              ;;IF NO GO GET NEXT CHAR.
 021330  013746  001152                    MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                   ;;AND THE NULL CHAR.
```

```
 021334  105366  000001            7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
 021340  002770                            BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
 021342  004737  021400                    JSR     PC,$TYPEC       ;;GO TYPE A NULL
 021346  105337  021444                    DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 021352  000770                            BR      7$              ;;LOOP

                                   ;HORIZONTAL TAB PROCESSOR

 021354  112716  000040            8$:     MOVB    #40,(SP)        ;;REPLACE TAB WITH SPACE
 021360  004737  021400            9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
 021364  132737  000007  021444            BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 021372  001372                            BNE     9$              ;;TAB STOP
 021374  005726                            TST     (SP)+           ;;POP SPACE OFF STACK
 021376  000723                            BR      2$              ;;GET NEXT CHARACTER
```

```
                                                        L06
021400  105777  157542          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
021404  100375                          BPL     $TYPEC
021406  116677  000002  157534          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
021414  122766  000015  000002          CMPB    #15,2(SP)       ;;BRANCH IF
021422  001003                          BNE     1$              ;;NOT <CR>
021424  105037  021444                  CLRB    $CHARCNT        ;;
021430  000406                          BR      $TYPEX          ;;EXIT
021432  122766  000012  000002  1$:     CMPB    #12,2(SP)       ;;BRANCH IF
021440  002002                          BGE     $TYPEX          ;;<LF>
021442  105227                          INCB    (PC)+           ;;INC SPACE
021444  000000                  $CHARCNT:.WORD  0               ;;COUNT
021446  000207                  $TYPEX: RTS     PC
                                ;;      EQUATES
        000011                  THT=11
        000200                  TCRLF=200

                                ;*************************************************************

                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

                                ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
                                ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
                                ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

021450                          $ERRTYP:
021450  10-400  001167                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
021454  010046                          MOV     R0,-(SP)        ;;SAVE R0
021456  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
021460  153700  001114                  BISB    @#$ITEMB,R0
021464  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                                ;;TYPE THE PC OF THE ERROR
021466  013746  001116                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                ;;ERROR ADDRESS
021472  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
021474  000445                          BR      10$             ;;GET OUT
021476  005300                  1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
021500  006300                          ASL     R0              ;;      WORK FOR THE ERROR TABLE
021502  006300                          ASL     R0
021504  006300                          ASL     R0
021506  062700  0C1172                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
021512  012037  021522                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
021516  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
```

```
021520  104400                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
021522  000000                  2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
021524  104400  001167                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
021530  012037  021540          3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
021534  001404                          BEQ     5$              ;;SKIP TYPEOUT IF 0
021536  104400                          TYPE                    ;;TYPE THE "DATA HEADER"
021540  000000                  4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
021542  104400  001167                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
021546  010146                  5$:     MOV     R1,-(SP)        ;;SAVE R1
021550  012001                          MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
021552  001415                          BEQ     9$              ;;BR IF NO DATA TO BE TYPED
021554  012000                          MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
021556  105720                  6$:     TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
021560  001003                          BNE     7$              ;;BR IF DECIMAL
021562  013146                          MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
021564  104401                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
021566  000402                          BR      8$
021570                          7$:
021570  013146                          MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
```

```
021572  104404                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
021574  005711                  8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
021576  001403                          BEQ     9$              ;;BR IF NO
021600  104400  021620                  TYPE    ,11$            ;;TYPE TWO(2) SPACES
021604  000764                          BR      6$              ;;LOOP

021606  012601                  9$:     MOV     (SP)+,R1        ;;RESTORE R1
021610  012600                  10$:    MOV     (SP)+,R0        ;;RESTORE R0
021612  104400  001167                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
021616  000207                          RTS     PC              ;;RETURN
021620  020040     000          11$:    .ASCIZ  /  /            ;;TWO(2) SPACES
        021624                          .EVEN
                                ;*************************************************************

                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT
```

```

021624  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
021630  116637  000001  022047          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
021636  112637  022051                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
021642  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
021646  000406                          BR      $TYPON
021650  112737  000001  022047  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
021656  112737  000006  022051          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
021664  112737  000005  022046  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
021672  010346                          MOV     R3,-(SP)        ;;SAVE R3
021674  010446                          MOV     R4,-(SP)        ;;SAVE R4
021676  010546                          MOV     R5,-(SP)        ;;SAVE R5
021700  113704  022051                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
021704  005404                          NEG     R4
021706  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
021712  110437  022050                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
021716  113704  022047                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
021722  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
021726  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
021730  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
021732  000404                          BR      3$              ;;GO DO MSB
021734  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
021736  006105                          ROL     R5
021740  006105                          ROL     R5
```

N06

```
021742  010503                          MOV     R5,R3
021744  006103                  3$:     ROL     R3                      ;;GET LSB OF THIS DIGIT
021746  105337  022050                  DECB    $0MODE                  ;;TYPE THIS DIGIT?
021752  100016                          BPL     7$                      ;;BR IF NO
021754  042703  177770                  BIC     #177770,R3              ;;GET RID OF JUNK
021760  001002                          BNE     4$                      ;;TEST FOR 0
021762  005704                          TST     R4                      ;;SUPPRESS THIS 0?
021764  001403                          BEQ     5$                      ;;BR IF YES
021766  005204                  4$:     INC     R4                      ;;DON'T SUPPRESS ANYMORE 0'S
021770  052703  000060                  BIS     #'0,R3                  ;;MAKE THIS DIGIT ASCII
021774  052703  000040          5$:     BIS     #' ,R3                  ;;MAKE ASCII IF NOT ALREADY
022000  110337  022044                  MOVB    R3,8$                   ;;SAVE FOR TYPING
022004  104400  022044                  TYPE    ,8$                     ;;GO TYPE THIS DIGIT
022010  105337  022046          7$:     DECB    $0CNT                   ;;COUNT BY 1
022014  003347                          BGT     2$                      ;;BR IF MORE TO DO
022016  002402                          BLT     6$                      ;;BR IF DONE
022020  005204                          INC     R4                      ;;INSURE LAST DIGIT ISN'T A BLANK
022022  000744                          BR      2$                      ;;GO DO THE LAST DIGIT
022024  012605                  6$:     MOV     (SP)+,R5                ;;RESTORE R5
022026  012604                          MCV     (SP)+,R4                ;;RESTORE R4
022030  012603                          MOV     (SP)+,R3                ;;RESTORE R3
022032  016666  000002  000004          MOV     2(SP),4(SP)             ;;SET THE STACK FOR RETURNING
022040  012616                          MOV     (SP)+,(SP)
022042  000002                          RTI                             ;;RETURN
022044     000                  8$:     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
022045     000                          .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
022046     000                  $0CNT:  .BYTE   0                       ;;OCTAL DIGIT COUNTER
022047     000                  $0FILL: .BYTE   0                       ;;ZERO FILL SWITCH
022050  000000                  $0MODE: .WORD   0                       ;;NUMBER OF DIGITS TO TYPE
                                ;*****************************************************************


                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
```

```

                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                ;*REPLACED WITH SPACES.
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)               ;;PUT THE BINARY NUMBER ON THE STACK
                                ;*      TYPDS                           ;;GO TO THE ROUTINE

022052                          $TYPDS:
022052  010046                          MOV     R0,-(SP)                ;;PUSH R0 ON STACK
022054  010146                          MOV     R1,-(SP)                ;;PUSH R1 ON STACK
022056  010246                          MOV     R2,-(SP)                ;;PUSH R2 ON STACK
022060  010346                          MOV     R3,-(SP)                ;;PUSH R3 ON STACK
022062  010546                          MOV     R5,-(SP)                ;;PUSH R5 ON STACK
022064  012746  020200                  MOV     #20200,-(SP)            ;;SET BLANK SWITCH AND SIGN
022070  016605  000020                  MOV     20(SP),R5               ;;GET THE INPUT NUMBER
022074  100004                          BPL     1$                      ;;BR IF INPUT IS POS.
022076  005405                          NEG     R5                      ;;MAKE THE BINARY NUMBER POS.
022100  112766  000055  000001          MOVB    #'-,1(SP)               ;;MAKE THE ASCII NUMBER NEG.
022106  005000                  1$:     CLR     R0                      ;;ZERO THE CONSTANTS INDEX
022110  012703  022266                  MOV     #$DBLK,R3               ;;SETUP THE OUTPUT POINTER
022114  112723  000040                  MOVB    #' ,(R3)+               ;;SET THE FIRST CHARACTER TO A BLANK
022120  005002                  2$:     CLR     R2                      ;;CLEAR THE BCD NUMBER
022122  016001  022256                  MOV     $DTBL(R0),R1            ;;GET THE CONSTANT
022126  160105                  3$:     SUB     R1,R5                   ;;FORM THIS BCD DIGIT
022130  002402                          BLT     4$                      ;;BR IF DONE
022132  005202                          INC     R2                      ;;INCREASE THE BCD DIGIT BY 1
```

```
022134  000774                          BR      3$
022136  060105                  4$:     ADD     R1,R5                   ;;ADD BACK THE CONSTANT
022140  005702                          TST     R2                      ;;CHECK IF BCD DIGIT=0
022142  001002                          BNE     5$                      ;;FALL THROUGH IF 0
022144  105716                          TSTB    (SP)                    ;;STILL DOING LEADING 0'S?
022146  100407                          BMI     7$                      ;;BR IF YES
022150  106316                  5$:     ASLB    (SP)                    ;;MSD?
022152  103003                          BCC     6$                      ;;BR IF NO
022154  116663  000001  177777          MOVB    1(SP),-1(R3)            ;;YES--SET THE SIGN
022162  052702  000060          6$:     BIS     #'0,R2                  ;;MAKE THE BCD DIGIT ASCII
022166  052702  000040          7$:     BIS     #' ,R2                  ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
022172  110223                          MOVB    R2,(R3)+                ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
022174  005720                          TST     (R0)+                   ;;JUST INCREMENTING
022176  020027  000010                  CMP     R0,#10                  ;;CHECK THE TABLE INDEX
022202  002746                          BLT     2$                      ;;GO DO THE NEXT DIGIT
022204  003002                          BGT     9$                      ;;GO TO EXIT
022206  010502                          MOV     R5,R2                   ;;GET THE LSD
022210  000764                          BR      6$                      ;;GO CHANGE TO ASCII
022212  105726                  8$:     TSTB    (SP)+                   ;;WAS THE LSD THE FIRST NON-ZERO?
022214  100003                          BPL     9$                      ;;BR IF NO
022216  116663  177777  177776          MOVB    -1(SP),-2(R3)           ;;YES--SET THE SIGN FOR TYPING
022224  105013                  9$:     CLRB    (R3)                    ;;SET THE TERMINATOR
022226  012605                          MOV     (SP)+,R5                ;;POP STACK INTO R5
022230  012603                          MOV     (SP)+,R3                ;;POP STACK INTO R3
022232  012602                          MOV     (SP)+,R2                ;;POP STACK INTO R2
022234  012601                          MOV     (SP)+,R1                ;;POP STACK INTO R1
022236  012600                          MOV     (SP)+,R0                ;;POP STACK INTO R0
```

```
022240  104400  022266                  TYPE    ,$DBLK                  ;;NOW TYPE THE NUMBER
022244  016666  000002  000004          MOV     2(SP),4(SP)             ;;ADJUST THE STACK
022252  012616                          MOV     (SP)+,(SP)
022254  000002                          RTI                             ;;RETURN TO USER
022256  023420                  $DTBL:  10000.
022260  001750                          1000.
022262  000144                          100.
022264  000012                          10.
022266  000004                  $DBLK:  .BLKW   4
                                ;*****************************************************-***********

                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;POWER DOWN ROUTINE
022276  012737  022420  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC        ;;SET FOR FAST UP
022304  012737  000340  000026          MOV     #340,@#PWRVEC+2         ;;PRIO:7
022312  010046                          MOV     R0,-(SP)                ;;PUSH R0 ON STACK
022314  010146                          MOV     R1,-(SP)                ;;PUSH R1 ON STACK
022316  010246                          MOV     R2,-(SP)                ;;PUSH R2 ON STACK
022320  010346                          MOV     R3,-(SP)                ;;PUSH R3 ON STACK
022322  010446                          MOV     R4,-(SP)                ;;PUSH R4 ON STACK
022324  010546                          MOV     R5,-(SP)                ;;PUSH R5 ON STACK
022326  010637  022424                  MOV     SP,$SAVR6               ;;SAVE SP
022332  012737  022344  000024          MOV     #$PWRUP,@#PWRVEC        ;;SET UP VECTOR
022340  000000                          HALT
022342  000776                          BR      .-2                     ;;HANG UP

                                ;POWER UP ROUTINE
022344  013706  022424          $PWRUP: MOV     $SAVR6,SP               ;;GET SP
022350  005037  022424                  CLR     $SAVR6                  ;;WAIT LOOP FOR THE TTY
022354  005237  022424          1$:     INC     $SAVR6                  ;;WAIT FOR THE INC
022360  001375                          BNE     1$                      ;;OF  WORD
022362  012605                          MOV     (SP)+,R5                ;;POP STACK INTO R5
022364  012604                          MOV     (SP)+,R4                ;;POP STACK INTO R4
```

```
022366  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
022370  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
022372  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
022374  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
022376  012737  022276  000024            MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
022404  012737  000340  000026            MOV     #340,@#PWRVEC+2 ;;PRIO:7
022412  104400                            TYPE                    ;REPORT THE POWER FAILURE
022414  022426                    $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
022416  000002                            RTI
022420  000000                    $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
022422  000776                            BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
022424  000000                    $SAVR6: 0                       ;;PUT THE SP HERE
022426  005015  047520  042527    $POWER: .ASCIZ  <15><12>"POWER"
022434  000122
                                          .EVEN
                                  ;*****************************************************************


                                  .SBTTL  TRAP DECODER

                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
```

```
                                  ;*GO TO THAT ROUTINE.

022436  010046                    $TRAP:  MOV     R0,-(SP)          ;;SAVE R0
022440  016600  000002                    MOV     2(SP),R0          ;;GET TRAP ADDRESS
022444  005740                            TST     -(R0)             ;;BACKUP BY 2
022446  111000                            MOVB    (R0),R0           ;;GET RIGHT BYTE OF TRAP
022450  006300                            ASL     R0                ;;POSITION FOR INDEXING
022452  016000  022460                    MOV     $TRPAD(R0),R0     ;;INDEX TO TABLE
022456  000200                            RTS     R0                ;;GO TO ROUTINE


                                  .SBTTL  TRAP TABLE

                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                  ;*BY THE "TRAP" INSTRUCTION.

                                  ;       ROUTINE
                                  ;       -------
022460                            $TRPAD:
022460  021226                            $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
022462  021650                            $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
022464  021624                            $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
022466  021664                            $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
022470  022052                            $TYPDS  ;;CALL=TYPDS    TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)

022472  003062  003450  003656    TSTADD: TST1,TST2,TST3

022500  004052  004204  004420            TST4,TST5,TST6

022506  004650  005424  006104            TST7,TST10,TST11

022514  006316  006472  006720            TST12,TST13,TST14

022522  007142  007642  010060            TST15,TST16,TST17

022530  010226  010624  011130            TST20,TST21,TST22

022536  011326  011400                    TST23,TST24

```

```


022542  042523  020124  052126  STUPM:  .ASCII  /SET VT61S TO  FULL DUPLEX, /<15><12>
022550  030466  020123  047524
022556  020040  052506  046114
022564  042040  050125  042514
022572  026130  006440     012
022577     071  030066  041060          .ASCIZ  /9600BAUD, REMOTE,PARITY MATCHED TO INTERFACE/<15><12>
022604  052501  026104  051040
022612  046505  052117  026105
022620  040520  044522  054524
022626  046440  052101  044103
022634  042105  052040  020117
022642  047111  042524  043122
022650  041501  006505  000012


```

```

022656  005015  042101  051104  DUNTST: .ASCIZ  <15><12>/ADDRESSES WITH RESPONSIVE VT61S ARE:/<15><12>
022664  051505  042523  020123
022672  044527  044124  051040
022700  051505  047520  051516
022706  053111  020105  052126
022714  030466  020123  051101
022722  035105  005015     000
022727     015  047012  020117  NOVT:   .ASCIZ  <15><12>/NO VT61 RESPONDED TO ESCZ SEQ. AUTO RETRY IN 30 SEC./<15><12>
022734  052126  030466  051040
022742  051505  047520  042116
022750  042105  052040  020117
022756  051505  055103  051440
022764  050505  020056  052501
022772  047524  051040  052105
023000  054522  044440  020116
023006  030063  051440  041505
023014  006456  000012


023020  005015  046104  030461  DLERR:  .ASCIZ  <15><12>/DL11 FAILED AT ADDRESS/
023026  043040  044501  042514
023034  020104  052101  040440
023042  042104  042522  051523
023050     000

023051     115  047101  040525  DMANA:  .ASCII  /MANUAL TEST SELECTED -/<15><12>
023056  020114  042524  052123
023064  051440  046105  041505
023072  042524  020104  006455
023100     012
023101     105  052116  051105          .ASCIZ  /ENTER ADDRESSES OF VT61S TO BE TESTED/<15><12>
023106  040440  042104  042522
023114  051523  051505  047440
023122  020106  052126  030466
023130  020123  047524  041040
023136  020105  042524  052123
023144  042105  005015     000

023151     105  052116  051105  DMANB:  .ASCIZ  /ENTER TESTS TO BE RUN/<15><12>
023156  052040  051505  051524
023164  052040  020117  042502
023172  051040  047125  005015
023200     000
```

E07

```
 4037
 4038  023201     101  020116  051505  EM1:    .ASCIZ  /AN ESC SEQ. TO THE VT61  FAILED - OCTAL EQUIV. IS:/<15><12>
 4039  023206  020103  042523  027121
 4040  023214  052040  020117  044124
 4041  023222  020105  052126  030466
 4042  023230  020040  040506  046111
 4043  023236  042105  026440  047440
 4044  023244  052103  046101  042440
 4045  023252  052521  053111  020056
 4046  023260  051511  006472  000012
 4047  023266  042524  052123  020043  DH1:    .ASCIZ  /TEST#  ERR PC  BYTE 1+2 BYTE 3+4/<15><12>
 4048  023274  042440  051122  050040
```

```
 4049  023302  020103  041040  052131
 4050  023310  020105  025461  020062
 4051  023316  054502  042524  031440
 4052  023324  032053  005015     000
 4053
 4054  023331     122  041505  044505  EM2:    .ASCIZ  /RECEIVE STATUS ERROR./<15><12>
 4055  023336  042526  051440  040524
 4056  023344  052524  020123  051105
 4057  023352  047522  027122  005015
 4058  023360     000
 4059  023361     101  042104  020056  DH2:    .ASCIZ  /ADD.  STAT.  ERR.BITS  CHAR./<15><12>
 4060  023366  051440  040524  027124
 4061  023374  020040  051105  027122
 4062  023402  044502  051524  020040
 4063  023410  044103  051101  006456
 4064  023416  000012
 4065
 4066  023420  047523  052106  040527  EM3:    .ASCIZ  /SOFTWARE (VSTAT) STATUS ERROR./<15><12>
 4067  023426  042522  024040  051526
 4068  023434  040524  024524  051440
 4069  023442  040524  052524  020123
 4070  023450  051105  047522  027122
 4071  023456  005015     000
 4072  023461     040  040520  051523  DH3:    .ASCIZ  / PASS#,  TEST#,  EXP.STAT,  ACT.STAT/<15><12>
 4073  023466  026043  020040  042524
 4074  023474  052123  026043  020040
 4075  023502  054105  027120  052123
 4076  023510  052101  020054  040440
 4077  023516  052103  051456  040524
 4078  023524  006524  000012
 4079
 4080  023530  042107  020056  040504  EM4:    .ASCIZ  /GD. DATA DOES NOT MATCH REC. DATA/<15><12>
 4081  023536  040524  042040  042517
 4082  023544  020123  047516  020124
 4083  023552  040515  041524  020110
 4084  023560  042522  027103  042040
 4085  023566  052101  006501  000012
 4086  023574  042524  052123  020043  DH4:    .ASCIZ  /TEST# ,REC.CNT.,GD. DATA, REC. DATA/<15><12>
 4087  023602  051054  041505  041456
 4088  023610  052116  026056  042107
 4089  023616  020056  040504  040524
 4090  023624  020054  042522  027103
 4091  023632  042040  052101  006501
 4092  023640  000012
 4093                                          .EVEN
 4094
 4095  023642  054502  042524  020123  EM5:    .ASCIZ  /BYTES EXPECTED DOES NOT EQUAL BYTES RECEIVED/<15><12>
 4096  023650  054105  042520  052103
 4097  023656  042105  042040  042517
```

```
  4098  023664  020123  047516  020124
  4099  023672  050505  040525  020114
  4100  023700  054502  042524  020123
  4101  023706  042522  042503  053111
  4102  023714  042105  005015     000
  4103  023721     102  052131  051505  DH5:    .ASCIZ  /BYTES EXP., BYTES REC./<15><12>
  4104  023726  042440  050130  026056
```

```
  4105  023734  041040  052131  051505
  4106  023742  051040  041505  006456
  4107  023750  000012
  4108
  4109  023752  052503  051522  051117  EM6:    .ASCIZ  /CURSOR POSITIONING ERROR/<15><12>
  4110  023760  050040  051517  052111
  4111  023766  047511  044516  043516
  4112  023774  042440  051122  051117
  4113  024002  005015     000
  4114  024005     107  020104  044514  DH6:    .ASCIZ  /GD LINE  GD COL.   BD LINE  BD COL/<15><12>
  4115  024012  042516  020040  042107
  4116  024020  041440  046117  020056
  4117  024026  020040  042102  046040
  4118  024034  047111  020105  041040
  4119  024042  020104  047503  006514
  4120  024050  000012
  4121
  4122  024052  044504  042522  052103  EM7:    .ASCIZ  /DIRECT CURSOR ADDRESSING FAILURE/<15><12>
  4123  024060  041440  051125  047523
  4124  024066  020122  042101  051104
  4125  024074  051505  044523  043516
  4126  024102  043040  044501  052514
  4127  024110  042522  005015     000
  4128  024115     120  051501  021523  DH7:    .ASCIZ  /PASS#  TEST #  ERROR PC   /<15><12>
  4129  024122  020040  042524  052123
  4130  024130  021440  020040  051105
  4131  024136  047522  020122  041520
  4132  024144  020040  006440  000012
  4133  024152  040520  051523  020043  DH10:   .ASCIZ  /PASS#  TEST#  BD.ROW  BD.COL/<15><12>
  4134  024160  052040  051505  021524
  4135  024166  020040  042102  051056
  4136  024174  053517  020040  042102
  4137  024202  041456  046117  005015
  4138  024210     000
  4139
  4140  024211     114  051501  020124  EM11:   .ASCIZ  /LAST TRANSMISSION TO VT61 CAUSED UNIT TO FAIL-HANG./<15><12>
  4141  024216  051124  047101  046523
  4142  024224  051511  044523  047117
  4143  024232  052040  020117  052126
  4144  024240  030466  041440  052501
  4145  024246  042523  020104  047125
  4146  024254  052111  052040  020117
  4147  024262  040506  046111  044055
  4148  024270  047101  027107  005015
  4149  024276     000
  4150
  4151  024277     126  033124  020061  EM12:   .ASCIZ  /VT61 UNDER TEST FAILED- ERROR DATA FOLLOWS/<15><12>
  4152  024304  047125  042504  020122
  4153  024312  042524  052123  043040
  4154  024320  044501  042514  026504
  4155  024326  042440  051122  051117
  4156  024334  042040  052101  020101
  4157  024342  047506  046114  053517
  4158  024350  006523  000012
  4159
  4160  024354  052126  030466  043040  EM10:   .ASCIZ  /VT61 FAILED SELF TEST FUNCTION/<15><12>
```

```
  4161  024362  044501  042514  020104
  4162  024370  042523  043114  052040
  4163  024376  051505  020124  052506
  4164  024404  041516  044524  047117
  4165  024412  005015     000
  4166
  4167
  4168  024415     120  051501  021523  DH12:   .ASCIZ  /PASS#,  TEST#, GD.CKSUM, BD.CKSUM/<15><12>
  4169  024422  020054  052040  051505
  4170  024430  021524  020054  042107
  4171  024436  041456  051513  046525
  4172  024444  020054  042102  041456
  4173  024452  051513  046525  005015
  4174  024460     000
  4175
  4176  024461     124  051505  044524  DABRT:  .ASCIZ  /TESTING ABORTED-TOO MANY FATAL XMITS/<15><12>
  4177  024466  043516  040440  047502
  4178  024474  052122  042105  052055
  4179  024502  047517  046440  047101
  4180  024510  020131  040506  040524
  4181  024516  020114  046530  052111
  4182  024524  006523  000012
  4183
  4184  024530  052126  030466  051040  EM13:   .ASCIZ  /VT61 RECEIVER CHECKSUM COMPARE ERROR/<15><12>
  4185  024536  041505  044505  042526
  4186  024544  020122  044103  041505
  4187  024552  051513  046525  041440
  4188  024560  046517  040520  042522
  4189  024566  042440  051122  051117
  4190  024574  005015     000
  4191
  4192  024577     126  033124  020061  EM14:   .ASCIZ  /VT61 TRANSMITTER CHECKSUM COMPARE ERROR/<15><12>
  4193  024604  051124  047101  046523
  4194  024612  052111  042524  020122
  4195  024620  044103  041505  051513
  4196  024626  046525  041440  046517
  4197  024634  040520  042522  042440
  4198  024642  051122  051117  005015
  4199  024650     000
  4200
  4201          024652                             .EVEN
  4202  024652  047125  052111  052440  DVUNIT:  .ASCII  /UNIT UNDER TEST /<15><12>
  4203  024660  042116  051105  052040
  4204  024666  051505  020124  005015
  4205  024674  041522  051123  020040           .ASCIZ  /RCSR   VECT.   IDENT/<15><12>
  4206  024702  053040  041505  027124
  4207  024710  020040  044440  042504
  4208  024716  052116  005015     000
  4209  024723     040  041522  051123  DH11:    .ASCIZ  / RCSR   VECT./<15><12>
  4210  024730  020040  053040  041505
  4211  024736  027124  005015     000
  4212  024743     120  044522  052116  DPRTR:   .ASCIZ  /PRINTER IS ATTACHED/<15><12>
  4213  024750  051105  044440  020123
  4214  024756  052101  040524  044103
  4215  024764  042105  005015     000
  4216  024771     103  050117  042511  DCOPYR:  .ASCIZ  /COPIER IS ATTACHED/<15><12>
```

```
  4217  024776  020122  051511  040440
  4218  025004  052124  041501  042510
  4219  025012  006504  000012
  4220  025016  047530  043106  052040  EM15:    .ASCIZ  /XOFF TO VT61 FAILED TO HALT BLOCK XMIT/<15><12>
  4221  025024  020117  052126  030466
  4222  025032  043040  044501  042514
  4223  025040  020104  047524  044040
  4224  025046  046101  020124  046102
  4225  025054  041517  020113  046530
  4226  025062  052111  005015     000
  4227  025067     130  047117  052040  EM16:    .ASCIZ  /XON TO VT61 FAILED TO RESTART BLOCK XMIT/<15><12>
  4228  025074  020117  052126  030466
  4229  025102  043040  044501  042514
  4230  025110  020104  047524  051040
  4231  025116  051505  040524  052122
  4232  025124  041040  047514  045503
  4233  025132  054040  044515  006524
  4234  025140  000012
  4235  025142  047516  054040  047117  EM17:    .ASCIZ  /NO XON RECEIVED WITHIN 3 SEC. AFTER A RESET/<15><12>
  4236  025150  051040  041505  044505
  4237  025156  042526  020104  044527
  4238  025164  044124  047111  031440
  4239  025172  051440  041505  020056
  4240  025200  043101  042524  020122
  4241  025206  020101  042522  042523
  4242  025214  006524  000012
  4243  025220  040514  052123  050040  EM20:    .ASCIZ  /LAST PERIPHERAL OPERATION ABORTED/<15><12>
  4244  025226  051105  050111  042510
  4245  025234  040522  020114  050117
  4246  025242  051105  052101  047511
  4247  025250  020116  041101  051117
  4248  025256  042524  006504  000012
  4249  025264  047503  046125  020104  EM21:    .ASCIZ  /COULD NOT CLEAR LAST ABORT FLAG./<15><12>
  4250  025272  047516  020124  046103
  4251  025300  040505  020122  040514
  4252  025306  052123  040440  047502
  4253  025314  052122  043040  040514
  4254  025322  027107  005015     000
  4255  025327     123  046517  047440  EM22:    .ASCIZ  /SOM OR EOM NOT RECEIVED DURING MAINT. MODE TRANSMIT/<15><12>
```

```
 4256  025334  020122  047505  020115
 4257  025342  047516  020124  042522
 4258  025350  042503  053111  042105
 4259  025356  042040  051125  047111
 4260  025364  020107  040515  047111
 4261  025372  027124  045440  042117
 4262  025400  020105  051124  047101
 4263  025406  046523  052111  005015
 4264  025414     000
 4265  025415     114  047111  020105  EM23:   .ASCIZ  /LINE FEED OR CURSOR RIGHT ISSUED FROM ROW 23 DID NOT CAUSE SCREEN TO SC
 4266  025422  042506  042105  047440
 4267  025430  020122  052503  051522
 4268  025436  051117  051040  043511
 4269  025444  052110  044440  051523
 4270  025452  042525  020104  051106
 4271  025460  046517  051040  053517
 4272  025466  031040  020063  044504
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 78
DZVTH.P11       TRAP TABLE                                                                                      SEQ 0089

 4273  025474  020104  047516  020124
 4274  025502  040503  051525  020105
 4275  025510  041523  042522  047105
 4276  025516  052040  020117  041523
 4277  025524  047522  046114  005015
 4278  025532     000
 4279  025533     120  051501  020123  DH13:   .ASCIZ  /PASS ,   TEST ,   VSTAT/<15><12>
 4280  025540  020054  020040  042524
 4281  025546  052123  026040  020040
 4282  025554  053040  052123  052101
 4283  025562  005015     000
 4284  025565     120  051501  026123  DH14:   .ASCIZ  /PASS,   TEST,   ERR PC,   VSTAT/<15><12>
 4285  025572  020040  052040  051505
 4286  025600  026124  020040  042440
 4287  025606  051122  050040  026103
 4288  025614  020040  053040  052123
 4289  025622  052101  005015     000
 4290
 4291  025627     105  041523  000040  DESC:   .ASCIZ  /ESC /
 4292
 4293
 4294
 4295  025634  042513  041131  040517  DKYBD:  .ASCII  /KEYBOARD TEST/<15><12>
 4296  025642  042122  052040  051505
 4297  025650  006524     012
 4298  025653     113  054505  052123          .ASCII  /KEYSTROKES ECHO:/<15><12>
 4299  025660  047522  042513  020123
 4300  025666  041505  047510  006472
 4301  025674     012
 4302  025675     101  042040  051511          .ASCII  /A DISPLAY CHAR. = A DISPLAY CHAR./<15><12>
 4303  025702  046120  054501  041440
 4304  025710  040510  027122  036440
 4305  025716  040440  042040  051511
 4306  025724  046120  054501  041440
 4307  025732  040510  027122  005015
 4308  025740  031463  036440  042440          .ASCII  /33 = ESC/<15><12>
 4309  025746  041523  005015
 4310  025752  032461  036440  041440          .ASCII  /15 = C-R/<15><12>
 4311  025760  051055  005015
 4312  025764  031061  036440  046040          .ASCII  /12 = L-F/<15><12>
 4313  025772  043055  005015
 4314  025776  033460  036440  041040          .ASCII  /07 = BELL/<15><12>
 4315  026004  046105  006514     012
 4316  026011     061  020060  020075          .ASCII  /10 = TAB/<15><12>
 4317  026016  040524  006502     012
 4318  026023     116  047117  042055          .ASCIZ  /NON-DISPLAY CHAR.= OCTAL EQUIV/<15><12>
 4319  026030  051511  046120  054501
 4320  026036  041440  040510  027122
 4321  026044  020075  041517  040524
 4322  026052  020114  050505  044525
 4323  026060  006526  000012
 4324
 4325  026064  040524  020102     000  DTAB:   .ASCIZ  /TAB /
 4326  026071     103  051055  000040  DCR:    .ASCIZ  /C-R /
 4327  026076  026514  020106     000  DLF:    .ASCIZ  /L-F /
 4328  026103     102  046105  020114  DBELL:  .ASCIZ  /BELL /
```

```
 4329  026110     000
 4330
 4331  026111     114  047517  020120  DLOOP:  .ASCII  /LOOP TEST - LOOP COMMANDS AND DATA THRU/<15><12>
 4332  026116  042524  052123  026440
 4333  026124  046040  047517  020120
 4334  026132  047503  046515  047101
 4335  026140  051504  040440  042116
 4336  026146  042040  052101  020101
 4337  026154  044124  052522  005015
 4338  026162  047510  052123  041040          .ASCII  /HOST BACK TO VT61 UNDER TEST. /<15><12>
 4339  026170  041501  020113  047524
 4340  026176  053040  033124  020061
 4341  026204  047125  042504  020122
 4342  026212  042524  052123  020056
 4343  026220  005015
 4344  026222  047503  052116  047522  DCNTZ:  .ASCIZ  /CONTROL C  EXITS TEST./<15><12>
 4345  026230  020114  020103  042440
 4346  026236  044530  051524  052040
 4347  026244  051505  027124  005015
 4348  026252     000
 4349
 4350  026253     105  044530  020124  DEXT:   .ASCIZ  /EXIT TEST./
 4351  026260  042524  052123  000056
 4352
 4353  026266  051120  047111  042524  DPRNT:  .ASCII  /PRINTER TEST -/<15><12>
 4354  026274  020122  042524  052123
 4355  026302  026440  005015
 4356  026306  031461  020062  047503          .ASCII  /132 COLUMNS OF A SLIDING PATTERN WILL BE/
 4357  026314  052514  047115  020123
 4358  026322  043117  040440  051440
 4359  026330  044514  044504  043516
 4360  026336  050040  052101  042524
 4361  026344  047122  053440  046111
 4362  026352  020114  042502
 4363  026356  047503  052116  047111          .ASCII  /CONTINOUSLY OUTPUTTED TO PRINTER/<15><12>
 4364  026364  052517  046123  020131
 4365  026372  052517  050124  052125
 4366  026400  042524  020104  047524
 4367  026406  050040  044522  052116
 4368  026414  051105  005015
 4369  026420  040503  027122  051040  DCRST:  .ASCIZ  /CAR. RET. TO START/<15><12>
 4370  026426  052105  020056  047524
 4371  026434  051440  040524  052122
 4372  026442  005015     000
 4373
 4374  026445     114  051501  020124  DEVERR: .ASCIZ  /LAST XMIT CAUSED VT61 HANG/<15><12>
 4375  026452  046530  052111  041440
 4376  026460  052501  042523  020104
 4377  026466  052126  030466  044040
 4378  026474  047101  006507  000012
 4379  026502  051120  042117  041525  DKBD:   .ASCII  /PRODUCTION KEYBOARD TEST. 10 ERRORS CAUSES TEST EXIT./<15><12>
 4380  026510  044524  047117  045440
 4381  026516  054505  047502  051101
 4382  026524  020104  042524  052123
 4383  026532  020056  030061  042440
 4384  026540  051122  051117  020123
```

```
 4385  026546  040503  051525  051505
 4386  026554  052040  051505  020124
 4387  026562  054105  052111  006456
 4388  026570     012
 4389  026571     104  050105  042522          .ASCIZ  /DEPRESS KEYS FROM LEFT TO RIGHT/<15><12>
 4390  026576  051523  045440  054505
 4391  026604  020123  051106  046517
 4392  026612  046040  043105  020124
 4393  026620  047524  051040  043511
 4394  026626  052110  005015     000
 4395  026633     104  050105  042522  DLSHFT: .ASCIZ  /DEPRESS LEFT SHIFT KEY AND THE "A" KEY /<15><12>
 4396  026640  051523  046040  043105
 4397  026646  020124  044123  043111
 4398  026654  020124  042513  020131
 4399  026662  047101  020104  044124
 4400  026670  020105  040442  C20042
 4401  026676  042513  020131  005015
 4402  026704     000
 4403  026705     104  050105  042522  DTOP:   .ASCIZ  /DEPRESS KEYS IN TOP ROW/<15><12>
 4404  026712  051523  045440  054505
 4405  026720  020123  047111  052040
 4406  026726  050117  051040  053517
 4407  026734  005015     000
 4408
 4409  026737     104  050105  042522  DRSHFT: .ASCIZ  /DEPRESS RIGHT SHIFT KEY AND THE "A" KEY /<15><12>
 4410  026744  051523  051040  043511
 4411  026752  052110  051440  044510
 4412  026760  052106  045440  054505
 4413  026766  040440  042116  052040
 4414  026774  042510  021040  021101
 4415  027002  045440  054505  006440
 4416  027010  000012
 4417  027012  042504  051120  051505  DSEC:   .ASCIZ  /DEPRESS KEYS IN SECOND ROW/<15><12>
 4418  027020  020123  042513  051531
 4419  027026  044440  020116  042523
 4420  027034  047503  042116  051040
 4421  027042  053517  005015     000
 4422
 4423  027047     104  050105  042522  DTHRD:  .ASCIZ  /DEPRESS KEYS IN THIRD ROW BEGINNING WITH 'A'/<15><12>
 4424  027054  051523  045440  054505
 4425  027062  020123  047111  052040
 4426  027070  044510  042122  051040
 4427  027076  053517  041040  043505
 4428  027104  047111  044516  043516
 4429  027112  053440  052111  020110
 4430  027120  040447  006447  000012
 4431  027126  042504  051120  051505  DCONT:  .ASCIZ  /DEPRESS CONTROL KEY ,AND THE "A" KEY /<15><12>
 4432  027134  020123  047503  052116
 4433  027142  047522  020114  042513
 4434  027150  020131  040454  042116
 4435  027156  052040  042510  021040
 4436  027164  021101  045440  054505
 4437  027172  006440  000012
 4438  027176  042504  051120  051505  DBOT:   .ASCIZ  /DEPRESS KEYS IN FORTH ROW EXCEPT SHIFT KEYS/<15><12>
```

```
  4439  027204  020123  042513  051531
  4440  027212  044440  020116  047506
```

```
  4441  027220  052122  020110  047522
  4442  027226  020127  054105  042503
  4443  027234  052120  051440  044510
  4444  027242  052106  045440  054505
  4445  027250  006523  000012
  4446  027254  042504  051120  051505   DSPCE: .ASCIZ  /DEPRESS SPACE BAR/<15><12>
  4447  027262  020123  050123  041501
  4448  027270  020105  040502  006522
  4449  027276  000012
  4450
  4451  027300  042504  051120  051505  DKPD:   .ASCIZ  /DEPRESS KEYPAD KEYS,LEFT TO RIGHT, TOP TO BOTTOM/<15><12>
  4452  027306  020123  042513  050131
  4453  027314  042101  045440  054505
  4454  027322  026123  042514  052106
  4455  027330  052040  020117  044522
  4456  027336  044107  026124  052040
  4457  027344  050117  052040  020117
  4458  027352  047502  052124  046517
  4459  027360  005015     000
  4460
  4461  027363     113  054505  047502  DKBERR: .ASCII  /KEYBOARD ERROR,KEY POSITION IN ROW SHOULD BE /
  4462  027370  051101  020104  051105
  4463  027376  047522  026122  042513
  4464  027404  020131  047520  044523
  4465  027412  044524  047117  044440
  4466  027420  020116  047522  020127
  4467  027426  044123  052517  042114
  4468  027434  041040  020105
  4469  027440  020040  005015          KYSTRK: .ASCII  /  /<15><12>
  4470  027444  041517  040524  020114          .ASCIZ  /OCTAL GD, OCTAL BAD/<15><12>
  4471  027452  042107  020054  041517
  4472  027460  040524  020114  040502
  4473  027466  006504  000012
  4474  027472  020040  020040  020040  DSPC6:  .ASCIZ  /       /
  4475  027500     000
  4476
  4477  027501     036     076     020  ROW1:   .BYTE   36,76,20,13,32,12,54,44,14,41,71,57,63,64,3,114,0
  4478  027504     013     032     012
  4479  027507     054     044     014
  4480  027512     041     071     057
  4481  027515     063     064     003
  4482  027520     114     000
  4483
  4484  027522     026     056     030  ROW2:   .BYTE   26,56,30,73,52,22,55,34,24,31,51,77,62,61,2,0
  4485  027525     073     052     022
  4486  027530     055     034     024
  4487  027533     031     051     077
  4488  027536     062     061     002
  4489  027541     000
  4490
  4491  027542     046     040     053  ROW3:   .BYTE   46,40,53,23,72,42,45,74,11,21,47,27,66,0
  4492  027545     023     072     042
  4493  027550     045     074     011
  4494  027553     021     047     027
  4495  027556     066     000
  4496
```

M07

```
   4497  027560      016      070      060    ROW4:    .BYTE   16,70,60,50,33,43,25,35,75,65,37,115,67,0
   4498  027563      050      033      043
   4499  027566      025      035      075
   4500  027571      065      037      115
   4501  027574      067      000
   4502
   4503  027576      001      000             CNTRA:   .BYTE   01,0
   4504
   4505  027600      101      000             SHFTA:   .BYTE   101,0
   4506
   4507  027602      015      000             SPCB:    .BYTE   15,0
   4508
   4509  027604      113      004      103    KYPD:    .BYTE   113,04,103,104,1,112,101,102,6,7,106,100,5
   4510  027607      104      001      112
   4511  027612      101      102      006
   4512  027615      007      106      100
   4513  027620      005
   4514  027621      010      105      107             .BYTE   10,105,107,110,17,111,0
   4515  027624      110      017      111
   4516  027627      000
   4517
   4518                                                .EVEN
   4519  027630  000500                       RCRLB:   .BLKB   500      ;RECEIVE CIRCULAR BUFFER
   4520
   4521  030330  000500                       TCRLB:   .BLKB   500              ;TRANSMIT CIRCULAR BUFFER
   4522  031030  000000                       ABUFP:   .WORD   0
   4523  031032  000062                       ABBUF:   .BLKB   50.
   4524  031114  000000                                0
   4525          000001                                .END
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 84
DZVTH.P11       CROSS REFERENCE TABLE -- USER SYMBOLS                                                              SEQ 0094

ABBUF    031032          811     2516     2518     4523#
ABSXT    020424         3453#
ABUFP    031030          811*    1087     1119     1122     1134     1137     1348     1349     1356     1904     1907     1911*    2515*
                        2516     2518*    2519*    2757*    2758     2870     2874     3350*    3352     3355     3356*    3422     3427
                        3450     4522#
AESCO    014542         2587     2619#
AESCP    014534         2585     2616#
ALWCNT   002200          651#     913     1175     1616     2441
AOUT     012536         2177     2227#
ASESC    014356         2522     2584#
ASTRT    003062          814      827#    1839     1910     1991     2016     2053
AUTO     002230          141      670#
AUTOA    002240          672#     676      755     2127
BASC3    004222         1059     1060#
BBLUP    013250         2274     2277     2370#
BEL      001746          442#     833     1926
BINOCT   017266         1942     3207#    3365     3379     3385
BIT0   = 000001          115#
BIT00  = 000001          105#     115      803     1830     2205     2521
BIT01  = 000002          104#     114      806      877     2521     2590
BIT02  = 000004          103#     113     2156
BIT03  = 000010          102#     112     2521     2586     2596
BIT04  = 000020          101#     111
BIT05  = 000040          100#     110
BIT06  = 000100           99#     109
BIT07  = 000200           98#     108     1214     2485
BIT08  = 000400           97#     107     3531
BIT09  = 001000           96#     106     3539     3601
BIT1   = 000002          114#
BIT10  = 002000           95#    3583
BIT11  = 004000           94#    3546
BIT12  = 010000           93#
```

N07

```
BIT13 = 020000             92#    3590
BIT14 = 040000             91#    3517
BIT15 = 100000             90#
BIT2  = 000004            113#
BIT3  = 000010            112#
BIT4  = 000020            111#
BIT5  = 000040            110#
BIT6  = 000100            109#
BIT7  = 000200            108#
BIT8  = 000400            107#
BIT9  = 001000            106#
BLDADA  002312            694#     714
BLDADD  002310            692#     698
BLDINA  017102           1981     3151#    3158
BLDINC  017076           1028     1069     1306     1492     1817     3150#    3157     3169
BLDTST  002410            708      711      716#
BLKM    002224            661#     775*    1061*    1299*    1333*    1642*    1814*    2675     2678#    2683*    2710     2852*    3021*
BOROW   020042           3370     3373#
BPTVEC= 000014            122#
BUBCT   002206            654#    1812*    1824     1828*    2037*    2046     2326*    2340*    2342     2346*    2350*    2372*    2379*
                         2394*    2414     2756*    2762*    3359*    3360
BUMPCT  020344           3421     3432#
BYSTOR  016024           1498*    2854     2902#    2947*
CALCK   017516           1314     2514     2699     3289#
```

```
CARRT   001750            444#    1724     1735     1934     1982     2552     3243     3258     3268
CDEV    012242            673      705      709     2165#
CDEVA   012312           2175#    2181     2187     2223     2226
CDWN    001764            460#
CESAM   014526           2601     2605     2609     2611     2613#    2617     2622     2627     2633     2635
CHKITT  015766           2883     2887     2889     2891#
CHOM    001760            454#    1184     1308     1547
CHRD    002160            643#     783*     866*     867      870*     917*    2117*    2257*    2259     2261     2264*    2268*    2328*
                         2334*    2336*    2337*    2338     2344     2348     2352     2356     2376*    2476*    2477*    2478     2959*
                         2960*    2962     2963     2964*    2965*    2966     2967     3351*    3355*    3357     3384
CKABRT  017572           1836     1988     3319#
CKDAT   005220           1204     1226#
CKEOM   003564            944      948#
CKEXT   006100           1321     1346     1355     1362     1369     1375#
CKGP  = 000106            592#    2606
CKKBD   017716           2045     3350#
CKLIN   004070           1023     1025#
CKMEM   005260           1227     1237#
CKMNT   003466            931      933#
CKOFF   020560           2893     3082     3493#
CKSCRA  006736           1545     1547#
CKSCRB  007026           1563     1565#
CKSFT   015366           1081     1092     1245     1329     1368     2791#    2896     3084
CKSRC   005746           1347#    1357
CKSTR   015734           2863     2873     2882#
CKSUM = 002000            378#    1315     2696     2713
CKSUMA  005442           1296     1298#
CKSUMB  005646           1323     1331#
CKVST   016000           2881     2894#
CLFT    001766            463#    1452
CLMAIN  011512           2043     2054#
CLRCK   002014            496#
CLREG   015530            904     1229     1324     1363     1467     1776     2817     2835#    2961     3075
CLTCK   002016            499#
CMPOS   016314           1637     2979#
CNTF    020162           3361     3395#
CNTRA   027576           2072     4503#
```

```
CONSP = 000022             324#    2562     2608     2612
CONRC   013442            2423#    3254
CPABRT= 000171             594#    2628
CPPOS   016356            1669     2994#
CRBUF   001744             434#
CRCSR   001742             433#
CRRJL   016750            3097#
CRT     001762             457#    1413
CUHME   002152             619#     990      993     1635     1641
CUP     001770             466#    1425
CURER   016216             994     1012     1043     1423     1584     1684     1722     1738     2957#
CURPOS= 000004             395#    2600     2894
CURS1   003674             977      979#
CURS1A  006122            1390     1392#
CURS1B  006334            1446     1449#
CURS2   007160            1604     1606#
CUR1XT  006314            1410     1427     1434#
C2CK    007576            1661     1665     1676#
C2XT    007640            1618     1634     1655     1677     1680     1687#
MAINDEC-11-CZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 96
DZVTH.P11       CROSS REFERENCE TABLE -- USER SYMBOLS                                                        SEQ 0096

DABRT   024461            2443     4176#
DAPNT   002032             517#
DATSC   017124            1758     3166#
DBELL   026103            1928     4328#
DBOT    027176            2064     4438#
DCNTZ   026222            1899     4344#
DCONT   027126            2066     4431#
DCOPYR  024771             805     4216#
DCOUNT  017074             860*     951*    1073*    1114*    1120*    1135*    2283*    2760*    3133*    3140#
DCR     026071            1936     4326#
DCRAD   002042             530#     590      997     1064     1624
DCRST   026420            4369#
DDISP = 177570              37#     170
DELAY   017032             861      952     1074     1115     1121     1136     2284     2761     3124#
DELIM   003256             845      874#
DEMP    002022             505#    2624
DESC    025627            1915     4291#
DEVERR  026445            4374#
DEXT    026253            2051     3456     4350#
DF0     001442             207      214      229      341#
DF1     001446             258      343#
DF2     001450             236      345#
DF3     001474             243      351#
DF4     001534             264      271      294      301      308      336      359#
DF5     001537             360#
DF6     001543             222      250      278      285      315      322      329      362#
DH1     023266             205     4047#
DH10    024152             248     4133#
DH11    024723             256     4209#
DH12    024415             276      283     4168#
DH13    025533             292      299     4279#
DH14    025565             313      320     4284#
DH2     023361             212     4059#
DH3     023461             220      327     4072#
DH4     023574             227     4086#
DH5     023721             234     4103#
DH6     024005             241     4114#
DH7     024115             262      269      306      334     4128#
DISPLA  001140             170#    2105*    3560*    3582*
DISPRE  000174             138#    2105
DKBD    026502            2035     4379#
DKBERR  027363            3376     4461#
```

C08

```
DAFD    027300           2066    4451#
DAYBC   025634           1897    4295#
DLAY    002216            659#    942*    945*   1199*   1205*   1347*   1351*   1657*   1662*   2329*   2332*   2858*   2866*
                         2871*   2876*   3026*   3033*   3037*   3042*   3473*   3476*   3493*   3495*
DLERR   023020           2216    4013#
DLF     026076           1940    4327#
DLOOP   026111           2011    4331#
DLSHFT  026633           2066    4395#
DLTBL   001610            411#    706     746    2174    2248    2280    2287
DLTPT   001712            417#    746*    749     769*
DMAIN   002004            485#
DMANA   023051            689    4019#
DMANB   023151            716    4032#
DMPOCT  020064           3376#
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 87
DZVTH.P11         CROSS REFERENCE TABLE -- USER SYMBOLS                                                        SEQ 0097

DOAROW  011434           2040#   2050    2057
DPNT    001776            475#
DPRNT   026266           1966    4353#
DPRTR   024743            808    4212#
DRECT   002012            493#   1040    2745    3189
DRSHFT  026737           2066    4409#
DSCHNG= 100000            394#
DSEC    027012           2064    4417#
DSMES   017172           1898    1967    2012    2036    2052    3183#   3457
DSPCE   027254           2066    4446#
DSPC6   027472           3382    4474#
DSWR  = 177570             36#    169
DTAB    026064           1932    4325#
DTHRD   027047           2064    4423#
DTOP    026705           2064    4403#
DTTBL   011560           2040    2070#   3363
DT0     001422            206     338#
DT1     001434            235     257     340#
DT2     001452            213     242     347#
DT3     001464            263     270     349#
DT4     001500            221     249     277     284     314     321     328     353#
DT5     001512            228     355#
DT6     001524            293     300     307     335     357#
DUNTST  022656           2288    3994#
DVUNIT  024652            786    4202#
EAPNT   002030            515#
EEMP    002020            503#   2619
EINST   002034            520#
EMAIN   002002            483#   2743
EMTVEC= 000030            125#   2087*   2088*
EM1     023201            204    4038#
EM10    024354            255    4160#
EM11    024211            261    4140#
EM12    024277            268    4151#
EM13    024530            275    4184#
EM14    024577            282    4192#
EM15    025016            291    4220#
EM16    025067            298    4227#
EM17    025142            305    4235#
EM2     023331            211    4054#
EM20    025220            312    4243#
EM21    025264            319    4249#
EM22    025327            326    4255#
EM23    025415            333    4265#
EM3     023420            219    4066#
EM4     023530            226    4080#
EM5     023642            233    4095#
```

```
EM6     023752          240    4109#
EM7     024052          247    4122#
ENDPS   010474         1839    1840#
ENDSEL  010462         1823    1831    1837#
ENDTAB  001722          422#   2147
ENSRT   006712         1519    1526#
EOM   = 000004          640#    940    1178    1313    1356    1619    2348    2411    2546    2682    2931    3193
EOS     001772          469#   1765
EPL   = 001000          379#   2556    3417    3425
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 88
DZVTH.P11        CROSS REFERENCE TABLE -- USER SYMBOLS                                                              SEQ 0098

EPNT    001774          473#   1973
ERPL    002036          522#
ERRVEC= 000004          118#   3522    3523*   3525*   3528*
ERSE    010076         1753    1756#
ERSXT   010224         1772    1774    1779#
ESAMB   014636         1914*   2521    2526*   2584    2586    2590*   2596*   2613*   2640#   2944*   3424*
ESC   = 000400          380#   1912    2527    3417    3425
ESCN    002130          606#    885    1183    1307    1417    1451    1646    1764    1834    1974    2523    3427
ESCO    002056          542#    598     888    1038    1066    1112    1186    1309    1310    1648    1804    2734    2744
                       2746    3188
ESCOI = 002056          598#    981     999    1453    1462    1725
ESCP    002116          587#    599     891    1514
ESCPI = 002116          599#
ESCYI = 002042          590#
ESCZ    002124          600     602#    936     938    2777
ESCZI = 002124          600#   1185
ESSEQ   002214          657#    849*    851     907     908
ESTEX   003436          843     916#
ESTST   003100          831     833#
EXINT   013010         2281    2287#
EXIT3   004414         1086    1088    1093#
EXMAIN  011124         1901    1949#   1968    2013
EXMNT   003654          958     963#
EXTST   020414         1908    1989    3440    3450#
FADD    012176         2146#   2157
FADD1   012214         2151#   2154
FEXIT   011476         2042    2048    2051#
FNDBT   020544         3475    3483#
FRCECT  020362         3431    3433    3436#
FTEXI   003442          914     918#
FTLCNT  002176          650#    810*    912*    913    1175    1207*   1353*   1616    2441    2449*   2738*   2879*   3058*
                       3481*
FTLEXT  016070         2915    2917#
GCMD    003136          841#    882     886     889     892     895     902
GDAD    012524         2210    2224#
GDCURP  006304         1411    1430#
GDSCRL  007140         1561    1578    1580    1585#
GDSTRK  020154         3358    3392#
GETON   015256         1806    2736    2756#
GNS   = ****** U        137    3956    3957    3958    3959    3960
GOTON   015324         2759    2765#
GTCR    017364         1970    3241#   3244
GTEXT   017476         3267    3269    3272#
GTNUM   017406          694     720    3254#   3275
HDFLG   016746         2749*   3024*   3071    3074*   3088#
IABT    002024          508#
IDENT   002122          597#    779*    781*    797     803     806     867    1830
INAG    006546         1497#   1521
INITA   012620          674     710    2248#
INRPL   006510         1486    1488#
INRXT   006716         1510    1518    1528#
INTAB   001650          412#    691     717     815    2145    2173
```

E08

```
INTRC   013746      762   2507#
INTXM   014670      764   2670#
INTXT   012756     2252   2280#
INXMT   003164      850#   898
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 89
DZVTH.P11       CROSS REFERENCE TABLE -- USER SYMBOLS                                                                 SEQ 0099

IOTVEC= 000020      123#  2085*  2086*  2169*  2170*  2230*  2231*
ITSUMA  006070     1301   1371#
JMPADO  013644     2462*  2463*  2467   2469#
KBOLP   017726     3352#  3390   3394
KIENA   015126     2673   2715#
KYBO    010642     1894   1896#
KYBXMT  011110     1916   1929   1933   1937   1941   1944#
KYPO    027604     2072   4509#
KYSTRK  027440     3374*  3375*  4469#
KYSTRT  010676     1903#  1925   1946
LDADO   013040      760   2182   2253   2305#
LDBUF   017564     1336   1394   1500   1549   1568   1706   1760   3312#  3327
LDOUT   016122     2928   2931#
LDPOS   020050     3368   3372   3374#
LDXMIT  016076     1253   1302   1495   1520   1827   1900   1902   1945   1969   2014   2044   2055   2925#
                   3377   3381   3383   3387   3389
LINXT   004202     1035   1037   1044#
LKKB    002006      488#
LNFED   001752      446#   699    728   1188   1719   1938   1983   2554   3266   3270   3429
LNRW    016420     1622*  1638   1641*  1673   2979   2981*  2983   2985*  2986*  2994   2996*  2998   3000*
                   3001*  3004#
LOOP    020164     2015   3406#
LOOPR   020250     3413   3417#
LOOPT   020216     3411#  3416   3428   3435   3441
LOOPTA  020222     3412#  3420
LPSTR   020270     3418   3422#
LPTST   011344     2008   2010#
LSTST   010244     1799   1801#
MAINT = 000004      402#
MANS    002262      142    687#
MANSA   002274      689#   707    712    759   2126
MEMA    004674     1163#  1254
MEMB    004730     1169#  1176
MEMC    005102     1199#  1216   1218   1221   1225
MEMD    005150     1202   1210#
MEMXT   005422     1177   1193   1209   1248   1277#
MEM1    004666     1159   1161#
MODCA   002516      748#  1877   2459
MODCK   002502      675    726    733    746#   758
MODE    002174      649#   670*   687*   732*   753    813   1837   2124   2439
MONIT   013474     2439#  3516
MPATT = 005402     1163   1196   1247   1264   1267#
MSTBL   011532     2041   2064#
MSTRT   000204      142#
NABRT = 000170      596#  3335
NCKGP = 000107      593#  2610
NOCALC  017562     3291   3293   3306#
NOER    015504     2801   2816   2824#
NOKIL   014706     2671   2675#  2692   2694
NOROUT  020464     3451   3460#
NORXT   016074     2916   2918#
NOSHFT  017360     3230   3234#
NOSOM   014730     2676   2680#
NOVT    022727     2282   4001#
NWLN    007660     1702   1704#
OCTBIN  017342      713    735   3228#
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 90
```

```
DZVTH.P11      CROSS REFERENCE TABLE -- USER SYMBOLS              F08                              SEQ 0100

OFFLP    004520        1119#    1125
ONE      002202         652#
ONLP     004576        1132#    1140
ONOFA    004560        1123     1129#
ONOFLP   004470        1114#    1138
ONOFXT   004646        1133     1143#
ONOF61   004436        1106     1108#
PABRT  = 010000         376#    2632     3319
PATGN    005350        1172     1211     1256#
PC     =%000007          49#    1857*    1860*    1872*    3592*    3649*    3652*    3659*    3666*    3680*    3682*    3741*
PDKBO    011416        2032     2035#
PIRQ   = 177772          35#
PIRQVE=  000240         129#
PMULT    017072        2112*    2118*    2122*    3127     3139#
POPIT    003246         869#     915
POF2SP=  022626         366#    2115     2121     2153     2192     2448
PRABRT=  000172         595#    2630
PRESC    002212         656#     836*     840      854      885*     888*     891*     905
PR0    = 000000          52#
PR1    = 000040          53#
PR2    = 000100          54#
PR3    = 000140          55#
PR4    = 000200          56#
PR5    = 000240          57#
PR6    = 000300          58#
PR7    = 000340          59#
PS     = 177776          32#      33     3648
PSW    = 177776          33#    2465*
PUSH2S=  024646         367#
PWRVEC=  000024         124#    2091*    2092*    3895*    3896*    3904*    3919*    3920*
QUST     017500        3261     3263     3273#
RABT     002110         564#
RBBUF    014630         772*    1036     1198     1200     1322     1326     1361     1365     1460     1469     1562     1579     1582*
                       1666     1667     1683*    1716     1732     1757*    1767*    1773     1777     2549     2561     2637#    2886
                       2890     2940     2964     3029     3415
RBUFP    014634        1200*    1201     1656     1658     1666*    2549*    2559     2561*    2569*    2570*    2639#    2940*    3029
                       3038     3411     3415*    3419     3422*    3423*    3429*    3430*    3434*
RCRLB    027630         770      772      989     1006     1421*    3408     3453     4519#
RCUR     002050         535#    1029
RDCUR    002052         538#     982     1000     1030     1726     2598
RDENA  = 000100         395#     812      834      918     1840     2207     2215     2458     2481
REBUF    014632         770*    2559     2638#    3407*    3453*
RECAD    013650        2201     2476#
RECDN  = 000200         393#    2330     2425     2450     3241
RECEX    013236        2335     2351     2355#
RECEXA   013242        2349     2357#
RECITT   016020        1405*    1497*    1565*    2891*    2900#    2945*
RECTM    013060        2324#    2780
RECXT    014352        2573     2577     2581#
REEX     013706        2482#    2486
REOM   = 020000         375#     949      957     1132     1203     1318     1358     1359     1660     2548     2861     2872     3040
                       3062
RERR   = 100000         396#
RESET    002126         604#    2733
RESETV   015146         933      980     1025     1063     1108     1161     1298     1393     1449     1488     1547     1607     1704
                       1756     1811     2731#    3183
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 91
DZVTH.P11      CROSS REFERENCE TABLE -- USER SYMBOLS                                                SEQ 0101

RESPTR   016136         774     1093     1164     1180     1300     1375     1621     1640     1896     2010     2880     2897     2939#
                       3059     3409     3455
RESVEC=  000010         119#
```

```
REVID = 000040            383#   2566    2621    2626
RFMER = 020000            398#
RORUN = 040000            397#
ROW1    027501           2070    4477#
ROW2    027522           2070    4484#
ROW3    027542           2070    4491#
ROW4    027560           2070    4497#
RPAR  = 010000            399#
RSMAIN  011526           2054    2058#
RSOM  = 040000            374#    943     955    2543
RSTER   014316           2528    2532    2537    2540    2544    2551    2557    2572#   2591    2597    2614
RSTT  = 004000            377#   2575
RTRP    012076           2113    2119    2123#
RXOFF = 100000            373#   1075    2531    2535    2864    3031    3494
R0    =%000000             40#    672*    673*    674*    690*    694*    705*    709*    710*    713*    720*    735*    760*
                          774*    778*    856*    861*    864*    904*    933*    935*    937*    939*    941*    952*    980*
                          985*    994*   1002*   1012*   1025*   1028*   1032*   1043*   1063*   1069*   1074*   1081*   1085*
                         1092*   1093*   1108*   1113*   1115*   1117*   1121*   1130*   1136*   1161*   1164*   1168*   1172*
                         1174*   1179*   1180*   1190*   1211*   1229*   1245*   1253*   1260*   1265*   1298*   1300*   1302*
                         1306*   1314*   1320*   1324*   1329*   1336*   1345*   1363*   1368*   1375*   1393*   1394*   1407*
                         1423*   1449*   1456*   1467*   1488*   1492*   1495*   1500*   1507*   1520*   1547*   1549*   1558*
                         1568*   1575*   1584*   1607*   1610*   1615*   1620*   1621*   1633*   1637*   1640*   1654*   1669*
                         1684*   1704*   1706*   1712*   1722*   1728*   1738*   1756*   1758*   1760*   1769*   1776*   1805*
                         1806*   1811*   1817*   1822*   1827*   1835*   1836*   1869*   1872    1896*   1898*   1900*   1902*
                         1908*   1922*   1924*   1942*   1945*   1967*   1969*   1970*   1975*   1981*   1988*   1989*   2010*
                         2012*   2014*   2015*   2036*   2044*   2045*   2052*   2055*   2159*   2182*   2240*   2253*   2255*
                         2274*   2277*   2284*   2285*   2286*   2298*   2310*   2359*   2384*   2416*   2430*   2446*   2514*
                         2699*   2735*   2736*   2739    2747*   2750*   2761*   2764*   2765*   2778*   2780*   2782*   2817*
                         2828*   2839*   2878*   2880*   2893*   2896*   2897*   2899*   2914*   2918*   2929*   2932*   2933*
                         2948*   2961*   2970*   2982*   2987*   2997*   3002*   3057*   3059*   3064*   3075*   3082*   3084*
                         3086*   3102*   3137*   3154*   3169*   3183*   3197*   3221*   3235*   3245*   3254*   3272*   3306*
                         3312    3314*   3321    3327*   3333*   3337    3341*   3365*   3377*   3379*   3381*   3383*   3385*
                         3387*   3389*   3395*   3409*   3440*   3442*   3455*   3457*   3459*   3460*   3483    3486*   3498*
                         3516*   3635    3636*   3637    3640*   3697    3698*   3699*   3706*   3707*   3708*   3709*   3710*
                         3711    3716    3722    3724*   3725    3739*   3836    3846*   3850    3866    3867    3880*   3897
                         3918*   3939    3940*   3941    3942*   3943*   3944*   3945*
ROSVE   002220            659#   2325*   2358    2371*   2383    2393*   2415    2424*   2429    2792*   2827    2958*   2969
                         3470*   3485
ROSV1   015364           2775*   2781    2783#
R00C08  002150            617#
R00C11  002144            614#
R00C20  002146            616#
R00C80  002154            619#
R01C00  002132            607#   1036    1042    1732    1737
R01C20  002134            608#   1716    1721
R1    =%000001             41#    695     699     721     728     748*    833*    841     948*    953*    979*    981*    982*
                          989*    990     996*    997*    998*    999*   1000*   1006*   1008    1026*   1029*   1030*   1060*
                         1064*   1065*   1066*   1067*   1182*   1183*   1184*   1185*   1186*   1187*   1188*   1198*   1201
                         1215    1223    1228*   1230    1231    1232    1234    1304*   1307*   1308*   1309*   1310*   1311*
                         1335*   1392*   1411*   1415    1422    1424    1448*   1450*   1451*   1452*   1453*   1454*   1489*
                         1490*   1493*   1499*   1548*   1567*   1623*   1624*   1625    1626*   1627*   1643*   1644    1646*
                         1647*   1648*   1649*   1656*   1658    1678*   1679    1683    1685    1705*   1723*   1724*   1725*
                         1726*   1759*   1815*   1907*   1917    1919    1921    1926    1930    1934    1938    1976*   1979*
```

```
                         1982*   1983*   2134*   2137*   2138*   2140    2144*   2146    2147    2155    2156    2175*   2183*
                         2185*   2200*   2201*   2202*   2203*   2204*   2206*   2211*   2232*   2234*   2235*   2237    2250*
                         2306*   2308    2452*   2453*   2454    2456*   2460*   2461*   2462    2508    2509*   2510*   2513
                         2519    2523    2529    2533    2541    2546    2552    2554    2564*   2568*   2569    2572    2578
                         2582*   2588    2592    2594    2598    2602    2606    2610    2616    2619    2624    2628    2630
                         2634    2793    2796*   2800    2803    2813*   2815    2820    2821    2824*   2854*   2882    2888
                         2890*   3046    3050    3053    3067*   3069    3080*   3097*   3099*   3100    3111*   3125    3128*
                         3129*   3136*   3151*   3167*   3186*   3187*   3188*   3189*   3191*   3193*   3211*   3212*   3213*
```

```
                                                              H08
                           3214     3228*    3234     3255*    3256  [illegible]  3260     3262     3264     3266     3268     3270*    3312*
                           3326*    3362*    3363*    3364*    3378*    3384*    3721     3722*    3727     3731     3733     3738*    3837
                           3850*    3851     3855     3879*    3898     3917*
R12C00    002140            610#
R2        =%000002           42#     692*     697      701      703      719*     723      730      749*     751      769      840*     899*
                            901*    1166*    1169*    1195*    1210*    1217     1224     1237     1240     1348*    1349     1412*    1415
                           1421     1424     1608*    1611*    1625*    1635     1638*    1644*    1671     1673*    1676     1685     1897*
                           1966*    2011*    2035*    2051*    2135*    2137     2139*    2146*    2147     2149     2150     2151*    2173*
                           2176     2180*    2233*    2234     2236*    2248*    2251     2270*    2272     2287*    2290     2296     2305
                           2794     2797*    2799*    2800     2814*    2815     2825*    2855*    2882     2696*    3038     3046     3051
                           3053     3098*    3099     3103*    3104     3106*    3108     3110*    3111     3112*    3113*    3126     3127*
                           3131*    3135*    3191     3229     3231*    3232*    3233*    3234*    3456*    3838     3849*    3853*    3856
                           3863*    3864*    3865     3870*    3878*    3899     3916*
R22C00    002136            609#
R23C00    002142            612#    1065     1583
R23C78    002156            620#    1622     1626
R23C79    002044            532#     998     1008     1011     1679
R3        =%000003           43#     691*     697*     703*     704*     717*     723*     730*     731*     750*     761      762*     763*
                            764*     765*     841*     844      849      874      877      880      897      899      900*     901     1027*
                           1068*    1072*    1077*    1197*    1220     1222*    1223*    1224*    1226     1234     1252*    1301*    1305*
                           1413*    1418     1425     1491*    1494*    1519*    1816*    1826*    1899*    1901*    1915*    1928*    1932*
                           1936*    1940*    1943*    1968*    1980*    2013*    2041*    2054*    2136*    2138     2172*    2190*    2200
                           2213*    2249*    2254*    2271*    2275     2927     3054*    3152*    3168*    3301*    3302*    3304     3376*
                           3380*    3382*    3386*    3388*    3778     3787*    3793*    3794*    3797*    3802*    3803*    3804     3813*
                           3839     3847*    3848*    3862*    3865*    3874*    3875*    3877*    3900     3915*
R4        =%000004           44#     718*     724*     725      837*     847      852      858      884*     894*    1118*    1124*    1131*
                           1139*    1163*    1173     1196*    1214*    1215     1222     1256*    1257     1259*    1261*    1262     1264*
                           1312*    1322     1325     1332*    1361     1364     2040*    2205*    2209     2305*    2306     2307*    2374
                           2375*    2376     2377*    2382*    2394     2397*    2399     2400     2404     2405*    2406     2414*    2478
                           2484*    2485     2487*    2489     3019     3020*    3045*    3052     3085*    3296     3300*    3301     3303
                           3305*    3357     3362     3378     3392*    3393     3411*    3419     3779     3781*    3782*    3783*    3784
                           3785*    3799     3801*    3809*    3812*    3901     3914*
R5        =%000005           45#    1119*    1122     1134*    1137     1162*    1153     1196     1212     1246*    1247     1250     1264
                           1313*    1496*    1511     1513     1516     1517     1977*    1992     2038*    2040     2041     2049     2056
                           2145*    2150*    2158*    2174*    2224*    2227*    2396*    2399*    2406     2513*    2698*    2851     2870*
                           2874     2898*    3025*    3048     3050*    3051*    3052*    3066*    3067     3068*    3076     3077     3078
                           3150*    3151     3155*    3156     3208     3209*    3214*    3215*    3216*    3217*    3220*    3289*    3290
                           3292     3302     3303*    3304*    3305     3363     3780     3786*    3788*    3790*    3791*    3792*    3793
                           3811*    3840     3842*    3844*    3851*    3855*    3870     3876*    3902     3913*
R6        =%000006           46#      48     2080*    2081*    2082     2189*    2190     2191     2397     2427*    2428*    2447
R7        =%000007           47#      49
SEQ4      003344            848      897#
SETA      011604            671      688     2079#    2445
SETREV    005416           1252     1275#    1826
SHFTA     027600           2072     4505#
SLSH    = 000057            591#    2338     2602
SOM     = 000002            639#     934     1167     1609     2344     2395     2541     2677     2925     3187
SP        =%000006           48#     766*     767*     779      783      787*     792*     797*     839*     851*     854*     866      870
MAINDEC-11-DZVTH-A        MACY11 27(732)  20-SEP-76  10:22  PAGE 93
DZVTH.P11         CROSS REFERENCE TABLE -- USER SYMBOLS                                                                               SEQ 0103

                            917      993*    1011*    1042*    1080*    1083*    1084*    1091*    1110*    1111*    1112*    1244*    1318*
                           1319*    1328*    1343*    1344*    1367*    1422*    1583*    1631*    1632*    1652*    1653*    1682*    1721*
                           1737*    1802*    1803*    1804*    1820*    1821*    1832*    1833*    1834*    1864*    1972*    1973*    1974*
                           2084*    2099*    2100*    2106     2107     2108     2165*    2166*    2197*    2198*    2217*    2257     2264
                           2268     2272*    2273*    2275*    2276*    2290*    2325     2326     2327*    2352*    2356*    2358*    2371
                           2372     2373     2374*    2382     2383*    2393     2415*    2424     2429*    2466*    2467*    2508*    2582
                           2732*    2733*    2734*    2742*    2743*    2744*    2745*    2746*    2775     2776*    2777*    2779*    2781*
                           2792     2796     2827*    2851*    2895*    2898     2912*    2913*    2958     2959     2969*    3019*    3062*
                           3063*    3083*    3085     3125*    3126*    3135     3136     3208*    3220     3255     3470     3471     3472
                           3485*    3522*    3525     3527     3528     3556     3557     3561*    3587     3603*    3606*    3635*    3636
                           3637*    3639     3640     3641*    3643     3645     3647     3648*    3653     3655*    3657*    3665*    3669
                           3673     3674     3678     3697*    3702*    3721*    3727*    3731*    3738     3739     3770*    3771     3772
                           3773*    3778*    3779*    3780*    3786     3811     3812     3813     3814*    3815*    3836*    3837*    3838*
```

```
                         3839*   3840*   3841*   3842    3845*   3858    3860*   3862    3872    3874    3876    3877    3878
                         3879    3880    3882*   3883*   3897*   3898*   3899*   3900*   3901*   3902*   3903    3909*   3913
                         3914    3915    3916    3917    3918    3939*   3940
SPCB      027602         2070    4507#
SPTN      003336          846     894#
STACK  =  001100           29#   2084
START     000200          141#
STKLMT=   177774           34#
STRBYT    015756         2885    2888#
STRO      014664         2634*   2653#   3335
STRP      014666         2616*   2654#
STRTAB    001714          419#   2146
STTEP     014662         2576    2578*   2579*   2652#   2814    2826*   2942*
STTER     014640         2642#   2813    2826    2942
STUPM     022542         2109    3978#
SVER1     002162          644#   1943    2793*   2824    3214*   3215*   3216*   3366*   3367    3369    3371*   3373*   3374
                         3375    3380    3386
SVER2     002164          645#   2794*   2825    3219*   3366
SWR       001136          169#   2102    2104*   3517    3531    3533    3539    3546    3583    3590    3595    3601
SWREG     000176          139#   2104
SW0    =  000001           87#
SW00   =  000001           77#     87
SW01   =  000C02           76#     86
SW02   =  000004           75#     85
SW03   =  000010           74#     84
SW04   =  000020           73#     83
SW05   =  000040           72#     82
SW06   =  00C100           71#     81
SW07   =  000200           70#     80
SW08   =  000400           69#     79
SW09   =  001000           68#     78
SW1    =  000002           86#
SW10   =  002000           67#
SW11   =  004000           66#
SW12   =  010000           65#
SW13   =  020000           64#
SW14   =  040000           63#
SW15   =  100000           62#
SW2    =  000004           85#
SW3    =  000010           84#
SW4    =  000020           83#
SW5    =  000040           82#
SW6    =  000100           81#
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 94
DZVTH.P11        CROSS REFERENCE TABLE -- USER SYMBOLS                                                                 SEQ 0104

SW7    =  000200           80#
SW8    =  000400           79#
SW9    =  001000           78#
TAB       001754          448#   1930
TBBUF     015136          773*    979     996    1026    1060    1182    1304    1335    1392    1448    1489    1494    1499
                         1548    1567    1623    1643    1676    1678    1705    1723    1759    1815    1976    1979    2691
                         2703    2714    2719#   2941    3167    3186    3326    3408*   3454*
TBITVE=   000014          120#
TBLCK     012226         2148    2155#
TBUFP     015142          934*    936*    938*    940*   1116*   1129*   1167*   1173*   1178*   1609*   1614*   1619*   1921*
                         1923*   2685    2689    2691*   2693*   2698    2700    2701    2703*   2705*   2714*   2721#   2925*
                         2927*   2931*   2941*
TCOMB  =  000104          403#   2208    2214    2480
TCRLB     030330          771     773    1197    1220    1226    1228    1406    1412    1498    1511    1566    1582    1671
                         3025    3048    3066    3068    3097    3100    3407    3454    4521#
TCRLF  =  000200         3645    3625#
TCUCH     002066          549#   1454    1465
TDATA     006706         1511    1524#
```

```
TDEV    012356       2169     2188#
TEBUF   015140        771*    2689     2701     2720#
TENA  = 000100        401#    1071     1317     1342     1630     1651     1819     1985     2184     2188     2483     2490     2538
                     2715     2856     2910     2939     3023     3171     3194     3439
TESC    013322        956     1113     1805     1835     1975     2392#    2735     2747     2778
TFUNCT  006700       1513     1523#
THT   = 000011       3643     3684#
TIMEXT  020546       3482     3484#
TKVEC = 000060        127#
TOADD   002204        653#    2373*    2375
TOFF    013672       2480#    2488
TOTCH = 003600        405#    1166     1217     1237     1239     1608     3170
TOTC1 = 003601        406#
TPENT   012222       2143     2153#
TPREG   002210        655#
TPRNT   011146       1964     1966#
TPVEC = 000064        128#
TRAPVE= 000034        126#    2089*    2090*
TRDY  = 000200        400#    2402     2409     2412
TRMID = 000002        386#    2604
TRPA    012042       2110     2115#
TRPB    012066       2116     2121#
TRPE    012202       2147#    2152
TRPVEC  012122        672      690     2134#
TRTVEC= 000014        121#
TSMAD   013732       2203     2489#
TSTADD  022472       2462     3962#
TSTER   002120        589#    1803
TSTNM   002226        338      349      353      355      357      662#     835*    2731*
TSTPTR  002172        648#     756      815*    2456     2464*
TST1    003062        829#    3962
TST10   005424       1294#    3966
TST11   006104       1388#    3966
TST12   006316       1444#    3968
TST13   006472       1484#    3968
TST14   006720       1543#    3968
TST15   007142       1602#    3970
TST16   007642       1700#    3970
```

K08

```
TST17    010060                1751#   3970
TST2     003450                 929#   3962
TST20    010226                1797#   3972
TST21    010624                1892#   3972
TST22    011130                1962#   3972
TST23    011326                2006#   3974
TST24    011400                2030#   3974
TST3     003656                 975#   3962
TST4     004052                1021#   3964
TST5     004204                1057#   3964
TST6     004420                1104#   3964
TST7     004650                1157#   3966
TXEX     015134                2711    2716#
TXRCK    002074                 554#   1311
TXSUM  = 000100                 382#   1334    2511    2550
TXTCK    002102                 559#
TYPDS  = 104404                1865    3732    3960#
TYPE   = 104400                 689     716     785     786     802     805     808    1863    1866    2109    2216    2222    2282
                               2288    2295    2443    3265    3274    3585    3593    3696    3713    3715    3718    3720    3735
                               3740    3805    3881    3921    3956#
TYPNUM   017514                3264*   3265    3273*   3274    3276#
TYPOC  = 104401                3704    3728    3957#
TYPON  = 104403                3959#
TYPOS  = 104402                 789     794     799    2219    2292    3958#
TYP6     002170                 647#
TIERR    003360                 868     904#
UNLKKB   002010                 490#
UPD4     013710                2479    2483#
VBIT     020554                3472*   3474    3487#
VDLAY    020556                3471*   3478*   3488#
VECPT    001710                 416#    747*    750     792     809*
VECT     001740                 432#    761*   1809
VRBUF    001732                 429#    862    2225*   2336    2476    2509    3243
VRCSR    001730                 428#    748     787     812*    834*    918*   1808    1840*   2175    2207*   2215*   2217    2224
                               2250    2330    2458*   2481*   2818    2819    3241
VSTAT    002222                 660#    777*    943     949     955     957     960    1062*   1075    1109*   1126    1132    1141
                               1165*   1194*   1203    1303*   1315*   1331*   1334*   1358*   1359    1606*   1629*   1645*   1660
                               1813*   1903*   1912    1944*   1978*   1986    2511    2527*   2531*   2535*   2536    2539*   2543*
                               2548*   2550*   2556*   2562    2566    2575*   2600*   2604*   2608*   2612*   2621*   2626*   2632*
                               2670    2672*   2696    2712*   2713*   2748*   2797    2804    2853*   2857*   2859    2861    2864
                               2872    2884    2908*   2917*   3022*   3027    3031    3040    3166*   3173    3184*   3195    3319
                               3323    3339    3410*   3412    3414*   3417    3425*   3474    3494
VVECT    001550                 410#    747    2172    2249
VXBUF    001736                 431#   2308    2395*   2404*   2411*   2489*   2677*   2682*   2700*
VXCSR    001734                 430#   1071*   1317*   1342*   1630*   1651*   1819*   1985*   2184*   2188*   2208*   2214*   2402
                               2409    2412    2480*   2483*   2490*   2538*   2715*   2856*   2910*   2939*   3023*   3171*   3194*
                               3439*
WDSTOR   016022                1406*   1566*   2855    2901#   2946*
WTBGND   020470                1085    1320    1345    1633    1654    1822    2914    3064    3469#
XMAD2    015716                2868    2878#
XMCNT    015144                 983*   1001*   1031*   1070*   1189*   1316*   1341*   1404*   1455*   1506*   1557*   1574*   1628*
                               1650*   1711*   1727*   1768*   1818*   1984*   2680    2707*   2723#   2909*   2943*   3170*   3185*
                               3190*   3332*   3426*   3432    3436*   3437
XMDNE  = 000001                 387#   1083    1343    1631    1652    1820    1986    2712    2859    2912    3027    3173    3195
                               3412    3414
XMITT    015606                2857#   2892
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 96
DZVTH.P11        CROSS REFERENCE TABLE -- USER SYMBOLS                                                                SEQ 0106

XMIT1    016026                 935     937     939     941    1117    1130    1168    1174    1179    1610    1615    1620    1922
                               1924    2908#   2929    2932
XMFIL  = 000200                 381#   2536    2539    2672
```

L08

```
XMREC   015552              985      1002      1032      1190      1407  [illegible]     1507      1558      1575      1712      1728      1769      2850#
                           3333
XMTAL   002060              544#     1067      1111      1187      1649
XMWT    020404             3438      3440#
XMXT    016016             2899#
XMZER   020466              776*     2687      3406*     3458*     3462#
XOFF  = 000023              641#     1116      2529      3292
XON   = 000021              642#     1087      1129      2533      2758      3290
XRCMP   016422             3018#
XREC    016524             3028      3030      3037#     3056
XRERR   016610             3035      3043      3057#
XREXT   016626             3041      3055      3061#
XROUT   016726             3060      3065      3070      3082#
ZERO    002166              646#      704       839      1110      1579      1773      1802      1832      1972      2327      2732      2742      2776
ZERST   003126              838#      872
ZFLAG   015326              778       864      2255      2774#
$BDADR  001122              163#      347      2821*     2836*     2967*
$BDDAT  001126              165#      338       340       347       353       355       907*      910*      960*     1040*     1231*     1240*     1326*
                           1365*     1419*     1464*     1465*     1469*     1513*     1719*     1735*     1766*     1777*     1809*     2082      2804*
                           2820*     2838*     2966*     3077*     3323*     3339*
$BELL   001162              181#     3585      3609
$CHARC  021444             3660*     3667      3676*     3681#
$CMTAG  001100              151#     2080      2087      2093      2094      2095
$CM3  = 000000              179#
$CRLF   001167              183#      785       802      2222      2295      3388      3593      3609      3650      3687      3696      3715      3720
                           3740
$DBLK   022266             3847      3881      3889#
$DOAGN  010600             1859      1870      1876#
$DTBL   022256             3850      3885#
$ENDAD  010570             1872#     3598
$ENDCT  010536             1661#     2093
$ENDMG  010604             1863      1878#
$ENULL  010621             1866      1881#
$EOP    010502             1851#
$EOPCT  010530             1858#     1862      2093*
$ERFLG  001103              154#     3506      3535      3537      3543*     3564      3580*     3609
$ERMAX  001115              160#     2096*     3537      3559*     3564
$ERROR  021060             2087      3579#
$ERRPC  001116              161#      338       349      3587*     3588*     3589      3609      3702
$ERRTB  001172              201#     3710
$ERRTY  021450             3592      3695#
$ERTTL  001112              158#     3586*     3609
$ESCAP  001160              180#     2095*     3558*     3604      3606      3609
$FILLC  001154              177#     3653      3687
$FILLS  001153              176#     3687
$GDADR  001120              162#      347       355       357      1126*     1141*     1232*     2739*     2818*     2835*     2963*     3078*
$GDDAT  001124              164#      338       340       347       353       355       905*      906*      959*     1038*     1039*     1230*     1239*
                           1325*     1364*     1417*     1418*     1462*     1463*     1468*     1514*     1718*     1734*     1764*     1765*     1808*
                           2803*     2819*     2837*     2962*     3076*     3321*     3322*     3337*     3338*
$GET42  010560             1867#
$HD   = 000000               13
$ICNT   001104              155#     3550*     3551      3553*     3563
$ILLUP  022420             3895      3924#
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 97
DZVTH.P11        CROSS REFERENCE TABLE -- USER SYMBOLS                                                                                     SEQ 0107

$ITEMB  001114              159#     3589*     3609      3699
$LF     001170              184#     3609      3687
$LPADR  001106              156#      831*      931*      977*     1023*     1059*     1106*     1159*     1296*     1390*     1446*     1486*     1545*
                           1604*     1702*     1753*     1799*     1894*     1964*     2008*     2032*     2097*     3541*     3556*     3561      3563
$LPERR  001110              157#     2098*     3541      3557*     3563      3603
$MAIL = ****** U            832       932       978      1024      1060      1107      1160      1297      1391      1447      1487      1546      1605
                           1703      1754      1800      1895      1965      2009      2033      2109      3556      3595      3637
$MXCNT  021056             3554      3563#
```

M08

```
$NULL   001152                   175#   3655    3687
$NWTST= 000000                   828#    928#    974#   1020#   1056#   1103#   1156#   1293#   1387#   1443#   1483#   1542#   1601#
                                1699#   1750#   1796#   1891#   1961#   2005#   2029#
$OCNT   022046                  3777*   3806*   3819#
$OMODE  022050                  3772*   3776*   3781    3784*   3795*   3821#
$OVER   021042                  3518    3534    3542    3552    3560#
$PASS   001100                   152#    349     353     357    1855*   1856*   1864    1878    3548    3564
$POWER  022426                  3922    3927#
$PWRDN  022276                  2091    3895#   3919
$PWRMG  022414                  3922#
$PWRUP  022344                  3904    3909#
$QUES   001166                   182#   3609    3687
$RDCHR= ****** U                3961
$RDDEC= ****** U                3961
$RDLIN= ****** U                3961
$RDOCT= ****** U                3961
$R2A   = ****** U               3961
$SAVRE= ****** U                3961
$SAVR6  022424                  3903*   3909    3910*   3911*   3926#
$SCOPE  020604                  2085    2230    3515#
$SETUP= 000037                   665#   1853    2085    2087    2089    2091    2093    2094    2095    2097    3598
$STUP = 177777                   665#
$SVLAD  021014                  3526    3555#
$SWR   = 167400                    3#     13      18      19      20      21      22      23      24     179     180     181     830
                                 930     976    1022    1058    1105    1158    1295    1389    1445    1485    1544    1603    1701
                                1752    1798    1848    1854    1868    1877    1878    1893    1963    2007    2031    2094    2095
                                2097    2098    3507    3508    3509    3510    3511    3517    3529    3531    3532    3535    3536
                                3537    3544    3545    3546    3557    3560    3563    3571    3572    3573    3574    3575    3583
                                3590    3595    3601    3609    3923
$SWRMK= 000000                    24      25    3511    3512    3533
$TIMES  001156                   179#    830*    930*    976*   1022*   1058*   1105*   1158*   1295*   1389*   1445*   1485*   1544*
                                1603*   1701*   1752*   1798*   1854*   1893*   1963*   2007*   2031*   2094*   3544*   3551    3554*
                                3563
$TKB    001144                   172#   2427    2452
$TKS    001142                   171#   2425    2450
$TN    = 000025                    3#     13     828     830#    928     930#    974     976#   1020    1022#   1056    1058#   1103
                                1105#   1156    1158#   1293    1295#   1387    1389#   1443    1445#   1483    1485#   1542    1544#
                                1601    1603#   1699    1701#   1750    1752#   1796    1798#   1891    1893#   1961    1963#   2005
                                2007#   2029    2031#
$TPB    001150                   174#   3673*   3687
$TPFLG  001155                   178#   3631    3687
$TPS    001146                   173#   3671    3687
$TRAP   022436                  2089    3939#
$TRP   = 000005                 3947#   3957#   3958#   3959#   3960#   3961#
$TRPAD  022460                  3944    3955#
$TSTNM  001102                   153#    835    1853*   2731    3506    3533    3555*   3560    3564    3582    3609
$TYPBN= ****** U                3961
$TYPDS  022052                  3835#   3960
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 98
DZVTH.P11       CROSS REFERENCE TABLE -- USER SYMBOLS                                                               SEQ 0108

$TYPE   021226                  3631#   3649    3947    3956
$TYPEC  021400                  3652    3659    3666    3671#   3672
$TYPEX  021446                  3677    3679    3682#
$TYPOC  021650                  3775#   3957
$TYPON  021664                  3774    3777#   3959
$TYPOS  021624                  3770#   3958
$XTSTR  020620                  3520#
$OFILL  022047                  3771*   3775*   3785    3820#
$4OCAT= ****** U                3517    3592
.      = 031116                  133#    137#    140#    150#    185     364#    410#    411#    412#    871    1267    1360    1878
                                1882    1987    2083    2097    2098    2186    2403    2410    2413    2426    3130    3174    3242
                                3313    3353    3563    3564    3609    3687    3743#   3889#   3906    3925    4201#   4519#   4521#
                                4523#
```

```
.BEL   = 000007             441#   1401    1554    1709
.CARRT= 000015              443#
.CDWN  = 000102             459#   1395    1504    1550
.CHOM  = 000110             453#   1501    1554    1572    1761
.CLFT  = 000104             462#   1398    1550
.CLRCK= 000133              495#   1371
.CLTCK= 000134              498#   1337
.CPYSC= 000135              575#   1833
.CRT   = 000103             456#   1395
.CUP   = 000101             465#   1401
.C00   = 000040             626#   1550
.C03   = 000043             627#
.C08   = 000050             628#
.C11   = 000053             629#
.C20   = 000064             630#   1707
.C21   = 000065             631#
.C40   = 000110             632#
.C79   = 000157             633#   1569
.DAPNT= 000171              516#
.DCRAD= 054433              529#
.DEMP  = 000152             504#
.DISAC= 000137              577#
.DMAIN= 000141              484#   1949    2058
.DPNT  = 000130             474#
.DRECT= 000103              492#   1371
.EAPNT= 000131              514#
.EEMP  = 000112             502#   1275
.EINST= 000111              519#   1523    1526
.EMAIN= 000101              482#
.ENAC  = 000136             576#
.EOS   = 000112             468#   1761
.EPNT  = 000127             472#
.ERPL  = 000151             521#   1523
.ESC   = 000033             585#   1275    1337    1371    1395    1398    1401    1501    1504    1526    1550    1554    1569
                           1572    1707    1709    1761    1949    2058    3328    3330
.ESCZ  = 055033             601#
.IABT  = 000137             507#   3328
.LKKB  = 000105             487#
.LNFED= 000012              445#   1554    1709
.O      = 000117            541#   1275    1337    1371    1395    1398    1401    1501    1504    1554    1569    1572    1709
                           1761    1949    2058    3328    3330
.P      = 000120            586#   1526
.PSCRN= 000150              578#
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 99
DZVTH.P11        CROSS REFERENCE TABLE -- USER SYMBOLS                                                                  SEQ 0109

.RABT  = 000140             563#   3328
.RDCUR= 000131              537#   1395    1398    1401    1569    1709
.RESET= 000122              603#
.R00   = 000040             621#   1707
.R00C1= 025440              613#
.R00C2= 032040              615#
.R01   = 000041             622#
.R12   = 000054             623#
.R22   = 000066             624#
.R23   = 000067             625#   1550    1569
.R23C0= 020067              611#
.R23C7= 067467              531#
.TAB   = 000011             447#
.TCUCH= 000127              548#   1501    1504    1554    1572
.TSTEP= 000124              588#
.TXRCK= 000135              553#
.TXTCK= 000136              558#   1337
.UNLKK= 000145              469#   3330
```

```
.XMTAL= 000126             543#    1337    1761
.X    = 000131             536#    1550    1569    1707
MAINDEC-11-DZVTH-A       MACY11 27(732)  20-SEP-76  10:22  PAGE 101                                               SEQ 0110
DZVTH.P11       CROSS REFERENCE TABLE -- MACRO NAMES

COMMEN     130#
ENDCOM     130#
ERROR       30#     911     961     967     992    1004    1010    1034    1041    1079    1090    1127    1142    1192    1208
          1223     1241    1327    1354    1366    1409    1420    1458    1466    1470    1509    1515    1560    1564    1577
          1581     1664    1691    1714    1720    1730    1736    1771    1775    1778    1810    2740    2805    2822    2968
          3073     3079    3324    3340    3480
ESCAPE     130#
MULT       130#
NEWTST     130#     828     928     974    1020    1056    1103    1156    1293    1387    1443    1483    1542    1601    1699
          1750     1796    1891    1961    2005    2029
POP        130#     779     782     865     869     916    2256    2263    2267    2324    2326    2370    2372    2373    2381
          2392     2423    2581    2774    2791    2796    2898    2957    2959    3085    3135    3220    3255    3469    3471
          3472     3876    3913
PUSH       130#     838     850     854     993    1011    1042    1080    1082    1084    1091    1110    1111    1112    1243
          1318     1319    1328    1343    1344    1367    1422    1583    1631    1632    1652    1653    1682    1721    1737
          1801     1803    1804    1820    1821    1832    1833    1834    1971    1973    1974    2272    2273    2275    2276
          2327     2352    2355    2357    2374    2383    2415    2429    2507    2732    2733    2734    2741    2743    2744
          2745     2746    2776    2779    2781    2827    2850    2894    2911    2913    2969    3018    3061    3063    3083
          3124     3207    3484    3835    3897
SCOPE       31#     829     929     975    1021    1057    1104    1157    1294    1388    1444    1484    1543    1602    1700
          1751     1797    1852    1892    1962    2006    2030
SETTRA    3947#    3957    3958    3959    3960
SETUP      130#    2079
SKIP       130#
SLASH      130#
SPACE      130#
STARS      130#     143     185     369     371     389     391     407     409     425     427     437     439     449     451
           477      479     510     512     525     527     567     569     571     573     581     583     635     638     665
           668      678     685     738     744     817     818     825     828     920     926     928     964     971     974
          1014     1018    1020    1047    1054    1056    1095    1101    1103    1145    1154    1156    1279    1291    1293
          1377     1385    1387    1436    1441    1443    1473    1481    1483    1530    1540    1542    1588    1599    1601
          1689     1697    1699    1742    1748    1750    1783    1794    1796    1841    1882    1889    1891    1952    1959
          1961     1996    2003    2005    2018    2027    2029    2075    2077    2128    2132    2160    2163    2193    2196
          2242     2246    2299    2303    2312    2322    2362    2368    2386    2390    2418    2421    2432    2437    2470
          2474     2493    2505    2656    2669    2724    2729    2752    2754    2767    2772    2785    2789    2808    2811
          2830     2833    2840    2848    2904    2906    2920    2923    2935    2937    2951    2955    2972    2974    2977
          2989     2992    3005    3016    3090    3095    3116    3122    3142    3148    3161    3164    3176    3181    3199
          3206     3223    3226    3237    3239    3248    3252    3280    3287    3307    3310    3315    3317    3343    3347
          3397     3404    3444    3448    3463    3467    3489    3491    3500    3564    3609    3687    3744    3822    3890
          3930
TRMTRP    3947#
TYPBIN     130#
TYPDEC     130#    1864    3730
TYPNUM     130#
TYPOCS     130#     787     792     797    2217    2289
TYPOCT     130#    3702    3727
TYPTXT     130#
$$CMRE     143#
$$CMTM     143#
$$ESCA     130#
$$NEWT     130#     828     928     974    1020    1056    1103    1156    1293    1387    1443    1483    1542    1601    1699
          1750     1796    1891    1961    2005    2029
$$SET     3947#    3957    3958    3959    3960
$$SKIP     130#
.EQUAT       3#      25
.HEADE       3#
MAINDEC-11-DZVTH-A       MACY11 27(732)  20-SEP-76  10:22  PAGE 102                                               SEQ 0111
DZVTH.P11       CROSS REFERENCE TABLE -- MACRO NAMES
```

C09

```
.SETUP      3#    665
.SWRHI      3#     13
.SWRLO      3#     25#
.$CATC      3#    130
.$CMTA      3#    143
.$EOP       3#   1841
.$ERRO      3#   3564
.$ERRT      3#   3687
.$POWE      3#   3890
.$SAVE      3#
.$SCOP      3#   3500
.$TRAP      3#   3930
.$TYPD      3#   3822
.$TYPE      3#   3609
.$TYPO      3#   2744
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 104
DZVTH.P11        CROSS REFERENCE TABLE -- PERMANENT SYMBOLS                                              SEQ 0112

ADD        809    1246    2139    2151    2180    2236    2254    2285    2307    2463    2579    2764    2879    3057    3234
          3459    3641    3710    3773    3783    3855
ASL       2461    3231    3232    3233    3707    3708    3709    3943
ASLB      2484    3860
ASRB      3211    3212    3213
BCC       3861
BEQ        702     707     726     804     807     814     842     845     853     875     878     881     898     909     956
           991    1009    1037    1088    1123    1170    1216    1218    1227    1238    1248    1323    1360    1362    1416
          1461    1512    1518    1563    1580    1612    1636    1677    1680    1717    1733    1774    1831    1838    1870
          1913    1987    2042    2125    2157    2177    2210    2252    2341    2349    2351    2398    2401    2403    2407
          2410    2413    2426    2451    2455    2479    2486    2512    2520    2537    2553    2563    2567    2577    2593
          2629    2676    2697    2708    2759    2801    2816    2862    2865    2883    2885    2889    2928    2980    2984
          2995    2999    3032    3043    3047    3055    3070    3157    3174    3196    3230    3242    3257    3259    3267
          3291    3293    3320    3336    3353    3358    3368    3413    3420    3428    3495    3532    3534    3536    3540
          3549    3581    3584    3602    3605    3644    3712    3717    3723    3734    3800
BGE       3552    3679
BGT       1859    2457    3807    3869
BHI       1920    2262    2524    3263    3438    3538
BHIS       914    1221    3049    3101    3361
BIC        834    1109    1165    1194    1256    1303    1331    1358    1606    1629    1645    1813    1840    1856    1903
          1944    1978    2188    2214    2215    2337    2428    2453    2458    2477    2480    2481    2490    2510    2535
          2539    2550    2612    2626    2713    2715    2799    2853    2857    2908    2939    3022    3166    3184    3228
          3289    3302    3303    3410    3414    3425    3797
BICB      3215
BIS        812     918    1071    1214    1315    1317    1334    1342    1630    1651    1819    1985    2184    2207    2208
          2483    2527    2531    2538    2543    2548    2556    2568    2575    2590    2596    2600    2604    2608    2621
          2632    2672    2712    2856    2910    3023    3171    3194    3304    3432    3802    3803    3863    3864
BISB      3216    3699
BIT        803     806     943     949     955     957    1075    1132    1203    1359    1660    1830    1912    1986    2156
          2330    2402    2409    2412    2425    2450    2485    2511    2521    2536    2562    2566    2584    2586    2696
          2859    2861    2864    2872    2884    3027    3031    3040    3173    3195    3241    3319    3412    3417    3474
          3494    3517    3531    3539    3546    3583    3590    3601
BITB      3053    3667
BLO        845    1176    1235    1263    1617    1659    1686    1918    2047    2148    2233    2260    2343    2442    3030
          3039    3261    3370    3433
BLOS      1672
BLT       2141    2309    3658    3808    3852    3868
BMI        843     848    1258    2043    2178    2258    3859
BNE        696     700     722     729     752     754     784     868     871     944     946     950     954     958    1076
          1078    1125    1133    1138    1140    1202    1204    1206    1213    1251    1350    1352    1357    1426    1661
          1663    1668    1825    1905    1927    1931    1935    1939    1993    2083    2186    2212    2265    2269    2281
          2297    2331    2333    2339    2345    2378    2380    2440    2517    2522    2525    2530    2534    2542    2547
          2555    2560    2585    2587    2589    2595    2599    2603    2607    2611    2620    2625    2631    2681    2686
          2688    2690    2702    2711    2763    2860    2867    2873    2875    2877    2892    3028    3034    3041    3072
          3081    3105    3109    3130    3132    3134    3153    3192    3244    3269    3313    3394    3418    3451    3475
          3477    3479    3497    3518    3547    3591    3599    3638    3646    3654    3668    3675    3700    3726    3798
```

D09

```
            3857    3912
BFL          757     760     859    1249    2573    2671    3218    3297    3596    3632    3672    3796    3843    3873
BR           676     698     709     712     714     727     733     736     758     872     882     886     889     892     895
             902     915     916     988    1003    1005    1033    1035    1086    1191    1193    1209    1225    1321    1346
            1355    1369    1408    1410    1427    1457    1459    1508    1510    1521    1559    1561    1576    1578    1624
            1639    1655    1665    1670    1674    1713    1715    1729    1731    1770    1772    1807    1823    1829    1909
            1916    1925    1929    1933    1937    1941    1946    1990    1994    2048    2050    2057    2103    2113    2119
            2152    2154    2181    2187    2223    2226    2266    2278    2335    2347    2353    2408    2488    2528    2532
            2540    2544    2551    2557    2591    2597    2601    2605    2609    2614    2617    2622    2627    2633    2635
            2673    2684    2692    2694    2704    2737    2823    2863    2868    2881    2897    2915    2916    2926    2930
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 105
DZVTH.F11        CROSS REFERENCE TABLE -- PERMANENT SYMBOLS                                                   SEQ 0113

            3035    3044    3056    3060    3065    3107    3114    3158    3210    3271    3275    3372    3390    3416    3421
            3431    3435    3441    3482    3520    3526    3529    3542    3545    3634    3651    3661    3670    3677    3705
            3729    3736    3774    3789    3810    3854    3871    3906    3925
CLC         3295
CLR          670     692     718     719     766     775     776     777     781     810     836     837     942    1062    1162
            1195    1199    1312    1332    1347    1419    1464    1496    1657    1718    1734    1757    1766    1767    1812
            1853    1854    1914    2037    2038    2081    2094    2095    2144    2158    2165    2197    2206    2225    2227
            2229    2235    2328    2329    2449    2465    2466    2487    2613    2748    2749    2835    2836    2837    2838
            2858    2871    2917    2943    2944    2946    2947    3020    3024    3026    3037    3351    3424    3458    3473
            3493    3544    3558    3698    3787    3846    3849    3910
CLRB         731    1493    1911    2678    2683    2757    3350    3356    3543    3676    3875
CMP          725     913     990    1008    1036    1122    1137    1175    1201    1217    1220    1226    1234    1237    1349
            1415    1424    1425    1511    1517    1616    1635    1658    1671    1676    1679    1685    1716    1732    2082
            2106    2140    2147    2237    2308    2441    2516    2559    2576    2689    2701    2800    2815    2874    2882
            3029    3038    3048    3100    3360    3419    3432    3437    3527    3551    3598    3867
CMPB         695     699     721     728     844     867     874     877     880    1087    1212    1215    1250    1262    1322
            1356    1361    1460    1562    1579    1667    1773    1917    1919    1926    1930    1934    1938    1992    2046
            2259    2261    2338    2342    2344    2348    2406    2454    2478    2523    2529    2533    2541    2546    2552
            2554    2588    2592    2594    2598    2602    2606    2610    2619    2624    2628    2630    2756    2979    2983
            2994    2998    3046    3104    3108    3156    3243    3256    3258    3260    3262    3266    3268    3290    3292
            3335    3357    3367    3369    3427    3450    3533    3537    3643    3645    3653    3674    3678
DEC          945     953    1077    1124    1139    1169    1205    1351    1611    1662    1857    2185    2211    2332    2379
            2460    2707    2762    2866    2876    2891    3033    3042    3054    3129    3131    3133    3152    3217    3434
            3476    3478    3496    3706
DECB        2340    2350    2981    2985    3657    3660    3795    3806
EMT           30
HALT         137    3597    3600    3633    3905    3924
INC          724     912    1207    1210    1353    1490    1828    1855    2464    2515    2570    2693    2705    2738    2879
            3045    3058    3074    3190    3359    3364    3392    3406    3423    3426    3430    3436    3481    3550    3586
            3801    3809    3853    3911
INCB        1261    2346    2996    3000    3103    3112    3155    3555    3580    3680
IOT           31
JMP          141     142     671     675     688     711     755     759    1177    1254    1618    1839    1877    1910    1991
            2016    2053    2126    2127    2445    2459    2469
JSR          672     673     674     690     694     705     709     710     713     720     735     760     774     778     856
             861     864     904     933     935     937     939     941     952     980     985     994    1002    1012    1025
            1028    1032    1043    1063    1069    1074    1081    1085    1092    1093    1108    1113    1115    1117    1121
            1130    1136    1161    1164    1168    1172    1174    1179    1180    1190    1211    1229    1245    1253    1298
            1300    1302    1306    1314    1320    1324    1329    1336    1345    1363    1368    1375    1393    1394    1407
            1423    1449    1456    1467    1488    1492    1495    1500    1507    1520    1547    1549    1558    1568    1575
            1584    1607    1610    1615    1620    1621    1633    1637    1640    1654    1669    1684    1704    1706    1712
            1722    1728    1738    1756    1758    1760    1769    1776    1805    1806    1811    1817    1822    1827    1835
            1836    1872    1896    1898    1900    1902    1908    1922    1924    1942    1945    1967    1969    1970    1975
            1981    1988    1989    2010    2012    2014    2015    2036    2044    2045    2052    2055    2182    2253    2255
            2274    2277    2284    2514    2699    2735    2736    2747    2761    2778    2780    2817    2880    2893    2896
            2897    2914    2929    2932    2961    3059    3064    3075    3082    3084    3169    3183    3254    3327    3333
            3365    3377    3379    3381    3383    3385    3387    3389    3409    3440    3455    3457    3516    3592    3649
            3652    3659    3666
MOV          687     691     697     703     704     717     746     747     748     749     750     761     762     763     764
             765     767     769     770     771     772     773     779     783     787     792     797     811     815     830
             831     833     839     840     841     849     851     854     860     866     870     884     885     888     891
             894     905     907     917     930     931     948 [illegible]    959     960     976     977     979     981     983
             989     993     996     997     998     999    1000    1001    1006    1011    1022    1023    1025    1027    1029
            1030    1031    1039    1040    1042    1058    1059    1060    1061    1064    1065    1066    1067    1068    1070
            1072    1073    1080    1083    1084    1091    1105    1106    1110    1111    1112    1114    1118    1119    1120
            1126    1131    1134    1135    1141    1158    1159    1163    1166    1182    1183    1184    1185    1186    1187
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 106
DZVTH.P11        CROSS REFERENCE TABLE -- PERMANENT SYMBOLS                                                        SEG 0114

            1128    1189    1196    1197    1198    1200    1224    1228    1232    1239    1240    1244    1252    1264    1295
            1296    1299    1301    1304    1305    1313    1316    1318    1319    1328    1333    1335    1341    1343    1344
            1348    1367    1389    1390    1392    1404    1405    1406    1411    1412    1413    1421    1422    1445    1446
            1448    1450    1453    1454    1455    1462    1485    1486    1489    1491    1494    1497    1498    1499    1506
            1513    1514    1519    1544    1545    1548    1557    1565    1566    1567    1574    1582    1583    1603    1604
            1608    1622    1623    1624    1625    1626    1628    1631    1632    1638    1641    1642    1643    1644    1646
            1647    1648    1649    1650    1652    1653    1656    1666    1673    1678    1682    1683    1701    1702    1705
            1711    1719    1721    1723    1724    1725    1726    1727    1735    1737    1752    1753    1759    1768    1798
            1799    1802    1803    1804    1808    1809    1814    1815    1816    1818    1820    1821    1826    1832    1833
            1834    1860    1864    1869    1893    1894    1897    1899    1901    1915    1928    1932    1936    1940    1943
            1963    1964    1966    1968    1972    1973    1974    1976    1977    1979    1980    1982    1983    1984    2007
            2008    2011    2013    2031    2032    2035    2040    2041    2051    2054    2080    2084    2085    2086    2087
            2088    2089    2090    2091    2092    2093    2097    2098    2099    2100    2101    2104    2105    2107    2108
            2110    2112    2116    2118    2122    2123    2134    2135    2136    2137    2138    2142    2143    2145    2146
            2150    2166    2168    2169    2170    2172    2173    2174    2175    2183    2190    2198    2200    2201    2202
            2203    2204    2205    2217    2224    2228    2230    2231    2232    2233    2234    2248    2249    2250    2257
            2264    2268    2272    2273    2275    2276    2283    2287    2290    2305    2306    2325    2326    2327    2334
            2352    2356    2358    2371    2372    2373    2374    2375    2376    2377    2382    2383    2393    2394    2396
            2397    2411    2414    2415    2424    2429    2462    2467    2508    2509    2513    2518    2526    2549    2561
            2578    2582    2691    2703    2714    2732    2733    2734    2739    2742    2743    2744    2745    2746    2756
            2760    2775    2776    2777    2779    2781    2792    2793    2794    2796    2797    2803    2804    2813    2814
            2818    2819    2824    2825    2826    2827    2851    2852    2854    2855    2870    2886    2895    2898    2909
            2912    2913    2940    2941    2942    2945    2958    2959    2964    2969    3019    3021    3025    3052    3062
            3063    3067    3068    3078    3083    3085    3097    3098    3106    3110    3125    3126    3127    3128    3135
            3136    3150    3167    3168    3170    3185    3186    3187    3188    3189    3208    3209    3220    3255    3301
            3305    3321    3323    3325    3332    3337    3339    3362    3376    3380    3382    3386    3388    3407    3408
            3411    3415    3453    3454    3456    3470    3471    3472    3485    3522    3523    3525    3528    3541    3553
            3554    3556    3557    3560    3561    3582    3587    3603    3606    3635    3636    3640    3648    3655    3697
            3702    3711    3716    3721    3722    3724    3727    3731    3738    3739    3770    3778    3779    3780    3786
            3793    3811    3812    3813    3814    3815    3836    3837    3838    3839    3840    3841    3842    3847    3850
            3870    3876    3877    3878    3879    3880    3882    3883    3895    3896    3897    3898    3899    3900    3901
            3902    3903    3904    3909    3913    3914    3915    3916    3917    3918    3919    3920    3939    3940    3944
MOVB         723     730     732     835     899     901     934     936     938     940     962    1116    1129    1167    1173
            1178    1222    1223    1230    1231    1307    1308    1309    1310    1311    1325    1326    1364    1365    1417
            1418    1451    1452    1465    1469    1469    1609    1614    1619    1627    1764    1765    1777    1907    1921
            1923    2096    2336    2395    2399    2404    2427    2452    2456    2476    2489    2519    2569    2616    2634
            2677    2682    2698    2700    2731    2820    2821    2890    2925    2927    2931    2962    2963    2966    2967
            2986    3001    3050    3051    3076    3077    3099    3111    3151    3191    3193    3214    3219    3264    3270
            3273    3312    3355    3366    3374    3375    3378    3384    3422    3429    3559    3589    3637    3665    3673
            3771    3772    3775    3776    3777    3781    3784    3785    3804    3845    3848    3862    3865    3874    3942
NEG         3782    3844
NOP          963    1013    1044    1143    1277    1434    1472    1528    1585    1687    1739    1779    1873    1874    1875
            3334
RESET       1871    2171    2239    2444
ROL         3788    3790    3791    3792    3794
ROLB        3300
RTI          768    2167    2199    2468    2482    2491    2583    2679    2709    2716    3562    3608    3642    3816    3884
            3923
RTS         1260    1265    2159    2240    2286    2298    2310    2359    2384    2416    2430    2446    2750    2765    2782
            2828    2839    2899    2918    2933    2948    2970    2982    2987    2997    3002    3086    3102    3137    3154
            3197    3221    3235    3245    3272    3306    3314    3341    3395    3442    3460    3486    3498    3682    3741
            3945
SEC         3299
SPL         2111
SUB         2189    2213    2270    2271    2564    2960    2965    3066    3080    3322    3338    3363    3371    3373    3588
            3851
```

```
SWAB       900     906     910    1039    1259    1463    2405    3113
SXT       2117
TRAP      3947    3957    3958    3959    3960
TST        701     706     751     753     847     852     858     862     897    1247    1257    1516    1824    2049    2056
          2102    2149    2155    2176    2191    2251    2280    2296    2447    2572    2670    2680    2687    2888    3069
          3071    3229    3483    3524    3548    3595    3604    3639    3647    3669    3733    3799    3856    3866    3941
TSTB       756     813     908    1837    1904    2124    2209    2400    2439    2675    2685    2710    3296    3352    3393
          3535  36[illegible]   3671    3725    3858    3872
WAIT      1906
.ASCII     182     183    3978    4019    4202    4295    4298    4302    4308    4310    4312    4314    4316    4331    4338
          4353    4356    4363    4379    4461    4469
.ASCIZ     181     184    1878    3742    3927    3983    3994    4001    4013    4024    4032    4038    4047    4054    4059
          4066    4072    4080    4086    4095    4103    4109    4114    4122    4128    4133    4140    4151    4160    4168
          4176    4184    4192    4205    4209    4212    4216    4220    4227    4235    4243    4249    4255    4265    4279
          4284    4291    4318    4325    4326    4327    4328    4344    4350    4369    4374    4389    4395    4403    4409
          4417    4423    4431    4438    4446    4451    4470    4474
.BLKB     4519    4521    4523
.BLKW      410     411     412    3889
.BYTE      153     154     159     160     175     176     177     178     341     343     345     351     359     360     362
           790     791     795     796     800     801    1275    1337    1371    1395    1398    1401    1501    1504    1524
          1526    1550    1554    1569    1572    1707    1709    1761    1881    1949    2058    2220    2221    2293    2294
          3276    3277    3328    3330    3817    3818    3819    3820    4477    4484    4491    4497    4503    4505    4507
          4509    4514
.ENABL       3
.END      4525
.ENDC        8      21      23      24      25      30      38     116     130     140     144     151     153     179     180
           181     182     186     370     372     390     392     408     410     426     428     438     440     450     452
           478     480     511     513     526     528     568     570     572     574     582     584     636     639     665
           666     669     679     686     739     745     791     792     796     797     801     802     818     819     826
           829     830     831     832     921     927     929     930     931     932     965     972     975     976     977
           978    1015    1019    1021    1022    1023    1024    1048    1055    1057    1058    1059    1060    1096    1102
          1104    1105    1106    1107    1146    1155    1157    1158    1159    1160    1280    1292    1294    1295    1296
          1297    1378    1386    1388    1389    1390    1391    1437    1442    1444    1445    1446    1447    1474    1482
          1484    1485    1486    1487    1531    1541    1543    1544    1545    1546    1589    1600    1602    1603    1604
          1605    1690    1698    1700    1701    1702    1703    1743    1749    1751    1752    1753    1754    1784    1795
          1797    1798    1799    1800    1842    1845    1847    1848    1850    1853    1859    1862    1863    1867    1868
          1877    1878    1881    1882    1883    1890    1892    1893    1894    1895    1953    1960    1962    1963    1964
          1965    1997    2004    2006    2007    2008    2009    2019    2028    2030    2031    2032    2033    2076    2078
          2084    2085    2087    2089    2091    2093    2094    2095    2097    2099    2109    2129    2133    2161    2164
          2194    2197    2221    2222    2243    2247    2294    2295    2300    2304    2313    2323    2363    2369    2387
          2391    2419    2422    2433    2438    2472    2476    2494    2506    2657    2670    2725    2730    2753    2755
          2768    2773    2786    2790    2809    2812    2831    2834    2841    2849    2905    2907    2921    2924    2936
          2938    2952    2956    2973    2975    2978    2990    2993    3006    3017    3091    3096    3117    3123    3143
          3149    3162    3165    3177    3182    3200    3207    3224    3227    3238    3240    3249    3253    3281    3288
          3308    3311    3316    3318    3344    3348    3398    3405    3445    3449    3464    3468    3490    3492    3501
          3507    3512    3517    3519    3530    3533    3534    3535    3537    3539    3546    3550    3555    3556    3560
          3563    3564    3565    3571    3580    3587    3592    3593    3594    3595    3601    3608    3609    3610    3637
          3688    3706    3744    3745    3823    3891    3903    3913    3923    3930    3931    3940    3943    3956    3957
          3958    3959    3960    3961
.EGUIV      30      31      33      48      49      78      79      80      81      82      83      84      85      86      87
           106     107     108     109     110     111     112     113     114     115
.EVEN      364    1266    3743    3929    4093    4201    4518
.IF          4      21      22      23      24      25      28      36      88     116     140     143     150     152     179
           180     181     185     186     369     371     389     391     407     409     425     427     437     439     449
           451     477     479     510     512     525     527     567     569     571     573     581     583     635     638
           665     668     678     685     738     744     790     791     795     796     800     801     817     818     825
```

```
           828     830     831     832     920     926     928     930     931     932     964     971     974     976     977
           978    1014    1018    1020    1022    1023    1024    1047    1054    1056    1058    1059    1060    1095    1101
```

```
            1103    1105    1106    1107    1145    1154    1156    1158    1159    1160    1279    1291    1293    1295    1296
            1297    1377    1385    1387    1389    1390    1391    1436    1441    1443    1445    1446    1447    1473    1481
            1483    1485    1486    1487    1530    1540    1542    1544    1545    1546    1588    1599    1601    1603    1604
            1605    1689    1697    1699    1701    1702    1703    1742    1748    1750    1752    1753    1754    1783    1794
            1796    1798    1799    1800    1841    1845    1846    1847    1848    1849    1850    1852    1858    1861    1863
            1867    1868    1877    1878    1882    1889    1891    1893    1894    1895    1952    1359    1961    1963    1964
            1965    1996    2003    2005    2007    2008    2009    2018    2027    2029    2031    2032    2033    2075    2077
            2080    2084    2085    2087    2089    2091    2093    2094    2095    2097    2109    2128    2132    2160    2163
            2193    2196    2220    2221    2242    2246    2293    2294    2299    2303    2312    2322    2362    2368    2386
            2390    2418    2421    2432    2437    2470    2474    2493    2505    2656    2669    2724    2729    2752    2754
            2767    2772    2785    2789    2808    2811    2830    2833    2840    2848    2904    2906    2920    2923    2935
            2937    2951    2955    2972    2974    2977    2989    2992    3005    3016    3090    3095    3116    3122    3142
            3148    3161    3164    3176    3181    3199    3206    3223    3226    3237    3239    3248    3252    3280    3287
            3307    3310    3315    3317    3343    3347    3397    3404    3444    3448    3463    3467    3489    3491    3500
            3506    3511    3516    3517    3529    3531    3532    3533    3535    3536    3537    3546    3548    3556    3557
            3562    3563    3564    3570    3580    3583    3590    3592    3593    3595    3598    3601    3608    3609    3637
            3687    3705    3721    3744    3822    3890    3903    3913    3921    3923    3927    3930    3939    3943    3947
            3957    3958    3959    3960    3961
.IFF          21      23      24      25      28      38     144     150     152     179     186     369     371     389     391
             407     409     425     427     437     439     449     451     477     479     510     512     525     527     567
             569     571     573     581     583     635     638     665     668     678     685     738     744     791     792
             796     600     801     818     825     829     830     831     920     926     929     930     964     971     975
             976    1014    1018    1021    1022    1047    1054    1057    1058    1095    1101    1104    1105    1106    1145
            1154    1157    1158    1159    1279    1291    1294    1295    1377    1385    1388    1389    1436    1441    1444
            1445    1473    1481    1484    1485    1530    1540    1543    1544    1588    1599    1602    1603    1604    1689
            1697    1700    1701    1742    1748    1751    1752    1783    1794    1797    1798    1799    1842    1849    1853
            1858    1861    1878    1882    1889    1892    1893    1894    1952    1959    1962    1963    1964    1996    2003
            2006    2007    2008    2018    2027    2030    2031    2032    2075    2077    2084    2128    2132    2160    2163
            2193    2196    2221    2222    2242    2246    2294    2295    2299    2303    2312    2322    2362    2368    2386
            2390    2418    2421    2432    2437    2470    2474    2493    2505    2656    2669    2724    2729    2752    2754
            2767    2772    2785    2789    2808    2811    2830    2833    2840    2848    2904    2906    2920    2923    2935
            2937    2951    2955    2972    2974    2977    2989    2992    3005    3016    3090    3095    3116    3122    3142
            3148    3161    3164    3176    3181    3199    3206    3223    3226    3237    3239    3248    3252    3280    3287
            3308    3311    3315    3317    3343    3347    3397    3404    3444    3448    3463    3467    3489    3491    3501
            3530    3533    3534    3537    3563    3565    3570    3583    3608    3609    3610    3688    3705    3721    3745
            3823    3891    3923    3931    3940
.IFT        3545    3593
.IFTF       3543    3592
.IIF           3       8      13      18      19      20      21      24      25     137     185     788     793     798    1847
            1853    1854    1865    1878    1882    2085    2087    2093    2094    2095    2097    2098    2218    2291    3507
            3508    3509    3510    3511    3512    3544    3545    3560    3563    3564    3571    3572    3573    3574    3575
            3598    3609    3687    3703    3728    3732    3956    3957    3958    3959    3960
.IRP         665     779     783     828     839     851     854     866     870     917     928     974     993    1011    1020
            1042    1056    1080    1083    1084    1091    1103    1110    1111    1112    1156    1244    1293    1318    1319
            1328    1343    1344    1367    1387    1422    1443    1483    1542    1583    1601    1631    1632    1652    1653
            1682    1699    1721    1737    1750    1796    1802    1803    1804    1820    1821    1832    1833    1834    1891
            1961    1972    1973    1974    2005    2029    2257    2264    2268    2272    2273    2275    2276    2325    2326
            2327    2352    2356    2358    2371    2372    2373    2374    2382    2383    2393    2415    2424    2429    2508
            2582    2732    2733    2734    2742    2743    2744    2745    2746    2775    2776    2779    2781    2792    2796
            2827    2851    2895    2898    2912    2913    2958    2959    2969    3019    3062    3063    3083    3085    3125
            3135    3208    3220    3255    3470    3471    3472    3485    3516    3836    3876    3897    3913
.LIST          2       3      24     130     137     179     369     370     371     372     389     390     391     392     407
             408     409     410     425     426     427     428     437     438     439     440     449     450     451     452
             477     478     479     480     510     511     512     513     525     526     527     528     567     568     569
```

```
             570     571     572     573     574     581     582     583     584     635     636     638     639     665     666
             668     669     678     679     685     686     738     739     744     745     818     819     825     826     828
             830     920     921     926     927     928     930     964     965     971     972     974     976    1014    1015
            1018    1019    1020    1022    1047    1048    1054    1055    1056    1058    1095    1096    1101    1102    1103
            1105    1145    1146    1154    1155    1156    1158    1279    1280    1291    1292    1293    1295    1377    1378
            1385    1386    1387    1389    1436    1437    1441    1442    1443    1445    1473    1474    1481    1482    1483
            1485    1530    1531    1540    1541    1542    1544    1588    1589    1599    1600    1601    1603    1689    1690
```

```
                                                      H09
           1697   1698   1699   1701   1742   1743   1748   1749   1750   1752   1783   1784   1794   1795   1796
           1798   1853   1882   1883   1889   1890   1891   1893   1952   1953   1959   1960   1961   1963   1996
           1997   2003   2004   2005   2007   2018   2019   2027   2028   2029   2031   2075   2076   2077   2078
           2128   2129   2132   2133   2160   2161   2163   2164   2193   2194   2196   2197   2242   2243   2246
           2247   2299   2300   2303   2304   2312   2313   2322   2323   2362   2363   2368   2369   2386   2387
           2390   2391   2418   2419   2421   2422   2432   2433   2437   2438   2470   2471   2472   2474   2475
           2476   2493   2494   2505   2506   2656   2657   2669   2670   2724   2725   2729   2730   2752   2753
           2754   2755   2767   2768   2772   2773   2785   2786   2789   2790   2808   2809   2811   2812   2830
           2831   2833   2834   2840   2841   2848   2849   2904   2905   2906   2907   2920   2921   2923   2924
           2935   2936   2937   2938   2951   2952   2955   2956   2972   2973   2974   2975   2977   2978   2989
           2990   2992   2993   3005   3006   3016   3017   3090   3091   3095   3096   3116   3117   3122   3123
           3142   3143   3148   3149   3161   3162   3164   3165   3176   3177   3181   3182   3199   3200   3206
           3207   3223   3224   3226   3227   3237   3238   3239   3240   3248   3249   3252   3253   3280   3281
           3287   3288   3315   3316   3317   3318   3343   3344   3347   3348   3397   3398   3404   3405   3444
           3445   3448   3449   3463   3464   3467   3468   3489   3490   3491   3492   3511   3598   3947   3956
           3957   3958   3959   3960   3961
.MACRO       25    143   3947
.MCALL        3    130
.NLIST        1      3     24    130    137    179    369    370    371    372    389    390    391    392    407
            408    409    410    425    426    427    428    437    438    439    440    449    450    451    452
            477    478    479    480    510    511    512    513    525    526    527    528    567    568    569
            570    571    572    573    574    581    582    583    584    635    636    638    639    665    666
            668    669    678    679    685    686    738    739    744    745    818    819    825    826    828
            830    920    921    926    927    928    930    964    965    971    972    974    976   1014   1015
           1018   1019   1020   1022   1047   1048   1054   1055   1056   1058   1095   1096   1101   1102   1103
           1105   1145   1146   1154   1155   1156   1158   1279   1280   1291   1292   1293   1295   1377   1378
           1385   1386   1387   1389   1436   1437   1441   1442   1443   1445   1473   1474   1481   1482   1483
           1485   1530   1531   1540   1541   1542   1544   1588   1589   1599   1600   1601   1603   1689   1690
           1697   1698   1699   1701   1742   1743   1748   1749   1750   1752   1783   1784   1794   1795   1796
           1798   1853   1882   1883   1889   1890   1891   1893   1952   1953   1959   1960   1961   1963   1996
           1997   2003   2004   2005   2007   2018   2019   2027   2028   2029   2031   2075   2076   2077   2078
           2128   2129   2132   2133   2160   2161   2163   2164   2193   2194   2196   2197   2242   2243   2246
           2247   2299   2300   2303   2304   2312   2313   2322   2323   2362   2363   2368   2369   2386   2387
           2390   2391   2418   2419   2421   2422   2432   2433   2437   2438   2470   2471   2472   2474   2475
           2476   2493   2494   2505   2506   2656   2657   2669   2670   2724   2725   2729   2730   2752   2753
           2754   2755   2767   2768   2772   2773   2785   2786   2789   2790   2808   2809   2811   2812   2830
           2831   2833   2834   2840   2841   2848   2849   2904   2905   2906   2907   2920   2921   2923   2924
           2935   2936   2937   2938   2951   2952   2955   2956   2972   2973   2974   2975   2977   2978   2989
           2990   2992   2993   3005   3006   3016   3017   3090   3091   3095   3096   3116   3117   3122   3123
           3142   3143   3148   3149   3161   3162   3164   3165   3176   3177   3181   3182   3199   3200   3206
           3207   3223   3224   3226   3227   3237   3238   3239   3240   3248   3249   3252   3253   3280   3281
           3287   3288   3315   3316   3317   3318   3343   3344   3347   3348   3397   3398   3404   3405   3444
           3445   3448   3449   3463   3464   3467   3468   3489   3490   3491   3492   3511   3598   3947   3956
           3957   3958   3959   3960   3961
.PAGE       143    185
.REPT       137    369    371    389    391    407    409    425    427    437    439    449    451    477    479
            510    512    525    527    567    569    571    573    581    583    635    638    665    668    678
            685    738    744    818    825    920    926    964    971   1014   1018   1047   1054   1095   1101
           1145   1154   1279   1291   1377   1385   1436   1441   1473   1481   1530   1540   1588   1599   1689
MAINDEC-11-DZVTH-A      MACY11 27(732)  20-SEP-76  10:22  PAGE 110
DZVTH.P11         CROSS REFERENCE TABLE -- PERMANENT SYMBOLS                                         SEQ 0118

           1697   1742   1748   1783   1794   1882   1889   1952   1959   1996   2003   2018   2027   2075   2077
           2128   2132   2160   2163   2193   2196   2242   2246   2299   2303   2312   2322   2362   2368   2386
           2390   2418   2421   2432   2437   2470   2474   2493   2505   2642   2656   2669   2724   2729   2752
           2754   2767   2772   2785   2789   2808   2811   2830   2833   2840   2848   2904   2906   2920   2923
           2935   2937   2951   2955   2972   2974   2977   2989   2992   3005   3016   3090   3095   3116   3122
           3142   3148   3161   3164   3176   3181   3199   3206   3223   3226   3237   3239   3248   3252   3280
           3287   3315   3317   3343   3347   3397   3404   3444   3448   3463   3467   3489   3491
.SBTTL       14     26    131    145    187   1843   3502   3566   3611   3689   3746   3824   3892   3932   3948
.TITLE        3
.WORD       137    138    139    152    155    156    157    158    161    162    163    164    165    166    167
            168    169    170    338    340    347    349    353    355    357    416    417    419    420    421
            422    423    424    428    429    430    431    432    433    434    442    444    446    448    450
```

```
          454     457     460     463     466     469     473     476     478     483     485     488     490     493     496
          499     503     505     508     511     515     517     520     522     526     530     532     533     535     538
          539     542     544     545     547     549     550     552     554     555     557     559     560     562     564
          565     568     587     589     597     602     604     606     607     608     609     610     612     614     616
          617     618     619     620     643     644     645     646     647     648     649     650     651     652     653
          654     656     657     658     659     660     661     662    1268    1269    1270    1271    1272    1273    1430
         1431    1432    1433    1523    1858    1861    2064    2066    2070    2072    2637    2638    2639    2640    2651
         2652    2653    2654    2719    2720    2721    2723    2783    2900    2901    2902    3004    3462    3681    3714
         3719    3821    3922    4522


ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DZVTH/SOL/CRF=DZVTH
RUN-TIME: 47 41 10 SECONDS
RUN-TIME RATIO: 257/100=2.5
CORE USED:  20K  (39 PAGES)
```

J09

```
Spooler runtime 19 Seconds, 81 KCS, 556 disk reads, 3 disk writes, 119 pages
[illegible] Date 08-Nov-76 1[illegible] Monitor [illegible]
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000111111111111111111111111111111110
000000000111111111122222222223333333333444444444455555555556666666666777777777788888888889999999999000000000011111111112222222222233312
[illegible]4567890123456789012 **
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000111111111111111111111111111111110
000000000111111111122222222223333333333444444444455555555556666666666777777777788888888889999999999000000000011111111112222222222233312
```