VT71

CONTROL/VIDIO TEST

MD-11-DZKVB-B

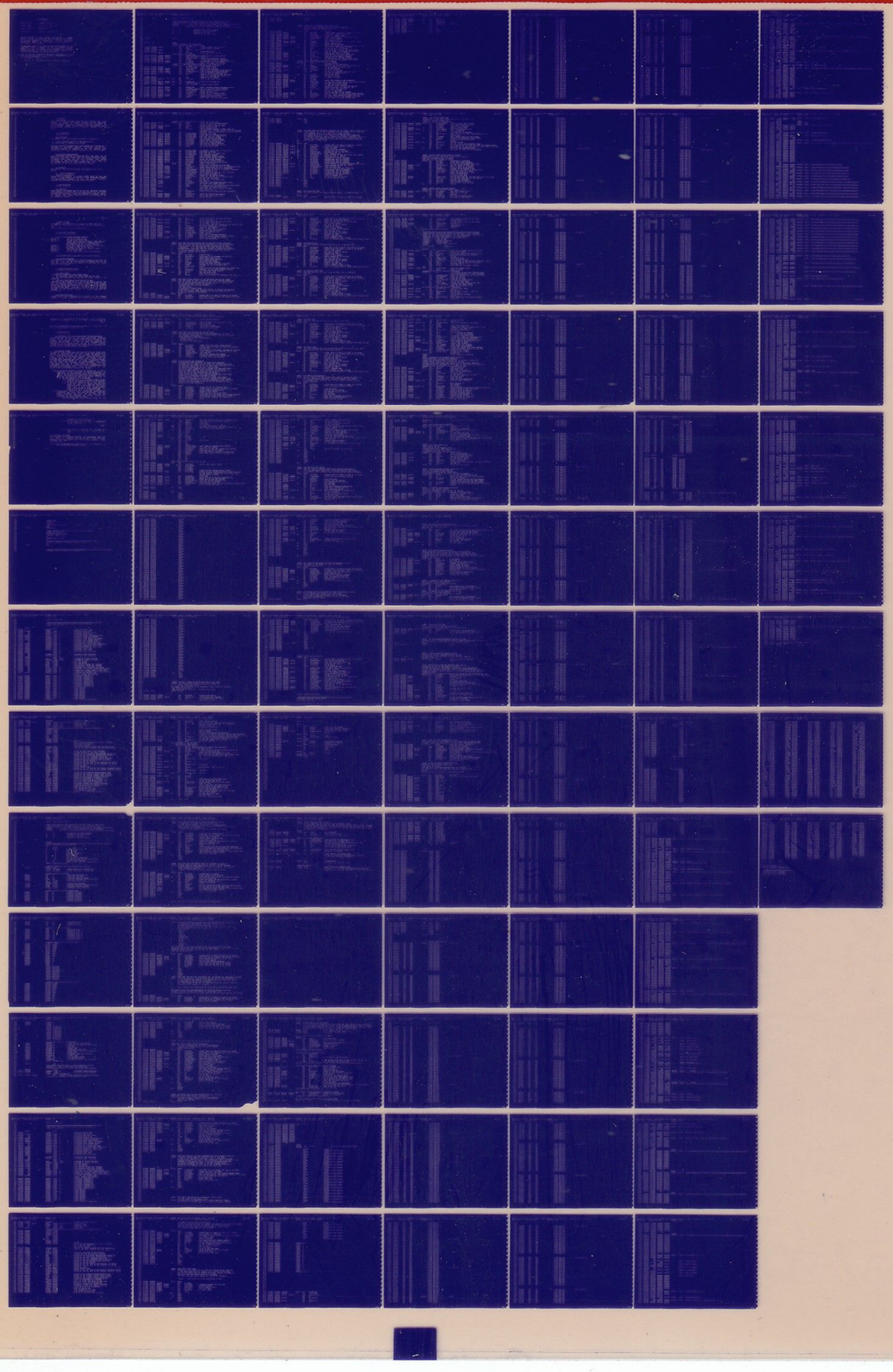
EP-DZKVB-B-DL

FICHE 1 OF 1

JUN 1978

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE: MAINDEC-11-D2KVB-B-D

PRODUCT NAME: VT71 CONTROL/VIDIO TEST

PRODUCT DATE: JANUARY 1977

MAINTAINER: DIAGNOSTIC ENGINEERING

```
                        1. ABSTRACT
                        -----------
DZKVB IS A PROGRAM WRITTEN TO TEST THE VT71  CONTROL  AND  VIDEO
BOARDS.  IT CHECKS FOR PROPER OPERATION  OF  ALL STATUS AND
CONTROL REGISTER BITS, AND INTERRUPTS UNIQUE TO THE CONTROL  AND
VIDEO  BOARDS.  ERRORS ARE REPORTED ON THE VT71 SCREEN, AND ON
ANY ADDITIONAL TERMINAL THAT IS  INTERFACED  DIRECTLY  WITH  THE
VT71.


                        2. REQUIREMENTS
                        ---------------

                        2.1 HARDWARE
FOR DZKVB TO RUN, THE FOLLOWING EQUIPMENT IS NESCECARY,

  A.  A VT71 TERMINAL WITH 8K OF READ/WRITE MEMORY
  B.  SOME MEANS OF LOADING THIS PROGRAM

THE ONLY OPTIONAL EQUIPMENT THAT THIS PROGRAM WILL UTILIZE IS  A
TELLETYPE  OR  EQUIVILENT  TERMINAL,  INTERFACED  WITH THE VT71.
THIS CAN BE USEFULL  IF  PROBLEMS  IN  THE  VT71  PREVENT  ERROR
INFORMATION FROM BEING DISPLAYED PROPERLY ON THE VT71'S SCREEN.


                        2.2 SOFTWARE REQUIREMENTS
IF AN ADDITIONAL TERMINAL IS INTERFACED WITH  THE  VT71,  IT  IS
BEST TO RUN THE LSI-11 MEMORY TEST, THE LSI-11 INSTRUCTION TEST,
THE LSI-11 TRAPS  TEST,  AND  THE  VT71  KEYBOARD  TEST,  BEFORE
ATTEMPTING  TO  RUN THIS PROGRAM.  THIS WILL HELP TO INSURE THAT
ANY  ERRORS  REPORTED  BY  THIS  PROGRAM  ARE  TRULY  DUE  TO
MALFUNCTIONS OF THE VT71 CONTRLO AND VIDEO BOARDS.


                        2.3 STORAGE
THIS PROGRAM USES LOCATIONS 000000 THRU 024000 OF THE VT71'S
MEMORY.

                        3.  LOADING PROCECDURE
                        ----------------------
THIS PROGRAM IS SUPPLIED  ON  PUNCHED  PAPER  TAPE  IN  THE  ABS
FORMAT.  IN MOST CASES IT IS EASIEST TO LOAD THE PROGRAM USING A
VT20 HOST PROGRAM B COMMAND.  THE VT20  HOST  PROGRAM  DOCUMENTS
DESCRIBE THIS PROCEDURE IN DETAIL(DZVTGA, DZVTEA).


                        4. USER PROCEDURE
                        -----------------

                        4.1 STARTING
AFTER LOADING THE PROGRAM USE ODT TO SET THE  PROGRAMS  SOFTWARE
SWITCH REGISTER TO THE DESIRED VALLUE(SEE SECTION 5.1 FOR SWITCH
REGISTER BIT FUNCTIONS).  START THE PROGRAM AT LOCATION  000200.
THE  PROGRAM  WILL  DISPLAY INSTRUCTIONS FOR THE OPERATOR ON THE
VT71 SCREEN AS NEEDED.
```

```
                        4.2 END OF PASS
                THE PROGRAM, UPON COMPLETION OF A PASS WILL CAUSE THE VT71
                'CLICKER' TO MAKE A TONE. THE PASS COUNT IS THEN DISPLAYED AND
                THE
                PROGRAM LOOPS BACK TO BEGIN ANOTHER PASS.


                        5. OPERATING PROCEDURE
                        ----------------------

                        5.1 SOFTWARE SWITCH REGISTER SETTINGS
                SW<15> = 1          HALT IF AN ERROR OCCURS
                SW<14> = 1          LOOP ON THE CURRENT TEST
                SW<13> = 1          INHIBIT ERROR MESSAGE DISPLAY.  THIS DOES NOT
                                    AFFECT THE DISPLAYING OF INSTRUCTIONS FOR
                                    THE OPERATOR, OR END OF PASS MESSAGES.
                SW<12> = 1          STALL FOR 2 SECONDS AT THE END OF EACH TEST.
                SW<11< = 1          DO TESTS 27 THRU 34 AND TEST 41
                SW<10> = 1          SOUND BELL ON ERROR-VT71 BEEPS ON ERROR
                SW<09> = 1          LOOP ON ERROR TEST
                SW<08> = 1          LOOP ON TEST WHOSE NUMBER IS IN SW<7:0>


                        THE SOFTWARE SWITCH REGISTER IS AT LOCATION 000176.

                        5.3 RESTART PROCEDURE
                THIS PROGRAM MAY BE RESTARTED AT LOCATION 000200 AT ANY TIME  IF
                NEED  BE.   THIS  CAN  BE  DONE MANUALY, BY STOPING THE VT71 AND
                USING ODT, OR IF AN ADDITIONAL TERMINAL IS AVAILABLE, BY  TYPING
                CRTL-B ON ITS KEYBOARD.


                        6. PROGRAM/OPERATOR ACTION
                        --------------------------

                        6.1 ERROR HALTS
                THE FOLLOWING THINGS CAN CAUSE ERROR HALTS,
                  A.  ANY INTERRUPT, EXCEPT FOR THOSE CAUSED BY THE VT71
                  B.  ANY TRAP CONDITION THAT DOES NOT USE EITHER  VECTOR  4  OR
                VECTOR 10
                  C.  ANY ERROR THAT OCCURS WHILE SWR<15> IS SET.
                CONDITIONS A AND B ABOVE CAUSE A HALT AT THE SECOND WORD OF  THE
                TRAP  OR INTERRUPT VECTOR.  TO FIND OUT WHERE THE ERROR OCCURED,
                USE ODT TO EXAMINE THE CONTENTS OF R6, THEN EXAMINE THE CONTENTS
                OF THE LOCATION POINTED TO BY R6.  THIS VALUE IS THE PC VALUE AT
                THE TIME OF THE ERROR.  CONDITION C CAUSES A  HALT  AT  LOCATION
                "HALTER"  IN  THE  ERRMES  ROUTINE.   NO  OTHER  ERROR HALTS ARE
                PROVIDED FOR.


                        6.2 ERROR PRINTOUTS
                UNLESS SOFTWARE SWITCH REGISTER BIT 13 IS SET,  IF  THE  PROGRAM
                DETECTS  AN  ERROR,  UNLESSS IT IS ON TEST 1 OR 2 OR 3, AN ERROR
```

C1

MESSAGE WILL BE DISPLAYED ON THE VT71 SCREEN. IF AN ADDITIONAL
TERMINAL IS INTERFACED WITH THE VT71, THEN THE ERROR MESSAGE IS
PAGE 3

SENT TO IT REGARDLESS OF WHAT TEST IS RUNNING. (UNLESS BIT 13
OF THE SOFTWARE SWITCH REGISTER IS SET.

7. MISCELLANEOUS

7.1 TESTING A VT71 WITH NON-STANDARD VECTORS/ADDRESSES

THIS PROGRAM CAN TEST THE VT71 THAT USES NON-STANDARD ADDRESSES
AND/OR VECTORS IF THE CORRECT INFORMATION IS SUPLIED TO THE
PROGRAM. THE STANDARD ADDRESSES ARE CONTAINED STARTING AT
LOCATION 000774('TPS'). IF ANY OF THESE ADDRESSES ARE NOT
CORRECT FOR THE VT71 TO BE TESTED, USE ODT TO MODIFY LOCATION
"LEAVEC" TO ANY NON ZERO VALUE, THEN CHANGE THOSE ADDRESSES THAT
ARE NOT CORRECT, BEFORE STARTING THE PROGRAM.

7.2 SPECIAL TESTS

TWO TEST ROUTINES, T0036 T0037 ARE INCLUDED TO ALLOW TIGHT SCOPE
LOOPING OF VERY BASIC VT71 FUNCTIONS. THESE TEST ROUTINES ARE
NOT RUN AUTOMATICLY AS PART OF THE STANDARD DIAGNOSTIC TEST
PASS. THEY CAN ONLY BE ENTERED MANUALY. THERE ARE TWO WAYS TO
DO THIS. ONE IS TO PUT THE DESIRED TEST NUMBER INTO THE
SOFTWARE SWITCH REGISTER AND START THE LSI11 AT LOCATION 000200.
OR, EACH TEST CAN BE STARTED AT ITS FIRST INSTRUCTION, SINCE
BOTH OF THESE TEST ROUTINES IS SELF SUFFICIENT. UNLIKE ALL
OTHER TESTS, ONCE ONE OF THESE TESTS IS ENTERED, IT WILL
AUTOMATICLY LOOP UNTIL IT IS MANUALY STOPPED, REGARDLESS OF THE
SOFTWARE SWITCH REGISTER BITS. ALSO, NO ATTEMPTS ARE MADE
WITHIN THESE TEST ROUTINES TO REPORT ANY ERRORS.

T0036 ALLOWS SCOPE LOOPING ON EITHER THE WRITING OF A SINGLE
CHARACTER, OR THE DISPLAYING OF A SINGLE CHARACTER. IS
THE ROUTINE IS SIMPLY STARTED AFTER THE PROGRAM HAS BEEN
LOADED, THE ROUTINE WILL CONTINOUSLY WRITE CHARACTER 101
THERE ARE TWO LOCATIONS THAT CAN BE MANUALY MODIFYED, TO
CAUSE THE TEST ROUTINE TO DO OTHERWISE.

XCODE WHATEVER VALUE IS IN LOCATION "XCODE" IS THE
VALUE OF THE CHARACTER THAT WILL BE WRITTEN.

XDISP IF LOCATION "XDISP" IS SET TO ANY NON-ZERO
VALUE THE TEST WILL SETUP TO DISPLAY THE
XHARACTER AFTER IS IS WRITEN FOR THE FIRST
TIME, THEN IT WILL TURN ON THE DISPLAY.
WHILE THE CHARACTER IS BEING DISPLAYED, THE
ROUTINE WILL BE IN A "DO NOTHING" LOOP.

T0037 ALLOWS SCOPE LOOPING WHILE DISPLAYINGA PREDETERMINED
CHARACTERS PER LINE, AND LINES PER SCREEN. LINES PER
SCREEN, CHARACTERS PER LINE, AND THE CODE FOR THE
CHARACTER TO BE DISPLAYED ARE MANUALY INPUTED INTO
LOCATIONS THAT T0037 READS AND USES.

YCODE SET LOACATION "YCODE" TO THE VALUE OF THE

D1

```
                                                    CHARACTER YOU WANT DISPLAYED.
                                                YBLOCK SET LOCATION "YBLOCK TO THE   OF  CHARACTERS
                                                       TO  BE  DISPLAYED  PER  LINE.   DEFAULT IS 1
                                                       CHARACTER PER LINE
                                                                                          PAGE 4


                                                YLINES SET LOCATION "YLINES" TO THE   OF LINES  YOU
                                                       WISH  TO  HAVE  DISPLAYED.  DEFAULT IS FOR 1
                                                       LINE

                                            7.3 ADITIONAL TELLETYPE
                                            IF A TELLETYPE OR EQUIVILENT TERMINAL  IS  INTERFACED  WITH  THE
                                            VT71.  THERE  IS 1 CHARACTER THAT CAN BE TYPED ON THAT TERMINALS
                                            KEYBOARD, THAT WILL AFFECT THE RUNNING OF THIS  PROGRAM.   THESE
                                            CHARACTERS ARE...

                                            CTRL-B WHEN THE PROGRAM SEES CTRL-B IT WILL.  ON  COMPLETION  OF
                                                   THE CURRENT TEST, RESTART ITSELF.

                                    .ENDR
```

```
                                .ENABLE AMA
                                .NLIST CND,MC,MD
                                .LIST ME
                                ;VT71 CONTROL/VIDEO TEST        V E R S I O N     T H R E E
                                .ABS
147400                          $SWR=147400

                                .TITLE  MAINDEC-11-DZKVB-B  VT71 CONTROL/VIDEO PROGRAM
                                ;*COPYRIGHT (C) 1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY J. COMEAU
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-D7QAC-C2), SEPT 14, 1976.
                                ;*
000001                          $TN=1


                                ;*********************************************************
                                ;SOFTWARE SWITCH REGISTER IS AT LOCATION 176
```

```
                        .SBTTL  COMMON TAGS

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

        000600                  .=600
000600                  $CMTAG:                 ;;START OF COMMON TAGS
000600  000000          $PASS:  .WORD   0       ;;CONTAINS PASS COUNT
000602     000          $TSTNM: .BYTE   0       ;;CONTAINS THE TEST NUMBER
000603     000          $ERFLG: .BYTE   0       ;;CONTAINS ERROR FLAG
000604  000000          $ICNT:  .WORD   0       ;;CONTAINS SUBTEST ITERATION COUNT
000606  000000          $LPADR: .WORD   0       ;;CONTAINS SCOPE LOOP ADDRESS
000610  000000          $LPERR: .WORD   0       ;;CONTAINS SCOPE RETURN FOR ERRORS
000612  000000          $ERTTL: .WORD   0       ;;CONTAINS TOTAL ERRORS DETECTED
000614     000          $ITEMB: .BYTE   0       ;;CONTAINS ITEM CONTROL BYTE
000615     001          $ERMAX: .BYTE   1       ;;CONTAINS MAX. ERRORS PER TEST
000616  000000          $ERRPC: .WORD   0       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
000620  000000          $GDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'GOOD' DATA
000622  000000          $BDADR: .WORD   0       ;;CONTAINS ADDRESS OF 'BAD' DATA
000624  000000          $GDDAT: .WORD   0       ;;CONTAINS 'GOOD' DATA
000626  000000          $BDDAT: .WORD   0       ;;CONTAINS 'BAD' DATA
000630  000000                  .WORD   0       ;;RESERVED--NOT TO BE USED
000632  000000                  .WORD   0
000634     000          $AUTOB: .BYTE   0       ;;AUTOMATIC MODE INDICATOR
000635     000          $INTAG: .BYTE   0       ;;INTERPUPT MODE INDICATOR
000636  000000                  .WORD   0
000640  177570          SWR:    .WORD   DSWR    ;;ADDRESS OF SWITCH REGISTER
000642  177570          DISPLAY: .WORD  DDISP   ;;ADDRESS OF DISPLAY REGISTER
000644  177560          $TKS:   177560          ;;TTY KBD STATUS
000646  177562          $TKB:   177562          ;;TTY KBD BUFFER
000650  177564          $TPS:   177564          ;;TTY PRINTER STATUS REG. ADDRESS
000652  177566          $TPB:   177566          ;;TTY PRINTER BUFFER REG. ADDRESS
000654     000          $NULL:  .BYTE   0       ;;CONTAINS NULL CHARACTER FOR FILLS
000655     002          $FILLS: .BYTE   2       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
000656     012          $FILLC: .BYTE   12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
000657     000          $TPFLG: .BYTE   0       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
000660  000000          $REGAD: .WORD   0       ;;CONTAINS THE ADDRESS FROM
                                                ;;WHICH  ($REG0) WAS OBTAINED
000662  000000          $REG0:  .WORD   0       ;;CONTAINS (($REGAD)+0)
000664  000000          $REG1:  .WORD   0       ;;CONTAINS (($REGAD)+2)
000666  000000          $REG2:  .WORD   0       ;;CONTAINS (($REGAD)+4)
000670  000000          $REG3:  .WORD   0       ;;CONTAINS (($REGAD)+6)
000672  000000          $REG4:  .WORD   0       ;;CONTAINS (($REGAD)+10)
000674  000000          $REG5:  .WORD   0       ;;CONTAINS (($REGAD)+12)
000676  000000          $REG6:  .WORD   0       ;;CONTAINS (($REGAD)+14)
000700  000000          $REG7:  .WORD   0       ;;CONTAINS (($REGAD)+16)
000702  000000          $TMP0:  .WORD   0       ;;USER DEFINED
000704  000000          $TMP1:  .WORD   0       ;;USER DEFINED
000706  000000          $TMP2:  .WORD   0       ;;USER DEFINED
000710  000000          $TMP3:  .WORD   0       ;;USER DEFINED
000712  000000          $TMP4:  .WORD   0       ;;USER DEFINED
000714  000000          $TMP5:  .WORD   0       ;;USER DEFINED
000716  000000          $TMP6:  .WORD   0       ;;USER DEFINED
000720  000000          $TMP7:  .WORD   0       ;;USER DEFINED
000722  000000          $TIMES: 0               ;;MAX. NUMBER OF ITERATIONS
```

```
000724  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
000726  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
000732     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
000733     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
000734  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;**************************************************************
000736  000000                  CNXFER: 000000
000740  000000                  CNCHAR: 000000
000742  000000                  CNRECV: 000000
000744  000000                  CNERRO: 000000
000746  000000                  BM0:    000000
000750  000000                  BM1:    000000
000752  000000                  BM2:    000000
000754  000000                  SPMODE: 000000
000756  000000                  TUBSWT: 000000                  ;SWITCH
000760  000000                  LEAVEC: 000000                  ;SET IF DEVICE ADDRESSES AND VECTOR ADDRESSES
                                                                ;ARE TO BE LEFT ALONE BY
                                                                ;THE FINDTT ROUTINE
000762  000001                  UPFAST: 000001                  ;THIS IS THE SPEED CONSTANT FOR FAST PANNING UP
000764  000001                  DOWNFA: 000001                  ;THIS IS THE SPEED CONSTANT FOR FAST PANNING DOWN
000766  000000                  TEMP:   000000
000770  000000                  LINCNT: 000000                  ;COUNT OF # OF LINES TO BE DISPLAYED
000772  000000                  STLCNT: 000000                  ;COUNTER DEVOTED TO THE STALL ROUTINE
000774  177564                  TPS:    177564                  ;POINTS TO ANY ADDITIONAL TELLEPRINTER STATUS WORD
000776  177566                  TPB:    177566                  ;POINTS TO ANY ADDITIONAL TELLEPRINTER BUFFERS
001000  177570                  KBSR:   177570                  ;POINTS TO THE VT71 KEYBOARD STATUS REG
001002  177572                  KBUF:   177572                  ;POINTS TO THE VT71 KEYBOARD BUFFER REGISTER
001004  177574                  LCSR:   177574                  ;POINTS TO THE LED STATUS/CONTROL REGISTER
001006  177576                  LBUF:   177576                  ;POINTS TO THE LED BUFFER
001010  000070                  KBVAD1: 000070                  ;POINTS TO THE 1ST WORD OF THE KEYBOARD INT VECTOR
001012  000072                  KBVAD2: 000072                  ;POINTS TO THE 2ND WORD OF THE KEYBOARD INT VECTOR
001014  000074                  LDVAD1: 000074                  ;VECTOR PC WORD
001016  000076                  LDVAD2: 000076                  ;VECTOR STATUS WORD
001020  000360                  DSVAD1: 000360                  ;POINTS TO THE 1ST WORD OF THE DISPLAY INTERRUPT VECTOR
001022  000362                  DSVAD2: 000362                  ;POINTS TO THE 2ND WORD OF THE DISPLAY INTERRUPT VECTOR
001024  000000                  WCHAR:  000000
001026  177670                  DCSR:   177670                  ;POINTS TO THE DISPLAYS CONTROL/STATUS WORD
001030  000370                  IDTP:   000370                  ;POINTS TO THE INITIAL DISPLAY TABLE POINTER
001032  000366                  CDTP:   000366                  ;POINTS TO THE CURRENT DISPLAY TABLE POINTER
001034  000364                  DCP:    000364                  ;POINTS TO THE DISPLAY CHARACTER POINTER
001036  000000                  CHRCNT: 000000                  ;COUNT OF CHARACTERS IN A MESSAGE
001040  000000                  PASCNT: 000000                  ;TALLY OF PASSES COMPLETED BY THIS PROG
001042  000000                  TTYAVA: 000000                  ;SET TO # 1 IF A TTY IS AVAILABLE
001044  000000                  TUBTM0: 000000                  ;TEMPORARY STORAGE FOR DISPLAY ROUTINES
001046  000000                  TUBTM1: 000000                  ;TEMPORARY STORAGE FOR DISPLAY ROUTINES
001050  000000                  INTCNT: 000000                  ;HOLDS COUNT OF # OF INTERRUPTS
001052  000000                  FAKEY:  000000                  ;FAKE TTY STATUS REG
001054  000000                  KBID0:  000000                  ;TEMP STORAGE FOR ID# TEST
001056  000000                  ERRPAS: 000000                  ;# OF ERRORS ON THIS PASS
001060  000022                  MAXBLK: 000022                  ;MAXIMUM # OF BLOCKS/LINE IN TEST 5
```

```
                                .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:
                                ;*      FM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT

001062                          $ERRTB:
                                ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            BELL ON ERROR
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>
                                ;*       7-0            #OF TEST TO LOOP ON IF SWR<8> IS SET
                                ;*       13             DISABLE ERROR MESSAGES
                                ;*       12             STALL AT EACH TEST FOR A SECOND OR TWO
                                .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

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE FOR LINE FEED
000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
```

```
                        000007          R7=     %7              ;;GENERAL REGISTER
                        000006          SP=     %6              ;;STACK POINTER
                        000007          PC=     %7              ;;PROGRAM COUNTER

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

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

                                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                        100000          BIT15=  100000
                        040000          BIT14=  40000
                        020000          BIT13=  20000
                        010000          BIT12=  10000
                        004000          BIT11=  4000
                        002000          BIT10=  2000
                        001000          BIT09=  1000
                        000400          BIT08=  400
                        000200          BIT07=  200
                        000100          BIT06=  100
                        000040          BIT05=  40
                        000020          BIT04=  20
                        000010          BIT03=  10
```

```
                        000004          BIT02=  4
                        000002          BIT01=  2
                        000001          BIT00=  1
                                        .EQUIV  BIT09,BIT9
                                        .EQUIV  BIT08,BIT8
                                        .EQUIV  BIT07,BIT7
                                        .EQUIV  BIT06,BIT6
                                        .EQUIV  BIT05,BIT5
                                        .EQUIV  BIT04,BIT4
                                        .EQUIV  BIT03,BIT3
                                        .EQUIV  BIT02,BIT2
                                        .EQUIV  BIT01,BIT1
                                        .EQUIV  BIT00,BIT0

                                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
                        000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
                        000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
                        000014          TBITVEC=14              ;;"T" BIT
                        000014          TPTVEC= 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
                                        .SBTTL  STARTING ADDRESS(ES)
000200  000137  001062          JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
                                        ;VARIABLES AND POINTERS AND CONSTANTS AND STUFF
```

```
                                .SBTTL  COMMON TAGS

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

                                        .=600
000600                          $CMTAG:                         ;;START OF COMMON TAGS
000600  000000                  $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
000602  000                     $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
000603  000                     $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
000604  000000                  $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
000606  000000                  $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
000610  000000                  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
000612  000000                  $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
000614  000                     $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
000615  001                     $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
000616  000000                  $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
000620  000000                  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
000622  000000                  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
000624  000000                  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
000626  000000                  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
000630  000000                          .WORD   0               ;;RESERVED--NOT TO BE USED
000632  000000                          .WORD   0
000634  000                     $AUTOR: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
000635  000                     $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
000636  000000                          .WORD   0
000640  177570                  SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
000642  177570                  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
000644  177560                  $TKS:   177560                  ;;TTY KBD STATUS
000646  177562                  $TKB:   177562                  ;;TTY KBD BUFFER
000650  177564                  $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
000652  177566                  $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
000654  000                     $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
000655  002                     $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
000656  012                     $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
000657  000                     $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
000660  000000                  $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                                ;;WHICH  ($REG0) WAS OBTAINED
000662  000000                  $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
000664  000000                  $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
000666  000000                  $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
000670  000000                  $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
000672  000000                  $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
000674  000000                  $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
000676  000000                  $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
000700  000000                  $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
000702  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
000704  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
000706  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
000710  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
000712  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
000714  000000                  $TMP5:  .WORD   0               ;;USER DEFINED
000716  000000                  $TMP6:  .WORD   0               ;;USER DEFINED
000720  000000                  $TMP7:  .WORD   0               ;;USER DEFINED
000722  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
```

```
000724  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
000726  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
000732     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
000733     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
000734  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*************************************************************
000736  000000                  CNXFER: 000000
000740  000000                  CNCHAR: 000000
000742  000000                  CNRECV: 000000
000744  000000                  CNERRO: 000000
000746  000000                  BM0:    000000
000750  000000                  BM1:    000000
000752  000000                  BM2:    000000
000754  000000                  SPMODF: 000000
000756  000000                  TUBSWT: 000000                  ;SWITCH
000760  000000                  LEAVEC: 000000                  ;SET IF DEVICE ADDRESSES AND VECTOR ADDRESSES
                                                                ;ARE TO BE LEFT ALONE BY
                                                                ;THE FINDTT ROUTINE
000762  000001                  UPFAST: 000001                  ;THIS IS THE SPEED CONSTANT FOR FAST PANNING UP
000764  000001                  DOWNFA: 000001                  ;THIS IS THE SPEED CONSTANT FOR FAST PANNING DOWN
000766  000000                  TEMP:   000000
000770  000000                  LINCNT: 000000                  ;COUNT OF # OF LINES TO BE DISPLAYED
000772  000000                  STLCNT: 000000                  ;COUNTER DEVOTED TO THE STALL ROUTINE
000774  177564                  TPS:    177564                  ;POINTS TO ANY ADDITIONAL TELLEPRINTER STATUS WORD
000776  177566                  TPB:    177566                  ;POINTS TO ANY ADDITIONAL TELLEPRINTER BUFFERS
001000  177570                  KBSR:   177570                  ;POINTS TO THE VT71 KEYBOARD STATUS REG
001002  177572                  KBUF:   177572                  ;POINTS TO THE VT71 KEYBOARD BUFFER REGISTER
001004  177574                  LCSR:   177574                  ;POINTS TO THE LED STATUS/CONTROL REGISTER
001006  177576                  LBUF:   177576                  ;POINTS TO THE LED BUFFER
001010  000070                  KBVAD1: 000070                  ;POINTS TO THE 1ST WORD OF THE KEYBOARD INT VECTOR
001012  000072                  KBVAD2: 000072                  ;POINTS TO THE 2ND WORD OF THE KEYBOARD INT VECTOR
001014  000074                  LDVAD1: 000074                  ;VECTOR PC WORD
001016  000076                  LDVAD2: 000076                  ;VECTOR STATUS WORD
001020  000360                  DSVAD1: 000360                  ;POINTS TO THE 1ST WORD OF THE DISPLAY INTERRUPT VECTOR
001022  000362                  DSVAD2: 000362                  ;POINTS TO THE 2ND WORD OF THE DISPLAY INTERRUPT VECTOR
001024  000000                  WCHAR:  000000
001026  177670                  DCSR:   177670                  ;POINTS TO THE DISPLAYS CONTROL/STATUS WORD
001030  000370                  IDTP:   000370                  ;POINTS TO THE INITIAL DISPLAY TABLE POINTER
001032  000366                  CDTP:   000366                  ;POINTS TO THE CURRENT DISPLAY TABLE POINTER
001034  000364                  DCP:    000364                  ;POINTS TO THE DISPLAY CHARACTER POINTER
001036  000000                  CHRCNT: 000000                  ;COUNT OF CHARACTERS IN A MESSAGE
001040  000000                  PASCNT: 000000                  ;TALLY OF PASSES COMPLETED BY THIS PROG
001042  000000                  TTYAVA: 000000                  ;SET TO # 1 IF A TTY IS AVAILABLE
001044  000000                  TUBTM0: 000000                  ;TEMPORARY STORAGE FOR DISPLAY ROUTINES
001046  000000                  TUBTM1: 000000                  ;TEMPORARY STORAGE FOR DISPLAY ROUTINES
001050  000000                  INTCNT: 000000                  ;HOLDS COUNT OF # OF INTERRUPTS
001052  000000                  FAKEY:  000000                  ;FAKE TTY STATUS REG
001054  000000                  KBID0:  000000                  ;TEMP STORAGE FOR ID# TEST
001056  000000                  ERRPAS: 000000                  ;# OF ERRORS ON THIS PASS
001060  000022                  MAXBLK: 000022                  ;MAXIMUM # OF BLOCKS/LINE IN TEST 5
```

M1

```
                                .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 SITEMB, THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
                                ;*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

001062                          SERRTB:


001062                          START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001062  012706  000600                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001066  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001070  022706  000640                  CMP     #SWR,R6 ;;DONE?
001074  001374                          BNE     .-6             ;;LOOP BACK IF NO
001076  012706  000600                  MOV     #600,SP         ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
001102  012737  014210  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001110  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
001116  013737  012762  012754          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
001124  005037  000722                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
001130  012737  001130  000606          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
001136  012737  001136  000610          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER, IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
001144  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
001150  012737  001204  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
001156  012737  177570  000640          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
001164  012737  177570  000642          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
001172  022777  177777  177440          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
001200  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
001202  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
001204  012716  001212          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
001210  000002                          RTI
001212  012737  000176  000640  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
001220  012737  000174  000642          MOV     #DISPREG,DISPLAY
001226  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

001232  012737  000176  000640          MOV     #SWREG,SWR      ;SETUP SOFTWARE SWITCH REGISTERS ADDRESS
001240  012737  000174  000642          MOV     #DISPREG,DISPLAY;AND THE SOFTWARE LIGHTS REGISTER
001246  012777  015246  177544          MOV     #DIHAN,@DSVAD1  ;SETUP DISPLAY INTERRUPT VECTOR
001254  012777  000340  177540          MOV     #340,@DSVAD2    ;VECTOR PRIORITY = 7
001262  004737  014136                  JSR     PC,CLRTUB       ;CLEAR THE VT71 DISPLAY TABLE
001266  012777  015760  177534          MOV     #DISTRL,@IDTP   ;SETUP POINTER TO DISPLAY TABLE
001274  012737  000001  000600          MOV     #1,$PASS        ;INITIALIZE THE PASS COUNT
001302  005037  000612                  CLR     $ERTTL          ;ZERO ERROR TOTAL FOR ALL PASSES
```

```
                                        ;FINDOUT IF AN EXTRA TERMINAL IF HOOKED UP TO THE VT71
001306  005737  000760          FINDTT: TST     LEAVEC          ;SHOULD WE LEAVE ADDRESSES ALONE?
001312  001100                          BNE     T0000           ;IF SO RO RIGHT INTO THE TESTS
001314  012737  001426  000004          MOV     #18,@#4         ;SETUP VECTOR IN CASE OF TRAP
001322  005037  000006                  CLR     @#6             ;VECTOR PRIORITY = 0
001326  005737  177570                  TST     @#177570        ;TRY TO ACCESS ADDRESS 1 HIGHER THAN THE
                                                                ;STANDARD VT71 KEYBOARD CONTROL REGISTER ADDRESS
001332  000240                          NOP                     ;WE DIDNT TRAP, WE NOW ASSUME THAT AN ADDITIONAL
001334  012737  177570  001000          MOV     #177570,KBSR    ;TERMINAL IS IN USE
001342  012737  177572  001002          MOV     #177572,KBUF    ;THAT USES THE STANDARD ADDRESS
001350  012737  177574  001004          MOV     #177574,LCSR    ;AND THAT THE VT71 ITSELF IS IN
001356  012737  177576  001006          MOV     #177576,LBUF    ;THE NEXT HIGHEST ADDRESS
001364  012737  000070  001010          MOV     #70,KBVAD1      ;SET KEYBOARD VECTOR ADDRESS
001372  012737  000072  001012          MOV     #72,KBVAD2      ;SETUP VECTOR PRIORITY ADDRESS
001400  012737  000074  001014          MOV     #74,LDVAD1      ;SETUP LED VECTOR ADDRESS
001406  012737  000076  001016          MOV     #76,LDVAD2      ;AND THE LED VECTOR PRIORITY
001414  012737  000001  001042          MOV     #1,TTYAVA       ;SET XTRA TTY AVAILABLE SWITCH
001422  000137  001514                  JMP     T0000           ;GO TO THE FIRST TEST

001426  012737  177560  001000  18:     MOV     #177560,KBSR    ;WE TRAPED, ASSUME THAT NO ADDITIONAL
001434  012737  177562  001002          MOV     #177562,KBUF    ;TERMINAL IS IN USE
001442  012737  177564  001004          MOV     #177564,LCSR    ;AND THAT THE VT71 ITSELF USES
001450  012737  177566  001006          MOV     #177566,LBUF    ;THE STANDARD ADDRESS
001456  012737  000060  001010          MOV     #60,KBVAD1      ;SET KEYBOARD VECTOR ADDRESS
001464  012737  000062  001012          MOV     #62,KBVAD2      ;SETUP VECTOR PRIORITY ADDRESS
001472  012737  000064  001014          MOV     #64,LDVAD1      ;SETUP LED VECTOR ADDRESS
001500  012737  000066  001016          MOV     #66,LDVAD2      ;AND THE LED VECTOR PRIORITY
001506  012737  000000  001042          MOV     #0,TTYAVA       ;CLEAR XTRA TTY AVAILABLE SWITCH
001514  004737  015002          T0000:  JSR     PC,LSTALL       ;MAKE IT LOOK NICE
001520  005037  001056                  CLR     ERRPAS          ;CLEAR OUT # OF ERRORS ON THIS PASS
001524  012737  015420  000004          MOV     #TRAPER,@#4     ;SETUP TRAP VECTOR TO POINT TO HANDLING ROUTINE
001532  012737  000340  000006          MOV     #340,@#6        ;SETUP VECTOR PRIORITY = NO INTERRUPTS
001540  012777  015244  177242          MOV     #KBSRV,@KBVAD1  ;SETUP VECTOR IN CASE OF A KEYBOARD INTERRUPT
001546  012777  000340  177236          MOV     #340,@KBVAD2    ;VECTOR PRIORITY = 7
001554  012777  015244  177232          MOV     #LDSRV,@LDVAD1  ;SETUP VECTOR IN CASE OF A LED INTERRUPT
001562  012777  000340  177226          MOV     #340,@LDVAD2    ;LED VECTOR PRIORITY = 7
001570  004737  014136                  JSR     PC,CLRTUB       ;CLEAR OUT THE DISPLAY TABLE
001574  012777  015760  177226          MOV     #DISTBL,@IDTP   ;SETUP THE POINTER TO THE DISPLAY TABLE
001602  023727  000600  000001          CMP     $PASS,#1        ;IS THIS THE 1ST PASS?
001610  001067                          BNE     T0001           ;IF NOT, GO TO THE FIRST TEST
001612  017700  177022                  MOV     @SWR,R0         ;GET TEST #
001616  042700  177600                  BIC     #177600,R0      ;CLEAR ALL BITS EXCEPT FOR TEST # BITS
001622  010037  000602                  MOV     R0,@TSTNM       ;SET TEST # FOR SCOPE ROUTINE
001626  001460                          BEQ     T0001           ;IF SWR<7-0> = 0 GO DIRECTLY TO THE FIRST TEST
001630  005337  000602                  DEC     @TSTNM          ;1ST SCOPE INCREMENTS THE TEST #
001634  000241                          CLC                     ;MAKE SURE THE C BIT ROTATES IN CLEAR
001636  006100                          ROL     R0              ;MULT TEST # BY 2
001640  062700  023050                  ADD     #TSTLST,R0      ;BUILD POINTER TO TABLE OF TEST ADDRESSES
001644  011001                          MOV     (R0),R1
001646  012737  015420  000004          MOV     #TRAPER,@#4     ;SETUP THE TIMEOUT TRAP VECTOR
001654  012737  000340  000006          MOV     #340,@#6        ;SETUP THE TRAP VECTOR ALSO
001662  023727  000602  000037          CMP     @TSTNM,#37      ;IS IT TEST 37?
```

```
                                        BEQ     2$              ;IF SO, DONT WRITE THE CHAR SET FIRST
                                        CMP     $TSTNM,#36      ;IS IT TEST 36?
                                        BEQ     2$              ;IF SO, DONT WRITE THE CHAR SET FIRST
                                        CMP     $TSTNM,#3       ;IS IT TEST 1 OR 2 OR 3?
                                        BLO     2$              ;IF SO, SKIP OVER THE LOADING OF THE CHARACTER SET
                                        CLR     TUBSWT          ;NO ERROR MESSAGE ON VT71 SCREEN
                                        MOV     #146,@DCSR      ;START UP THE DISPLAY
                                        JSR     PC,@STALL       ;STALL TO MAKE SURE IT IS GOING
                                        MOV     #ENDCHR,R5      ;SETUP CHAR SET END ADDRESS
                                        MOV     #CHARS,@DCP     ;LOAD THE CHARACTER SET
                                        MOV     #100146,@DCSR   ;START LOADING
                                        JSR     PC,TSTLOD       ;MAKE SURE IT HAPPENS CORRECTLY
                                2$:     MOV     R1,$LPADR       ;SETUP AN INITIAL SCOPE LOOPBACK ADDRESS
                                        ADD     #2,$LPADR       ;MAKE IT RIGHT AFTER THE "SCOPE"
                                        JMP     (R1)            ;GO TO TEST SPECIFIED


                                .SBTTL  TEST THAT THE DCSR REGISTER CAN BE ACCESSED WITHOUT TRAPPING
                                ;T0001  THIS TEST FIRST SETS UP VECTOR LOCATIONS 4 & 6 IN CASE OFA TRAP.
                                ;       THEN IT WRITES INTO THE DCSR REGISTER LOCATION WITH A "CLR"
                                ;       INSTRUCTION.  IF A TRAP RESULTS, AN ERROR MESSAGE IS PRINTED.
                                ;       ***WARNING***  THIS TEST DOES NOT ATTEMPT TO DISPLAY ANY MESSAGES
                                ;       ON THE VT71 SCREEN. ERROR MESSAGES WILL BE SENT ONLY TO THE
                                ;       ADDITIONAL TERMINAL IF AVAILABLE.
                                T0001:  SCOPE
                                        JSR     PC,FXTST        ;DISPLAY TEST #
                                        MOV     #1$,$LPADR      ;LOOP BACK TO 1$ IF ERROR AND SWR<14>=1
                                        MOV     #2$,@#4         ;SETUP ERROR TRAP VECTOR
                                        MOV     #340,@#6        ;SETUP TRAP VECTOR PRIORITY
                                1$:     MTPS    #340            ;NO INTERRUPTS PLEASE
                                        MOV     #600,SP         ;SETUP THE STACK POINTER
                                        MOV     #0,@DCSR        ;ACCESS THE DCSR REG. SEE IF WE TRAP
                                        BR      T0002           ;NO TRAP! GO ON TO THE NEXT TEST
                                2$:     BIT     #40000,@SWR     ;LOOP?
                                        BNE     1$              ;YUP. DO IT NOW IN THE INTRESTS OF A TIGHT LOOP
                                        MOV     #MSG22,R5       ;TRAPPED. GET THE ERROR MESSAGE ADDRESS
                                        JSR     PC,TTYOUT       ;AND PRINT THE ERROR MESSAGE
                                3$:     INCB    $ERFLG          ;SET THE ERROR FLAG
                                        BEQ     3$              ;MAKE SURE IT IS NOT 0
                                        INC     ERRPAS          ;ADD 1 TO THE TOTAL # OF ERRORS ON THIS PASS


                                .SBTTL  TEST THAT A SMALL PORTION OF THE CHARACTER SET CAN BE LOADED
                                ;T0002  THIS TEST TRIES TO LOAD 1 CHARACTER INTO THE CHARACTER GENERATOR
                                ;       RAM. IF LOADING IS NOT FINISHED AT THE  END OF A 2 SECOND WAIT,
                                ;       THE FOLLOWING ERROR MESSAGE IS DISPLAYED...
                                ;        CHARACTER SET DID NOT LOAD PROPERLY
                                ;        DCP WAS ??????
                                ;        DCP SHOULD HAVE BEEN XXXXX
                                ;       ***WARNING***  THIS TEST DOES NOT ATTEMPT TO DISPLAY ANY MESSAGES
                                ;       ON THE VT71 SCREEN. ERROR MESSAGES WILL BE SENT ONLY TO THE
                                ;       ADDITIONAL TERMINAL IF AVAILABLE.
                                T0002:  SCOPE
                                        JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
                                        MOV     #1$,$LPADR      ;LOOP BACK TO 1$ IF ERROR, AND SWR<13> IS CLEAR
                                        MOV     #TRAPER,4       ;SETUP TRAP VECTOR
```

```
 797  002114  012737  000340  000006          MOV     #340,6          ;SETUP VECTOR PRIORITY
 798
 799  002122  012777  022342  176704  18:     MOV     #PLUSS,@DCP     ;SETUP ADDRESS OF A CHARACTER TO LOAD
 800  002130  012777  100006  176670          MOV     #100006,@DCSR   ;SET LOAD BIT
 801  002136  012705  022412                  MOV     #ENDPLS,R5      ;SETUP END ADDRESS
 802  002142  005237  000756                  INC     TUBSWT          ;NO OUTPUT TO THE SCREEN
 803  002146  004737  015254                  JSP     PC,TSTLOD       ;WAIT FOR A GOOD LOAD
 804
 805
 806                                  .SBTTL  TEST THAT THE CHARACTER SET CAN BE LOADED
 807                                  ;T0003  THIS TEST TRIES TO LOAD THE ENTIRE CHARACTER SET. IF IT IS
 808                                  ;       NOT FINISHED LOADING AFTER A 2 SECOND PERIOD, THEN THE FOLLOWING
 809                                  ;       ERROR MESSAGE IS DISPLAYED...
 810                                  ;        CHARACTER SET DID NOT LOAD PROPERLY
 811                                  ;        DCP WAS ??????
 812                                  ;        DCP SHOULD HAVE BEEN XXXXX
 813                                  ;
 814  002152  000004                  T0003:  SCOPE
 815  002154  004737  014136                  JSR     PC,CLRTUB       ;START THIS TEST WITH A CLEAR SCREEN
 816  002160  004737  015102                  JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
 817  002164  004737  015002                  JSP     PC,LSTALL       ;NOT SO FAST
 818  002170  012737  002176  000606          MOV     #18,@LPADR      ;LOOP BACK TO 1$ IF ERROR, AND SWR<13> IS CLEAR
 819  002176  012777  016110  176630  18:     MOV     #CHARS,@DCP     ;SETUP ADDRESS OF THE CHARACTER SET TO LOAD
 820  002204  012777  100006  176614          MOV     #100006,@DCSR   ;SET LOAD BIT
 821  002212  012705  022340                  MOV     #ENDCHR,R5      ;SETUP END ADDRESS FOR COMPARE
 822  002216  005237  000756                  INC     TUBSWT          ;NO ERROR OUTPUT TO THE SCREEN YET.
 823  002222  004737  015254                  JSP     PC,TSTLOD       ;WAIT FOR LOAD TO BE COMPLETE
 824
 825
 826                                  .SBTTL  TEST CHARACTER SET ADDRESSING
 827                                  ;T0004  IN THIS TEST EVERY CHARACTER IS LOADED WITH SOMETHING
 828                                  ;       CHARACTER 000 IS LOADED WITH ALL BITS SET. CHARACTER 001 WITH
 829                                  ;       ALL BITS SET SAVE THOSE OF THE 1ST SCAN LINE. CHARACTER 002 WITH
 830                                  ;       ALL BITS SET EXCEPT FOT THE 1ST 2 SCANN LINES.
 831                                  ;       AND SO ON, UNTIL CHARACTER 012 IS LOADED WITH ALL BITS CLEAR.
 832                                  ;       THEN THE PATERN REPEATS ITSELF, STARTING WITH CHARACTER 013
 833                                  ;       WHICH IS LOADED WITH ALL BITS SET, AND SO ON AND SO ON
 834                                  ;       UNTIL THE END OF THE CHARACTER SET IS REACHED
 835                                  ;       AFTER LOADING IS DONE, A TEXT STRING IS DISPLAYED, THAT SHOULD
 836                                  ;       RESULT IN 24 "DECENDING STAIRCASE" PATTERNS
 837                                  ;       BEING DISPLAYED, ONE PER LINE OF THE SCREEN. EACH "STEP"
 838                                  ;       REPRESENTS A CHARACTER SET LOCATION. MISSING OR UNEVEN STEPS
 839                                  ;       INDICATE THAT EITHER THE CHARACTER REPRESENTED BY THE DEFECTIVE
 840                                  ;       STEP DID NOT LOAD PROPERLY, OR THET IT WAS AFFECTED BY THE
 841                                  ;       WRITING OF ANOTHER CHARACTER.
 842                                  ;
 843  002226  000004                  T0004:  SCOPE
 844  002230  004737  014136                  JSR     PC,CLRTUB       ;CLEAR OUT THE VT71 DISPLAY TABLE
 845  002234  004737  015102                  JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
 846  002240  012777  003054  176562          MOV     #108,@IDTP      ;SETUP DISPLAY TABLE POINTER
 847  002246  012777  000146  176552          MOV     #146,@DCSR      ;START UP THE DISPLAY
 848  002254  004737  015172                  JSR     PC,MSTALL       ;STALL TO MAKE SURE IT IS GOING
 849  002260  012737  002310  000606          MOV     #1$,@LPADR      ;LOOP BACK TO 1$ IF ERROR, AND SWR<13> IS CLEAR
 850  002266  012705  027727                  MOV     #MSG39,R5       ;GET ADDRESS OF THE "DONT PANIC- SCREEN IS
 851  002272  004737  014722                  JSR     PC,TTYOUT       ;GOING BLANK" MESSAGE, AND DISPLAY IT
 852  002276  004737  015002                  JSR     PC,LSTALL       ;WAIT
```

D2

```
 853  002302  012777  002544  176520          MOV     #98,@IDTP       ;SET UP A DISPLAY POINTER TO A DISPLAY TABLE
 854  002310                          18:     ;CLR    @DCSR           ;TURN OFF THE DISPLAY FOR A MOMENT
 855  002310  012704  000340                  MOV     #340,R4         ;LEFT SIDE
 856  002314  012701  000300                  MOV     #300,R1         ;RIGHT SIDE
 857  002320  012702  000013          28:     MOV     #11.,R2         ;11 CHARS THEN BEGIN A NEW LINE
 858  002324  012703  000012          38:     MOV     #10.,R3         ;SET FOR 20 WORDS PER CHARACTER DESCRIPTION
 859  002330  012700  022776                  MOV     #NUMBER,R0      ;SETUP ADDR OF THE CHARACTER
 860  002334  042704  000037          48:     BIC     #37,R4
 861  002340  042701  000037                  BIC     #37,R1
 862  002344  020302                          CMP     R3,R2
 863  002346  103004                          BHIS    58
 864  002350  052704  000037                  BIS     #37,R4
 865  002354  052701  000037                  BIS     #37,R1
 866  002360  010420                  58:     MOV     R4,(R0)+
 867  002362  010120                          MOV     R1,(R0)+        ;RIGHT SIDE
 868  002364  005303                          DEC     R3
 869  002366  001362                          BNE     48
 870  002370  005020                          CLR     (R0)+
 871  002372  012777  022776  176434          MOV     #NUMBER,@DCP    ;SETUP ADDR OF CHARACTER SET PORTION
 872  002400  012777  100146  176420          MOV     #100146,@DCSR   ;AND LOAD THE PORTION
 873  002406  012705  023046                  MOV     #ENDNUM,R5      ;SETUP END OF CHAR SET ADDRESS FOR COMPARE
 874  002412  004737  015254                  JSR     PC,TSTLOD       ;WAIT FOR LOADING TO FINISH
 875  002416  062704  000400                  ADD     #400,R4         ;BUMP LEFT SIDE TO REFLECT THE NEXT CODE
 876  002422  062701  000400                  ADD     #400,R1         ;BUMP RIGHT SIDE TO REFLECT THE NEXT CODE
 877  002426  020127  000337                  CMP     R1,#337
 878  002432  001403                          BEQ     78
 879  002434  005302                          DEC     R2
 880  002436  001332                          BNE     38
 881  002440  000727                          BR      28
 882                          ;NOW DISPLAY THE THE WHOLE THING AT ONCE
 883  002442  000240                  78:     NOP
 884  002444  012777  002550  176356          MOV     #98,@IDTP       ;SETUP A NEW DISPLAY TABLE
 885  002452  012777  000106  176346          MOV     #106,@DCSR
 886  002460  004737  015002                  JSR     PC,LSTALL
 887  002464  004737  015002                  JSR     PC,LSTALL
 888  002470  004737  014136                  JSR     PC,CLRTUB       ;CLEAR OUT THE REGULAR DISPLAY TABLE
 889  002474  012777  000146  176324          MOV     #146,@DCSR      ;ENABLE SPECIAL CHARACTERS AGAIN, SINCE 1ST CHAR
 890                                                                  ;IN THE CURRENT DISPLAY TEXT IS AN (EOS), THE
 891                                                                  ;SCREEN SHOULD GO BLANK
 892  002502  004737  015002                  JSR     PC,LSTALL       ;GIVE TIME
 893  002506  012777  016110  176320          MOV     #CHARS,@DCP     ;SETUP POINTER TO REGULAR CHARACTER SET
 894  002514  012705  022340                  MOV     #ENDCHR,R5      ;SETUP END OF CHAR SET ADDRESS
 895  002520  012777  100146  176300          MOV     #100146,@DCSR   ;LOAD THE REGULAR CHARACTER SET
 896  002526  004737  015254                  JSR     PC,TSTLOD       ;WAIT FOR LOAD TO FINISH
 897  002532  012777  015760  176270          MOV     #DISTBL,@IDTP   ;POINT DISPLAY TABLE POINTER TO REGULAR DISPLAY TABLE
 898  002540  000137  003064                  JMP     T0005           ;GO ON TO THE NEXT TEST
 899
 900  002544  003776                  88:     3776
 901  002546  030064                          MSG41
 902
 903  002550                          98:
 904  002550  003764                          3764
 905  002552  026715                          MSG35A
 906  002554  003672                          3672
 907  002556  027325                          MSG35Y
 908  002560  003764                          3764
```

```
002562  076730          MSG35B
002564  003672          3672
002566  027325          MSG35Y
002570  003764          3764
002572  076743          MSG35C
002574  003672          3672
002576  027325          MSG35Y
002600  003764          3764
002602  076756          MSG35D
002604  003672          3672
002606  027325          MSG35Y
002610  003764          3764
002612  076771          MSG35E
002614  003672          3672
002616  027325          MSG35Y
002620  003764          3764
002622  077004          MSG35F
002624  003672          3672
002626  027325          MSG35Y
002630  003764          3764
002632  077017          MSG35G
002634  003672          3672
002636  027325          MSG35Y
002640  003764          3764
002642  077032          MSG35H
002644  003672          3672
002646  027325          MSG35Y
002650  003764          3764
002652  077045          MSG35I
002654  003672          3672
002656  027325          MSG35Y
002660  003764          3764
002662  077060          MSG35J
002664  003672          3672
002666  027325          MSG35Y
002670  003764          3764
002672  077073          MSG35K
002674  003672          3672
002676  027325          MSG35Y
002700  003764          3764
002702  077106          MSG35L
002704  003672          3672
002706  027325          MSG35Y
002710  003764          3764
002712  077121          MSG35M
002714  003672          3672
002716  027325          MSG35Y
002720  003764          3764
002722  077134          MSG35N
002724  003672          3672
002726  027325          MSG35Y
002730  003764          3764
002732  077147          MSG35O
002734  003672          3672
002736  027325          MSG35Y
002740  003764          3764
```

```
 965  002742  027162                          MSG35P
 966  002744  003672                          3672
 967  002746  027325                          MSG35Y
 968  002750  003764                          3764
 969  002752  027175                          MSG35Q
 970  002754  003672                          3672
 971  002756  027325                          MSG35Y
 972  002760  003764                          3764
 973  002762  027210                          MSG35R
 974  002764  003672                          3672
 975  002766  027325                          MSG35Y
 976  002770  003764                          3764
 977  002772  027223                          MSG35S
 978  002774  003672                          3672
 979  002776  027325                          MSG35Y
 980  003000  003764                          3764
 981  003002  027236                          MSG35T
 982  003004  003672                          3672
 983  003006  027325                          MSG35Y
 984  003010  003764                          3764
 985  003012  027251                          MSG35U
 986  003014  003672                          3672
 987  003016  027325                          MSG35Y
 988  003020  003764                          3764
 989  003022  027264                          MSG35V
 990  003024  003672                          3672
 991  003026  027325                          MSG35Y
 992  003030  003764                          3764
 993  003032  027277                          MSG35W
 994  003034  003672                          3672
 995  003036  027325                          MSG35Y
 996  003040  003764                          3764
 997  003042  027312                          MSG35X
 998  003044  003672                          3672
 999  003046  027325                          MSG35Y
1000  003050  100000                          100000
1001  003052  002550                          98
1002  003054  003761                  108:    3761
1003  003056  024611                          MSG17
1004  003060  003706                          3706
1005  003062  027727                          MSG39
1006
1007
1008                                  .SBTTL  DISPLAY A NUMBER OF BLOCKS PER LINE ON ALL LINES
1009                                  ;T0005  THIS TEST SIMPLY DISPLAYS ALTERNATING BLOCKS OF
1010                                  ;       "A"'S, "B"'S, AND "C"'S.
1011                                  ;TO MODIFY THE # OF BLOCKS TO BE DISPLAYED, MODIFY LOCATION "MAXBLK".
1012                                  ;ITS DEFAULT VALUE IF 8.
1013                                  ;THE LAST BLOCK ON EACH LINE   CONTAINS A EOL CHAR
1014  003064  000004                  T0005:  SCOPE
1015                                  ;
1016                                  ;
1017  003066  004737  015102                  JSR     PC,FXTST        ;DISPLAY THE TEST #
1018  003072  012737  003336  000606          MOV     #88,@LPADR      ;SETUP ERROR LOOP ADDRESS
1019  003100  012701  000117                  MOV     #79.,R1         ;# OF POSITIONS TO BE FILLED
1020  003104  005002                          CLR     R2              ;ZERO BLOCK LENGTH
```

```
1021  003106  013703  001060          MOV     MAXBLK,R3       ;COPY MAXIMUN BLOCK #
1022  003112  042703  177740          BIC     #177740,R3      ;DONT DO TEST FOR MORE THAN 32 BLOCKS/LINE
1023  003116  001507                  BEQ     6$              ;AND IF 0 BLOCKS SPECIFIED, DO 1 INSTEAD
1024  003120  005303                  DEC     R3              ;RESERVE 1 FOR EOL
1025  003122  160301          1$:     SUB     R3,R1           ;DIVIDE BY SUBTRACTION
1026  003124  100402                  BMI     2$
1027  003126  005202                  INC     R2              ;ADD 1 TO BLOCK LENGTH
1028  003130  000774                  BR      1$              ;KEEP DIVIDING
1029  003132  060301          2$:     ADD     R3,R1           ;GET THE REMAINDER OF THE DIVISION
1030  003134  060201                  ADD     R2,R1           ;LENGTH OF 1ST BLOCK IS REGULAR LENGTH PLUS REMAINDER
1031  003136  005101                  COM     R1              ;NEGATE IT TO MAKE IT A BLOCK COUNT FOR THE DISPLAY TABL
1032  003140  042701  174000          BIC     #174000,R1      ;DONT LEAVE ANY SPECIAL DISPLAY MODE BITS SET.
1033  003144  005102                  COM     R2              ;DO THE SAME THING TO THE REGULAR BLOCK COUNT
1034  003146  042702  174000          BIC     #174000,R2      ;NO SPECIAL DISPLAY MODE BITS SET
1035  003152  005037  000766          CLR     TEMP            ;START OFF WITH A'S
1036                                ;NOW R1=LENGTH OF THE 1ST BLOCK
1037                                ;    R2=LENGTH OF OTHER BLOCKS
1038                                ;    R3=# OF BLOCKS PER LINE -1
1039  003156  012700  003352              MOV     #9$,R0  ;SETUP ADDRESS OF DISPLAY TABLE BEGINING
1040  003162  012704  000024              MOV     #20.,R4 ;INIT LINE COUNT TO 20
1041  003166  013737  000766          3$:     MOV     R4,TEMP         ;FIGURE OUT WHICH LETTER TO START WITH BY THE
1042  003172  042737  177774  000766        BIC     #177774,TEMP    ;LINE # WE ARE DOING NOW
1043  003200  010305                  MOV     R3,R5           ;SETUP THE COUNT OF BLOCKS ON A LINE
1044  003202  010120                  MOV     R1,(R0)+        ;PUT COUNT FOR 1ST BLOCK INTO THE DISPLAY TABLE
1045  003204  005337  000766          4$:     DEC     TEMP            ;FIND OUT WHICH CHARACTER TO DISPLAY 1ST
1046  003210  001003                  BNE     5$
1047  003212  012720  025374          MOV     #MSG23D,(R0)+   ;USE "D"'S
1048  003216  000423                  BR      7$
1049  003220  023727  000766  000001  5$:     CMP     TEMP,#1
1050  003226  001003                  BNE     6$
1051  003230  012720  025252          MOV     #MSG23C,(R0)+   ;USE "C"'S
1052  003234  000414                  BR      7$
1053  003236  023727  000766  000002  6$:     CMP     TEMP,#2         ;SHOULD WE BE USING "A" OR "D"
1054  003244  001003                  BNE     10$
1055  003246  012720  025130          MOV     #MSG23B,(R0)+   ;USE "B"'S
1056  003252  000405                  BR      7$
1057  003254  012720  025006          10$:    MOV     #MSG23A,(R0)+   ;USE "A"'S
1058
1059  003260  012737  000003  000766        MOV     #3,TEMP
1060  003266  010220          7$:     MOV     R2,(R0)+
1061  003270  005305                  DEC     R5              ;MORE BLOCKS TO DO ON THIS LINE?
1062  003272  001344                  BNE     4$              ;IS SO, GO BACK AND DO EM
1063  003274  012760  003776  177776        MOV     #3776,-2(R0)    ;MODIFY BLOCK LENGTH FOR FINAL BLOCK
1064  003302  012720  024315                MOV     #MSG15,(R0)+    ;TEXT IS A EOL
1065  003306  005304                  DEC     R4              ;TAKE 1 FROM THE # OF LINES LEFT TO DO
1066  003310  001326                  BNE     3$              ;IF ANY REMAIN TO BE DONE, DO THEM
1067  003312  012720  100000                MOV     #100000,(R0)+   ;PUT A JUMP INTO THE DISPLAY TABLE
1068  003316  012720  003352                MOV     #9$,(R0)+       ;BACK TO THE BEGINING
1069  003322  012777  003352  175500        MOV     #9$,@IDTP       ;USE A SPECIAL DISPLAY TABLE FOR THIS TEST
1070  003330  012777  000146  175470        MOV     #146,@DCSR      ;TURN ON THE DISPLAY
1071  003336  004737  015002          8$:     JSR     PC,LSTALL       ;LET THE RESULTS BE SEEN
1072  003342  004737  015002                JSR     PC,LSTALL       ;AND SEEN WELL
1073  003346  000137  007352                JMP     @#6$            ;NOW GO ON TO THE NEXT TEST
1074
1075  003352  002000          9$:     .BLKW   2000
1076
```

```
                                        .SBTTL  MAKE SURE THAT THE DISPLAY DOES INTERRUPTS AT LEVEL 0
                                        ;T0006  IN THIS TEST, THE LSI-11 PRIORITY IS SET TO 0
                                        ;       AND THE RUNNING DISPLAY IS EXPECTED TO INTERRUPT WITHIN
                                        ;       ABOUT A TENTH  OF A SECOND. IF IT DOES NOT INTERRUPT, THE FOLLOWING
                                        ;       ERROR MESSAGE IS DISPLAYED....
                                        ;        DISPLAY DID NOT INTERRUPT FOR A FULL TENTH  OF A SECOND
                                        ;
007352  000004                        T0006:  SCOPE
007354  004737  014136                        JSR     PC,CLRTUB       ;CLEAR OUT THE DISPLAY TABLE
007360  012737  007412  000606                MOV     #18,@LPADR      ;LOOP BACK TO 18 IF ERROR, AND SWR<13> IS CLEAR
007366  004737  015102                        JSR     PC,FXTST        ;CHANGE TEST # TO ASCII
007372  012777  007454  171430                MOV     #98,@TDTP       ;SETUP DISPLAY TABLE POINTER
007400  012777  000146  171420                MOV     #146,@DCSR      ;TURN ON THE DISPLAY
007406  004737  015002                        JSR     PC,LSTALL       ;LEAVE THE TEST # ON THE SCREEN FOR A SECOND
007412  005037  001050                18:     CLR     INTCNT          ;ZERO THE INTERRUPT COUNT
007416  106427  000000                        MTPS    #0              ;ENABLE INTERRUPTS
007422  004737  015172                        JSR     PC,MSTALL       ;WAIT
007426  106427  000340                        MTPS    #340            ;THATS ENOUGH TIME FOR AT LEAST ONE INTERRUPT
007432  005737  001050                        TST     INTCNT          ;DID ANY INTERRUPTS HAPPEN WHILE WE WERE WAITING?
007436  001012                              BNE     T0007           ;IF SO JUST GO ON TO THE NEXT TEST
007440  012705  026161                        MOV     #MSG2B,R5       ;NO INTERRUPTS, SETUP ADDRESS OF ERROR MESSAGE TEXT
007444  004737  014532                        JSR     PC,ERMES        ;AND DISPLAY THE ERROR MESSAGE
007450  000137  007464                        JMP     T0007           ;GO TO THE NEXT TEST
007454  003761                        98:     3761
007456  024611                              MSG17
007460  003776                              3776
007462  030064                              MSG41


                                        .SBTTL  MAKE SURE THAT THE DISPLAY DOES NOT INTERRUPT AT LEVEL 7
                                        ;T0007  IN THIS TEST THE LSI-11 PRIORITY IS SET TO 7. IF THE RUNNING
                                        ;       DISPLAY CAUSES AN INTERRUPT WITHIN 1/10 OF A SECOND, THE FOLLOWING
                                        ;       ERROR MESSAGE IF DISPLAYED....
                                        ;        DISPLAY INTERRUPTED WITH LSI 11 PRIORITY SET TOO HIGH
                                        ;
007464  000004                        T0007:  SCOPE
007466  004737  015102                        JSR     PC,FXTST        ;CHANGE TEST # TO ASCII
007472  012777  007556  171330                MOV     #98,@TDTP       ;SETUP DISPLAY TABLE POINTER
007500  012737  007520  000606                MOV     #18,@LPADR      ;LOOP BACK TO 18 IF ERROR, AND SWR<13> IS CLEAR
007506  012777  000146  171312                MOV     #146,@DCSR      ;TURN ON THE DISPLAY
007514  004737  015002                        JSR     PC,LSTALL       ;LEAVE THE TEST # ON THE SCREEN FOR A SECOND
007520  005037  001050                18:     CLR     INTCNT          ;ZERO THE INTERRUPT COUNT
007524  106427  000340                        MTPS    #340            ;DISABLE INTERRUPTS
007530  004737  015172                        JSR     PC,MSTALL       ;WAIT
007534  005737  001050                        TST     INTCNT          ;DID ANY INTERRUPTS HAPPEN WHILE WE WERE WAITING?
007540  001412                              BEQ     T0010           ;IF NOT JUST GO ON TO THE NEXT TEST
007542  012705  027636                        MOV     #MSG3B,R5       ;NO INTERRUPTS, SETUP ADDRESS OF ERROR MESSAGE TEXT
007546  004737  014532                        JSR     PC,ERMES        ;AND DISPLAY THE ERROR MESSAGE
007552  000137  007566                        JMP     T0010           ;GO TO THE NEXT TEST
007556  003761                        98:     3761
007560  024611                              MSG17
007562  003776                              3776
007564  030064                              MSG41


                                        .SBTTL  MAKE SURE 'END OF LINE' CHARACTERS ARE RECOGNIZED
```

```
1133                                ;T0010  THIS TEST DISPLAYS A LINE OF TEXT THAT HAS END OF LINE CHARACTERS IN IT,
1134                                ;       IF EACH WORD OF THE MESSAGE IS NOT ON A SEPARATE LINE,
1135                                ;       IT MEANS THAT AN END OF LINE CHARACTER
1136                                ;       WAS NOT RECOGNIZED, IF THE END  OF LINE CHARACTERS WORK, THEN THE
1137                                ;       FOLLOWING SHOULD APPEAR ON THE SCREEN
1138                                ;       TEST # 000010
1139                                ;        EACH 1
1140                                ;        WORD 2
1141                                ;        OF 3
1142                                ;        THIS 201
1143                                ;        MESSAGE 202
1144                                ;        SHOULD 203
1145                                ;        BE 202
1146                                ;        ON 201
1147                                ;        A 3
1148                                ;        SEPARATE 2
1149                                ;        LINE 1
1150                                ;THE NUMBER AFTER EACH WORD REPRESENTS THE VALUE OF THE EOL CHAR THAT
1151                                ;IS AT THE END OF THE LINE, IF TWO WORDS OR MORE APPEAR ON THE SAME LINE, LOOK
1152                                ;EMBEDDED WITHIN THE LINE, THEY ARE THE EOL CHAR VALUES
1153                                ;THAT ARE NOT RECOGNIZED
1154                                ;
1155  007566  000004                T0010:  SCOPE
1156  007570  004737  015102                       JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
1157  007574  012737  007616  000606               MOV     #26,$LPADR      ;LOOP BACK TO 26 IF ERROR, AND SWR<13> IS CLEAR
1158  007602  012777  007640  171220               MOV     #98,@IDTP       ;SETUP POINTER TO SPECIAL DISPLAY TABLE
1159  007610  012777  000146  171210               MOV     #146,@DCSR      ;TURN ON THE DISPLAY
1160  007616  004737  015002              26:     JSR     PC,LSTALL       ;WAIT TO MAKE MESSAGE READABLE
1161  007622  004737  015002                       JSR     PC,LSTALL       ;LEAVE TIME FOR THE TEST TO BE VIEWED
1162  007626  004737  015002                       JSR     PC,LSTALL       ;LEAVE TIME FOR THE TEST TO BE VIEWED
1163  007632  004737  015002                       JSR     PC,LSTALL       ;LEAVE TIME FOR THE TEST TO BE VIEWED
1164  007636  000406                               BR      T0011
1165  007640  003761                      98:     3761
1166  007642  024611                              MSG17
1167  007644  003630                              3630
1168  007646  024315                              MSG15
1169  007650  100000                              100000
1170  007652  016074                              TBL22
1171                                .SBTTL  MAKE SURE 'END OF LINE' CHARACTERS ARE  BE IGNORED IF  DCSR BIT 5 IS CLEAR
1172                                ;T0011  THIS TEST DISPLAYS A LINE OF TEXT THAT HAS END OF LINE CHARACTERS IN IT,
1173                                ;       IF ALL WORDS OF THE MESSAGE ON THE SCREEN ARE NOT ON THE SAME LINE,
1174                                ;       IT MEANS THAT AN END OF LINE CHARACTER
1175                                ;       WAS  RECOGNIZED, IF THE END  OF LINE CHARACTERS ARE IGNORED, THEN THE
1176                                ;       FOLLOWING SHOULD APPEAR ON THE SCREEN
1177                                ;
1178                                ;       TEST # 000011 ALL 1 WORDS 2 SHOULD 3 BE 201 ON 202 THE 203 SAME LINE
1179                                ;
1180                                ;THE NUMBER AFTER EACH WORD REPRESENTS THE VALUE OF THE EOL CHAR THAT
1181                                ;IS THERE, IF TWO OR MORE LINES APPEAR ON THE SCREEN, LOOK
1182                                ;AT THE NUMBERS AT THE END OF THE LINE, THEY ARE THE EOL CHARS THAT ARE NOT IGNORED
1183                                ;
1184  007654  000004                T0011:  SCOPE
1185  007656  004737  015102                       JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
1186  007662  012737  007710  000606               MOV     #26,$LPADR      ;LOOP BACK TO 26 IF ERROR, AND SWR<13> IS CLEAR
1187  007670  012777  007732  171132               MOV     #98,@IDTP       ;SETUP POINTER TO SPECIAL DISPLAY TABLE
1188  007676  004737  015210                       JSR     PC,XSTALL       ;WAIT FOR AN INTERRUPT
```

```
1189  007702  012777  000106  171116          MOV     #106,@DCSR      ;TURN ON THE DISPLAY
1190  007710  004737  015002              28: JSR     PC,LSTALL       ;WAIT TO MAKE MESSAGE READABLE
1191  007714  004737  015002                  JSR     PC,LSTALL       ;LEAVE TIME FOR THE TEST TO BE VIEWED
1192  007720  004737  015002                  JSR     PC,LSTALL       ;LEAVE TIME FOR THE TEST TO BE VIEWED
1193  007724  004737  015002                  JSR     PC,LSTALL       ;LEAVE TIME FOR THE TEST TO BE VIEWED
1194  007730  000406                          BR      T0012
1195  007732  003660                  98:     3660
1196  007734  025514                          MSG24
1197  007736  003734                  108:    3734
1198  007740  030624                          MSG57
1199  007742  100000                          100000
1200  007744  007736                          108
1201
1202                                  .SBTTL  MAKE SURE CHAR CODE 200 IS RECOGNIZED AS A EOS CHAR
1203
1204                                  ;MAKE SURE 'END OF TEXT' CHAR IS RECOGNIZED
1205                                  ;T0012  THIS TEST DISLAYS A END OF SCREEN CHARACTER(200)
1206                                  ;       FOLLOWED BY A ERROR MESSAGE. THE ERROR MESSAGE
1207                                  ;       SHOULD NEVER BE SEEN
1208  007746  000004                  T0012:  SCOPE
1209  007750  004737  015102                  JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
1210  007754  012737  010046  000606          MOV     #28,@LPADR      ;LOOP BACK TO 28 IF ERROR, AND SWR<13> IS CLEAR
1211  007762  012777  010104  171040          MOV     #98,@IDTP       ;RESET DISPLAY POINTER
1212  007770  012777  000146  171030          MOV     #146,@DCSR      ;RECOGNIZE EOL & EOS CHARS
1213  007776  004737  015210                  JSR     PC,XSTALL       ;WAIT LONG ENOUGH FOR AN INTERRUPT
1214  010002  012777  016110  171024          MOV     #CHARS,@DCP     ;POINT POINTER AT CHAR SET
1215  010010  012777  100146  171010          MOV     #100146,@DCSR   ;LOAD THE CHARACTER SET
1216  010016  012705  022340                  MOV     #ENDCHR,R5      ;SET UP ADDRESS OF CHAR SET END
1217  010022  004737  015254                  JSR     PC,TSTLOD       ;WAIT FOR LOAD TO BE FINISHED
1218  010026  012777  010104  170774          MOV     #98,@IDTP       ;NEW DISPLAY TABLE
1219  010034  012700  000100                  MOV     #100,R0         ;SETUP A COUNTER
1220  010040  012777  000146  170760          MOV     #146,@DCSR      ;TURN ON THE DISPLAY
1221  010046  004737  015172              28: JSR     PC,MSTALL       ;THIS WAIT MAKES THE DISPLAY READABLE
1222  010052  017701  170754                  MOV     @CDTP,R1        ;GET CURRENT DISPLAY POINTER
1223  010056  020127  010114                  CMP     R1,#108         ;IS THE DISPLAY POINTER OUT OF BOUNDS?
1224  010062  100003                          BPL     38              ;IF SO, GO REPORT AN ERROR
1225  010064  005300                          DEC     R0              ;ITS OK, ADD 1 TO COUNT
1226  010066  001367                          BNE     28              ;AND TRY AGAIN
1227  010070  000413                          BR      T0013           ;GO ON TO THE NEXT TEST
1228  010072  012705  024464              38: MOV     #MSG16A,R5      ;SETUP ADDRESS OF THE ERROR MESSAGE
1229  010076  004737  014532                  JSR     PC,ERMES        ;REPORT THE ERROR
1230  010102  000406                          BR      T0013           ;GO ON TO THE NEXT TEST
1231  010104  003761                  98:     3761
1232  010106  024611                          MSG17
1233  010110  003744                          3744
1234  010112  024534                          MSG16B
1235  010114  100000                  108:    100000
1236  010116  016074                          TBL22
1237
1238
1239                                  .SBTTL  MAKE SURE CHAR CODE 000 IS RECOGNIZED AS A EOS CHAR
1240                                  ;MAKE SURE 'END OF TEXT' CHAR IS RECOGNIZED
1241                                  ;T0013  THIS TEST DISLAYS A END OF SCREEN CHARACTER(000)
1242                                  ;       FOLLOWED BY A ERROR MESSAGE. THE ERROR MESSAGE
1243                                  ;       SHOULD NEVER BE SEEN
1244  010120  000004                  T0013:  SCOPE
```

```
1245  010122  004737  015102                JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
1246  010126  012737  010154  000606        MOV     #1$,$LPADR      ;LOOP BACK TO 1$ IF ERROR, AND SWR<13> IS CLEAR
1247  010134  012777  010212  170666        MOV     #9$,@IDTP       ;NEW DISPLAY TABLE
1248  010142  012700  000100                MOV     #100,R0         ;SETUP A COUNTER
1249  010146  012777  000146  170652        MOV     #146,@DCSR      ;TURN ON THE DISPLAY
1250  010154  004737  015172          1$:   JSR     PC,$STALL       ;THIS WAIT MAKES THE DISPLAY READABLE
1251  010160  017701  170646                MOV     @CDTP,R1        ;GET CURRENT DISPLAY POINTER
1252  010164  020127  010222                CMP     R1,#10$         ;IS THE DISPLAY POINTER OUT OF BOUNDS?
1253  010170  100003                        BPL     2$              ;IF SO, GO REPORT AN ERROR
1254  010172  005300                        DEC     R0              ;ITS OK, ADD 1 TO COUNT
1255  010174  001367                        BNE     1$              ;AND TRY AGAIN
1256  010176  000413                        BR      T0014           ;GO ON TO THE NEXT TEST
1257  010200  012705  024464          2$:   MOV     #MSG16A,R5      ;SETUP ADDRESS OF THE ERROR MESSAGE
1258  010204  004737  014532                JSR     PC,ERMES        ;REPORT THE ERROR
1259  010210  000406                        BR      T0014           ;GO ON TO THE NEXT TEST
1260  010212  003761                  9$:   3761
1261  010214  024611                        MSG17
1262  010216  003744                        3744
1263  010220  024457                        MSG16
1264  010222  100000                  10$:  100000
1265  010224  016074                        TBL22
1266
1267
1268                                  .SBTTL  TEST THAT DCSR BIT 6 CLEAR, DISSABLES THE END OF SCREEN CHARACTER
1269                                  ;T0014  A SPECIAL DISPLAY TABLE IS DISPLAYED FROM IN THIS TEST.
1270                                  ;       IT HAS IN IT A POINTER TO THE TEST # ASCII, A POINTER TO A TEXT
1271                                  ;       MESSAGE CONTAINING TWO EOS(0 & 200) CHARACTERS, AND A JUMP BACK TO THE
1272                                  ;       BEGINING OF THE DISPLAY TABLE. IF THE END OF SCREEN CHARACTERS
1273                                  ;       ARE TRUELY DISABLED, THE TEST # AND TEXT MESSAGE SHOULD
1274                                  ;       APPEAR ON THE SCREEN 24 TIMES, IF AN END OF SCREEN CHARACTER IS
1275                                  ;       RECOGNIZED, THE MESSAGE WILL APPEAR LESS THAN THAT.
1276                                  ;
1277  010226  000004                  T0014:  SCOPE
1278  010230  012737  010256  000606          MOV     #1$,$LPADR      ;LOOP BACK TO 1$ IF ERROR, AND SWR<13> IS CLEAR
1279  010236  004737  015102                JSR     PC,FXTST        ;CHANGE THE TEST # TO ASCII
1280  010242  012777  010276  170560        MOV     #9$,@IDTP       ;SETUP POINTER TO THE CUTE LITTLE DISPLAY TABLE
1281  010250  012777  000106  170550        MOV     #106,@DCSR      ;TURN ON THE DISPLAY, WITH SPECIAL CHARACTER BIT CLEAR
1282  010256  004737  015002          1$:   JSR     PC,LSTALL       ;LET THE OPERATOR SEE THE RESULTS
1283  010262  004737  015002                JSR     PC,LSTALL       ;ALLOW MORE TIME
1284  010266  004737  015002                JSR     PC,LSTALL       ;ALLOW MORE TIME
1285  010272  000137  010312                JMP     T0015           ;GO ON TO THE NEXT TEST
1286  010276  003761                  9$:   3761
1287  010300  024611                        MSG17
1288  010302  003675                        3675
1289  010304  030464                        MSG41
1290  010306  100000                  10$:  100000
1291  010310  010276                        9$
1292
1293
1294
1295
1296
1297                                  .SBTTL  TEST THAT A SET BIT 15 IN A PARAMETER CAUSES "JUMP"
1298                                  ;       TO A NEW PLACE IN THE DISPLAY TABLE
1299                                  ;T0015  A DISPLAY TABLE IS DISPLAYED FROM IN WHICH THERE IS A TEST #
1300                                  ;       TEXT POINTER, A DISPLAY TABLE JUMP INSTRUCTION, AND A MESSAGE
```

```
1301                                    ;       THAT SAYS THAT THE JUMP DID NOT WORK.
1302                                    ;       IF THE JUMP DOES NOT WORK, THE MESSAGE THAT SAYS IT WONT WILL BE DISPLAYED.
1303                                    ;       ALSO THE VALUE OF THE CDTP IS WATCHED, AND IF IT HAS A VALUE
1304                                    ;       THAT IT SHOULD NOT HAVE, THE JUMP IS ASSUMED TO BE AT FAULT,
1305                                    ;       AND A ERROR MESSAGE SAYING SO IS DISPLAYED
1306                                    ;
1307  010312  000004                   T0015:  SCOPE
1308  010314  004737  015102                   JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
1309  010320  012777  000146  170500           MOV     #146,@DCSR      ;ENSBLE SPECIAL CHARACTERS
1310  010326  012737  010346  000606           MOV     #1$,$LPADR      ;LOOP BACK TO 1$ IF ERROR, AND SWR<13> IS CLEAR
1311  010334  012777  010404  170466           MOV     #3$,@IDTP       ;NEW DISPLAY TABLE
1312  010342  012700  000100                   MOV     #100,R0         ;SETUP A COUNTER
1313  010346  004737  015172           1$:     JSR     PC,MSTALL       ;THIS WAIT MAKES THE DISPLAY READABLE
1314  010352  017701  170454                   MOV     @CDTP,R1        ;GET CURRENT DISPLAY TABLE POINTER
1315  010356  021127  010420                   CMP     @R1,#4$         ;IS THE DISPLAY POINTER OUT OF BOUNDS?
1316  010362  001403                           BEQ     2$              ;IF SO, GO REPORT AN ERROR
1317  010364  005300                           DEC     R0              ;ITS OK, ADD 1 TO COUNT
1318  010366  001367                           BNE     1$              ;AND TRY AGAIN
1319  010370  000415                           BR      T0016           ;GO ON TO THE NEXT TEST
1320  010372  012705  026067           2$:     MOV     #MSG27,R5       ;SETUP ADDRESS OF THE ERROR MESSAGE
1321  010376  004737  014532                   JSR     PC,ERMES        ;REPORT THE ERROR
1322  010402  000410                           BR      T0016           ;GO ON TO THE NEXT TEST
1323  010404  003761                   3$:     3761
1324  010406  024611                           MSG17
1325  010410  100000                           100000
1326  010412  010420                           4$
1327  010414  003710                           3710
1328  010416  026067                           MSG27
1329  010420  003772                   4$:     3772
1330  010422  024457                           MSG16
1331
1332
1333                                    .SBTTL  ZERO BLOCK COUNT TEST
1334                                    ;T0016  DISPLAY 80 CHAR LINES WITH A 0 CHAR LINE BETWEEN EVERY OTHER ONE
1335                                    ;       IF THE ZERO COUNT DOES NOT DISPLAY AS 0, THE BAR TO THE RIGHT
1336                                    ;       SIDE OF THE SCREEN WILL BE CROOKED OR BROKEN, INSTEAD OF STRAIGHT
1337                                    ;       AND SOLID DOWN THE ENTIRE LENGTH OF THE SCREEN.
1338                                    ;       ALSO THE FOLLOWING ERROR MESSAGE MAY APPEAR ON THE SCREEN...
1339                                    ;
1340                                    ;         ERROR  :  ZERO BLOCK COUNT NOT RECOGNIZED AS SUCH
1341                                    ;
1342  010424  000004                   T0016:  SCOPE
1343  010426  004737  015102                   JSR     PC,FXTST
1344  010432  012737  000001  000606           MOV     #1,$LPADR       ;LOOPBACK ADDRESS FOR THIS TEST
1345  010440  012777  000146  170360           MOV     #146,@DCSR      ;REGULAR DISPLAY
1346  010446  012777  010470  170354           MOV     #9$,@IDTP       ;SETUP DISPLAY TABLE
1347  010454  004737  015002           1$:     JSR     PC,LSTALL       ;LET IT BE SEEN
1348  010460  004737  015002                   JSR     PC,LSTALL       ;A LITTLE LONGER
1349  010464  000137  010514                   JMP     T0017
1350  010470  003761                   9$:     3761
1351  010472  024611                           MSG17
1352  010474  003657                   10$:    3657
1353  010476  030333                           MSG44
1354  010500  003657                           3657
1355  010502  030333                           MSG44
1356  010504  003777                           3777
```

```
1357  010506  076001                        MSG26
1358  010510  100000                        100000
1359  010512  010474                        10$
1360
1361
1362                              .SBTTL  DISPLAY FROM DIFFERENT MEMORY AREAS TEST
1363                              ;T0017  SIZE MEMORY AND DISPLAY FROM EACH 2K HUNK OF MEMORY
1364                              ;       A DISPLAY TABLE AND SOME TEXT IS PLACED IN EACH 2K HUNK OF
1365                              ;       MEMORY, AND USED TO DISPLAY WITH.
1366  010514  000004              T0017:  SCOPE
1367  010516  004737  015102              JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
1368  010522  012706  000600              MOV     #600,SP         ;NEED ROOM ON STACK, SO RESET IT
1369  010526  004737  014136              JSR     PC,CLRTUB       ;CLEAR OUT DISPLAY TABLE
1370  010532  012777  010762  170270      MOV     #T0017C,@IDTP   ;SETUP THE POINTER TO THE DISPLAY TABLE
1371  010540  012777  000146  170260      MOV     #146,@DCSR      ;TURN ON THE DISPLAY
1372  010546  005000                      CLR     R0              ;SET MEMORY COUNT TO 0
1373  010550  012737  010612  000004      MOV     #3$,@#4         ;SETUP VECTOR IN CASE OF NON EXISTANT MEMORY
1374  010556  012737  000140  000006      MOV     #140,@#6        ;SETUP TRAP VECTOR
1375  010564  010001              1$:     MOV     R0,R1           ;SETUP WORKING ADDRESS
1376  010566  062700  010000              ADD     #10000,R0       ;BUMP REAL ADDRESS TO BEGINING OF NEXT 2K HUNK
1377  010572  005721              2$:     TST     (R1)+           ;FIND OUT IF THE MEMORY IS THERE
1378  010574  020100                      CMP     R1,R0           ;IT IS, IS ALL OF 2K THERE?
1379  010576  001375                      BNE     2$              ;DONT KNOW YET, GO BACK AND KEEP TESTING
1380  010600  010046                      MOV     R0,-(SP)        ;ITS ALL THERE, PUT ADDRESS ON THE STACK
1381  010602  162716  010000              SUB     #10000,(SP)     ;MAKE IT THE BEGINING ADDRESS
1382  010606  162706  000004              SUB     #4,SP           ;ANTICIPATING WHAT WILL HAPPEN NEXT
1383  010612  062706  000004      3$:     ADD     #4,SP           ;KEEP THE STACK WHERE IT BELONGS
1384  010616  020027  160000              CMP     R0,#160000      ;REACHED END OF POSIBLE MEMORY?
1385  010622  001360                      BNE     1$              ;IF NOT GO BACK AND CHECK ANOTHER HUNK
1386  010624  004737  015002              JSR     PC,LSTALL       ;LEAVE TEST # ON THE SCREEN FOR A MOMENT
1387                              ;WE NOW HAVE A BUNCH OF ADDRESSES ON THE STACK
1388  010630  012600              T0017A: MOV     (SP)+,R0        ;GET AN ADDRESS
1389  010632  001002                      BNE     1$              ;CONTINUE ON ANY ADDR BUT 0
1390  010634  000137  010772              JMP     T0020           ;NEED NOT TRY 1ST 2K HUNK, GO TO NEXT TEST
1391  010640  020027  030001      1$:     CMP     R0,#30001       ;IS THE ADDRESS IN THE 1ST 6K?
1392  010644  103771                      BLO     T0017A          ;IF SO GET ANOTHER
1393  010646  020027  040000              CMP     R0,#40000       ;IS IT IN THE 2ND 4K?
1394  010652  103000                      BHIS    2$              ;IF NOT WE GO DISPLAY FROM THAT AREA
1395                                                              ;NOW TRY TO DISPLAY FROM IT
1396  010654  010037  000766      2$:     MOV     R0,TEMP         ;SETUP HUNK VALUE
1397  010660  012705  026523              MOV     #MSG33T,R5      ;SO THAT IT CAN BE
1398  010664  004737  015504              JSR     PC,BIOCT        ;CONVERTED TO ASCII
1399  010670  010037  000766              MOV     R0,TEMP
1400  010674  062737  000004  000766      ADD     #4,TEMP         ;ASCII TEXT STARTS AT 4TH CHARACTER
1401  010702  013737  000766  026470      MOV     TEMP,@#STB1     ;IN THE HUNK, SETUP DISPLAY TABLE ENTRY CORRECTLY
1402  010710  012705  026555              MOV     #MSG33A,R5      ;ADDRESS TO PUT ASCII VALUE
1403  010714  004737  015504              JSR     PC,BIOCT        ;OF THE ASCII TEXT ADDRESS
1404  010720  012701  026466              MOV     #MSTB0,R1       ;ADDRESS OF STUFF TO MOVE INTO THE HUNK
1405  010724  012702  000037              MOV     #37,R2          ;SETUP COUNT FOR # OF WORDS TO MOVE
1406  010730  012120              4$:     MOV     (R1)+,(R0)+     ;MOVE A WORD INTO THE HUNK
1407  010732  005302                      DEC     R2              ;IS IT THE LAST WORD?
1408  010734  001375                      BNE     4$              ;IF NOT, GO BACK AND DO ANOTHER
1409  010736  162737  000004  026470      SUB     #4,@#STB1       ;BACK UP POINTER TO THE DISPLAY TABLE ADDRESS
1410  010744  013777  026470  170056      MOV     @#STB1,@IDTP    ;IF DONE, START DISPLAYING STUFF FROM THE HUNK
1411  010752  004737  015002              JSR     PC,LSTALL       ;AND ALLOW TIME FOR THE MESSAGE TO BE SEEN
1412  010756  000137  010630              JMP     T0017A          ;GO TRY DISPLAYING FROM ANOTHER AREA
```

```
1413
1414  010762  003761                    T0017C: 3761
1415  010764  024611                            MSG17
1416  010766  003776                            3776
1417  010770  030064                            MSG41
1418
1419
1420
1421
1422                                    .SBTTL  MAKE SURE THAT THE CHARACTER GENERATOR CAN BE LOADED WHILE DISPLAY IS GOING
1423                                    ;T0020  A TEXT MESSAGE IS PUT ONTO THE SCREEN AND THE DISPLAY IS KEPT RUNNING,
1424                                    ;       THEN CHARACTER CODE 41 IS LOADED WITH A CROSS PATTERN,
1425                                    ;       SINCE THERE ARE A LOT OF 041 CODES IN THE MESSAGE BEING DISPLAYED
1426                                    ;       THE OPERATOR SHOULD BE  ABLE TO OBSERVE THE CHANGE. CHARACTER CODE 41 IS
1427                                    ;       THEN LOADED WITH A TRIANGLE PATTERN, AND AGAIN THE CHANGE SHOULD BE
1428                                    ;       VISABLE, THEN CHARACTER CODE 41 IS LOADED WITH ALTERNATING TRIANGLES,
1429                                    ;       THEN CROSSES, 19 MORE TIMES
1430                                    ;
1431  010772  000004                    T0020:  SCOPE
1432  010774  004737  015102                    JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
1433  011000  012737  011020  000606            MOV     #1$,@LPADR      ;SETUP THE ERROR LOOPBACK ADDRESS
1434  011006  012706  000600                    MOV     #600,SP         ;INIT THE STACK
1435  011012  012777  011144  170010            MOV     #9$,@IDTP       ;DISPLAY SPECIAL MESSAGE
1436  011020  012703  000024            1$:     MOV     #20.,R3         ;SETUP COUNT FOR 20 CHANGES
1437  011024  012777  022342  170002    2$:     MOV     #PLUS5,@DCP     ;ADDRESS OF THE "+" FOR CODE 21
1438  011032  012777  100146  167766            MOV     #100146,@DCSR   ;LOAD CHARACTER SET
1439  011040  012705  022412            3$:     MOV     #ENDPLS,R5      ;SETUP ADDRESS OF CHAR SET ENDING FOR COMPARE
1440  011044  004737  015254                    JSR     PC,TSTLOD       ;WAIT FOR LOAD TO FINISH
1441  011050  004737  015172                    JSR     PC,MSTALL       ;TRANSITIONS MUST BE VIEWABLE
1442  011054  004737  015172                    JSR     PC,MSTALL       ;TRANSITIONS MUST BE VIEWABLE
1443  011060  004737  015172                    JSR     PC,MSTALL       ;TRANSITIONS MUST BE VIEWABLE
1444  011064  004737  015172                    JSR     PC,MSTALL       ;TRANSITIONS MUST BE VIEWABLE
1445  011070  012777  022270  167736            MOV     #FAKE,@DCP      ;SETUP ADDRESS OF TRIANGLE CHARACTER FOR CODE 41
1446  011076  012705  022340                    MOV     #ENDCHR,R5      ;SETUP ADDRESS OF CHAR SET ENDING
1447  011102  012777  100146  167716            MOV     #100146,@DCSR   ;LOAD CHARACTER SET
1448  011110  004737  015254                    JSR     PC,TSTLOD       ;WAIT FOR CHAR SET LOAD TO BE DONE
1449  011114  004737  015172                    JSR     PC,MSTALL       ;TRANSITIONS MUST BE VIEWABLE
1450  011120  004737  015172                    JSR     PC,MSTALL       ;TRANSITIONS MUST BE VIEWABLE
1451  011124  004737  015172                    JSR     PC,MSTALL       ;TRANSITIONS MUST BE VIEWABLE
1452  011130  004737  015172                    JSR     PC,MSTALL       ;TRANSITIONS MUST BE VIEWABLE
1453  011134  005303                            DEC     R3              ;DONE 20 CHANGES YET?
1454  011136  001332                            BNE     2$              ;IF NOT GO BACK AND DO MORE
1455  011140  000137  011160                    JMP     T0021           ;GO ON TO THE NEXT TEST
1456  011144  003761                    9$:     3761
1457  011146  024611                            MSG17
1458  011150  003660                            3660
1459  011152  024157                            MSG14
1460  011154  003771                            3771
1461  011156  024457                            MSG16
1462
1463
1464                                    .SBTTL  TEST UNDERLINE MODE
1465                                    ;T0021  THIS TEST DISPLAYS THE WORDS "UNDERLINE MODE", IN UNDERLINE MODE,
1466                                    ;
1467  011160  000004                    T0021:  SCOPE
1468  011162  004737  015102                    JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
```

```
1469  011166  012737  011262  000606          MOV     #28,$LPADR      ;LOOP BACK TO 28 IF ERROR, AND SWR<13> IS CLEAR
1470  011174  004737  014136                  JSR     PC,CLRTUB       ;CLEAN OUT THE DISPLAY TABLE
1471  011200  012777  015760  167622          MOV     #DISTBL,@IDTP   ;SETUP THE POINTER TO THE DISPLAY TABLE
1472  011206  012777  000146  167612          MOV     #146,@DCSR      ;TURN ON THE DISPLAY
1473  011214  012705  024611                  MOV     #MSG17,R5       ;GET ADDR OF TEST # MESSAGE
1474  011220  004737  013750                  JSR     PC,TUBOUT       ;DISPLAY THE TEST #
1475  011224  012700  000013                  MOV     #13,R0          ;COUNT TO FILL SCREEN
1476  011230  012705  024632          18:     MOV     #MSG18,R5       ;SETUP ADDRESS OF THE MESSAGE
1477  011234  012737  010000  000754          MOV     #10000,SPMODE   ;AND ITS MODE BITS
1478  011242  004737  013750                  JSR     PC,TUBOUT       ;SEND MESSAGE TO THE SCREEN
1479  011246  012705  024611                  MOV     #MSG17,R5       ;GET ADDR OF TEST # MESSAGE
1480  011252  004737  013750                  JSR     PC,TUBOUT       ;DISPLAY THE TEST #
1481  011256  005300                          DEC     R0              ;COUNT 1 MESSAGE MORE
1482  011260  001363                          BNE     18              ;FILLED SCREEN? GO BACK IF NOT.
1483  011262  004737  015002          28:     JSR     PC,LSTALL       ;WAIT, SO MESSAGE STAYS ON SCREEN LONG ENOUGH TO SEE
1484
1485
1486                                  .SBTTL  TEST REGULAR MODE
1487                                  ;T0022  THIS TEST DISPLAYS THE WORDS "REGULAR MODE" IN REGULAR MODE
1488  011266  000004                  T0022:  SCOPE
1489  011270  012737  011356  000606          MOV     #28,$LPADR      ;LOOP BACK TO 28 IF ERROR, AND SWR<13> IS CLEAR
1490  011276  004737  015102                  JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
1491  011302  012777  000146  167516          MOV     #146,@DCSR      ;TURN ON THE DISPLAY
1492  011310  012705  024611                  MOV     #MSG17,R5       ;GET ADDR OF TEST # MESSAGE
1493  011314  004737  013750                  JSR     PC,TUBOUT       ;DISPLAY THE TEST #
1494  011320  012700  000013                  MOV     #13,R0          ;COUNT TO FILL SCREEN
1495  011324  012705  026403          18:     MOV     #MSG31,R5       ;SETUP ADDRESS OF THE MESSAGE
1496  011330  012737  000000  000754          MOV     #0000,SPMODE    ;AND ITS MODE BITS
1497  011336  004737  013750                  JSR     PC,TUBOUT       ;SEND MESSAGE TO THE SCREEN
1498  011342  012705  024611                  MOV     #MSG17,R5       ;GET ADDR OF TEST # MESSAGE
1499  011346  004737  013750                  JSR     PC,TUBOUT       ;DISPLAY THE TEST #
1500  011352  005300                          DEC     R0              ;COUNT 1 MESSAGE MORE
1501  011354  001363                          BNE     18              ;FILLED SCREEN? GO BACK IF NOT.
1502  011356  004737  015002          28:     JSR     PC,LSTALL       ;WAIT, SO MESSAGE STAYS ON SCREEN LONG ENOUGH TO SEE
1503
1504
1505                                  .SBTTL  TEST REVERSE VIDEO MODE
1506                                  ;T0023  THIS TEST DISPLAYS THE WORDS "REVERSE VIDEO MODE" IN REVERSE
1507                                  ;       VIDEO MODE
1508  011362  000004                  T0023:  SCOPE
1509  011364  004737  015102                  JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
1510  011370  012737  011452  000606          MOV     #28,$LPADR      ;LOOP BACK TO 28 IF ERROR, AND SWR<13> IS CLEAR
1511  011376  012777  000146  167422          MOV     #146,@DCSR      ;TURN ON THE DISPLAY
1512  011404  012705  024611                  MOV     #MSG17,R5       ;GET ADDR OF TEST # MESSAGE
1513  011410  004737  013750                  JSR     PC,TUBOUT       ;DISPLAY THE TEST #
1514  011414  012700  000013                  MOV     #13,R0          ;COUNT TO FILL SCREEN
1515  011420  012705  024707          18:     MOV     #MSG21,R5       ;SETUP ADDRESS OF THE MESSAGE
1516  011424  012737  004000  000754          MOV     #4000,SPMODE    ;AND ITS MODE BITS
1517  011432  004737  013750                  JSR     PC,TUBOUT       ;SEND MESSAGE TO THE SCREEN
1518  011436  012705  024611                  MOV     #MSG17,R5       ;GET ADDR OF TEST # MESSAGE
1519  011442  004737  013750                  JSR     PC,TUBOUT       ;DISPLAY THE TEST #
1520  011446  005300                          DEC     R0              ;COUNT 1 MESSAGE MORE
1521  011450  001363                          BNE     18              ;FILLED SCREEN? GO BACK IF NOT.
1522  011452  004737  015002          28:     JSR     PC,LSTALL       ;WAIT, SO MESSAGE STAYS ON SCREEN LONG ENOUGH TO SEE
1523
1524
```

C3

```
                                        .SBTTL  TEST BOLD MODE
                                        ;T0024  THIS TEST DISPLAYS THE WORDS "BOLD MODE" IN BOLD MODE
011456  000004                          T0024:  SCOPE
011460  004737  015102                          JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
011464  012737  011546  000606                  MOV     #28,$LPADR      ;LOOP BACK TO 28 IF ERROR, AND SWR<13> IS CLEAR
011472  012777  000146  167326                  MOV     #146,@DCSR      ;TURN ON THE DISPLAY
011500  012705  024611                          MOV     #MSG17,R5       ;GET ADDR OF TEST # MESSAGE
011504  004737  013750                          JSR     PC,TUROUT       ;DISPLAY THE TEST #
011510  012700  000013                          MOV     #13,R0          ;COUNT TO FILL SCREEN
011514  012705  024652                  18:     MOV     #MSG19,R5       ;SETUP ADDRESS OF THE MESSAGE
011520  012737  020000  000754                  MOV     #20000,SPMODE   ;AND ITS MODE BITS
011526  004737  013750                          JSR     PC,TUROUT       ;SEND MESSAGE TO THE SCREEN
011532  012705  024611                          MOV     #MSG17,R5       ;GET ADDR OF TEST # MESSAGE
011536  004737  013750                          JSR     PC,TUROUT       ;DISPLAY THE TEST #
011542  005300                                  DEC     R0              ;COUNT 1 MESSAGE MORE
011544  001363                                  BNE     18              ;FILLED SCREEN? GO BACK IF NOT.
011546  004737  015002                  28:     JSR     PC,LSTALL       ;WAIT, SO MESSAGE STAYS ON SCREEN LONG ENOUGH TO SEE

                                        .SBTTL  TEST BLANK MODE
                                        ;T0025  THIS TEST DISPLAYS THE WORDS "BLANK MODE ERROR" IN BLANKING MODE. (YOU
                                        ;       SHOULD NOT SEE THE WORDS "BLANK MODE ERROR".)
011552  000004                          T0025:  SCOPE
011554  004737  015102                          JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
011560  012737  011642  000606                  MOV     #28,$LPADR      ;LOOP BACK TO 28 IF ERROR, AND SWR<13> IS CLEAR
011566  012777  000146  167232                  MOV     #146,@DCSR      ;TURN ON THE DISPLAY
011574  012705  024611                          MOV     #MSG17,R5       ;GET ADDR OF TEST # MESSAGE
011600  004737  013750                          JSR     PC,TUROUT       ;DISPLAY THE TEST #
011604  012700  000013                          MOV     #13,R0          ;COUNT TO FILL SCREEN
011610  012705  024665                  18:     MOV     #MSG20,R5       ;SETUP ADDRESS OF THE MESSAGE
011614  012737  040000  000754                  MOV     #40000,SPMODE   ;AND ITS MODE BITS
011622  004737  013750                          JSR     PC,TUROUT       ;SEND MESSAGE TO THE SCREEN
011626  012705  024611                          MOV     #MSG17,R5       ;GET ADDR OF TEST # MESSAGE
011632  004737  013750                          JSR     PC,TUROUT       ;DISPLAY THE TEST #
011636  005300                                  DEC     R0              ;COUNT 1 MESSAGE MORE
011640  001363                                  BNE     18              ;FILLED SCREEN? GO BACK IF NOT.
011642  004737  015002                  28:     JSR     PC,LSTALL       ;WAIT, SO MESSAGE STAYS ON SCREEN LONG ENOUGH TO SEE

                                        .SBTTL  TEST PANNING UPWARDS(FAST)
                                        ;T0026  THIS IS A TEST OF THE PAN OFFSET BITS. PANNING UPWARDS IS DONE AT A
                                        ;       DONE AT A REASONABLY FAST RATE. THE PANNING SHOULD LOOK SMOOTH.
                                        ;       IF IT DOESNT, THAT INDICATES THAT THE PAN OFFSET BITS ARE NOT ALL
                                        ;       WORKING PROPERLY.
011646  000004                          T0026:  SCOPE
011650  004737  015102                          JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
011654  012737  011662  000606                  MOV     #18,$LPADR      ;SETUP THE ERROR LOOPBACK ADDRESS
011662  012737  000140  000766          18:     MOV     #140,TEMP
011670  012706  000600                          MOV     #600,SP         ;INIT THE STACK
011674  012777  012056  167126                  MOV     #68,@TDTP       ;DISPLAY THE PAN MESSAGE
011702  012700  000145                          MOV     #145,R0         ;PAN REGISTERS FIRST VALUE-1
011706  005200                          28:     INC     R0              ;PAN FURTHER
011710  010077  167112                          MOV     R0,@DCSR        ;SET PAN BITS
011714  106427  000000                          MTPS    #0              ;MAKE SURE INTERRUPTS ARE ALLOWED
011720  005037  000772                          CLR     STLCNT          ;ZERO OUT A COUNTER LOCATION
011724  005037  001050                          CLR     INTCNT          ;ZERO INTERRUPT SWITCH
```

```
1581  011730  023737  000762  001050  38:   CMP     UPFAST,INTCNT   ;ENOUGH DISPLAY INTERRUPTS YET?
1582  011736  001420                        BEQ     4$              ;IF SO GO SERVICE
1583  011740  005237  000772                INC     STLCNT          ;IF NOT, IS ONE LONG OVERDUE?
1584  011744  001371                        BNE     3$              ;IF NOT WAIT MORE
1585  011746  004737  014136                JSR     PC,CLRTUB       ;OVERDUE, CLEAR OUT THE REGULAR DISPLAY TABLE
1586  011752  012777  015760  167050        MOV     #DISTBL,@IDTP   ;SET TO DISPLAY FROM IT
1587  011760  004737  015102                JSR     PC,FXTST        ;DISPLAY TEST #
1588  011764  012705  026161                MOV     #MSG20,R5       ;SETUP THE ADDRESS OF THE ERROR MESSAGE
1589  011770  004737  014532                JSR     PC,ERMES        ;DISPLAY THE ERROR MESSAGE
1590  011774  000137  012102                JMP     T0027           ;AND GO ON TO THE NEXT TEST
1591  012000  020027  000157        48:     CMP     R0,#157         ;GONE ALL THE WAY YET?
1592  012004  001340                        BNE     2$              ;IF NOT GO BACK AND PAN SOME MORE
1593  012006  062777  000004  167014        ADD     #4,@IDTP        ;TAKE ONE LINE FROM THE TOP OF THE DISPLAY LIST
1594  012014  012777  000146  167004        MOV     #146,@DCSP      ;SET PAN TO 0
1595  012022  027727  167002  012066        CMP     @IDTP,#7$       ;DISPLAY SAME AS INITIAL?
1596  012030  001003                        BNE     5$              ;IF NOT LEAVE IT ALONE
1597  012032  012777  012056  166770        MOV     #6$,@IDTP       ;IF SO RESET IT SO WE DONT RUN OUT OF ROOM
1598  012040  012700  000145        5$:     MOV     #145,R0
1599  012044  005337  000766                DEC     TEMP
1600  012050  001316                        BNE     2$
1601  012052  000137  012102                JMP     T0027           ;GO ON TO THE NEXT TEST(PAN DOWN)
1602  012056  003660          6$:     3660
1603  012060  024021                  MSG12
1604  012062  003761                  3761
1605  012064  024611                  MSG17
1606  012066  003660          7$:     3660
1607  012070  024021                  MSG12
1608  012072  003761                  3761
1609  012074  024611                  MSG17
1610  012076  100000                  100000
1611  012100  012056          8$:     6$
1612
1613
1614                          .SBTTL  TEST PANNING DOWN (FAST)
1615                          ;T0027  THIS IS A TEST OF THE PAN OFFSET BITS, PANNING DOWNWARDS IS
1616                          ;       DONE AT A REASONABLY FAST RATE, THE PANNING SHOULD LOOK SMOOTH,
1617                          ;       IF IT DOESNT, THAT INDICATES THAT THE BITS ARE NOT WORKING PROPERLY.
1618  012102  000004          T0027:  SCOPE
1619  012104  004737  015102                JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
1620  012110  012737  012116  000606        MOV     #1$,@LPADR      ;SETUP THE ERROR LOOPBACK ADDRESS
1621  012116  012737  000140  000766  1$:    MOV     #140,TEMP
1622  012124  012706  000600                MOV     #600,SP         ;INIT THE STACK
1623  012130  012777  012330  166672        MOV     #9$,@IDTP       ;DISPLAY THE PAN MESSAGE
1624  012136  012700  000160                MOV     #160,R0         ;PAN REGISTERS FIRST VALUE-1
1625  012142  005300          2$:     DEC     R0              ;PAN FURTHER
1626  012144  010077  166656                MOV     R0,@DCSP        ;SET PAN BITS
1627  012150  106427  000000                MTPS    #0              ;MAKE SURE INTERRUPTS ARE ALLOWED
1628  012154  005037  000772                CLR     STLCNT          ;ZERO OUT A COUNTER LOCATION
1629  012160  005037  001050                CLR     INTCNT          ;ZERO INTERRUPT SWITCH
1630  012164  023737  000764  001050  3$:    CMP     DOWNFA,INTCNT   ;GOT ENOUGH DISPLAY INTERRUPTS YET?
1631  012172  001420                        BEQ     4$              ;IF SO GO SERVICE
1632  012174  005237  000772                INC     STLCNT          ;IF NOT, IS ONE LONG OVERUE?
1633  012200  001371                        BNE     3$              ;IF NOT WAIT MORE
1634  012202  004737  014136                JSR     PC,CLRTUB       ;OVERDUE, CLEAR OUT THE REGULAR DISPLAY TABLE
1635  012206  012777  015760  166614        MOV     #DISTBL,@IDTP   ;SET TO DISPLAY FROM IT
1636  012214  004737  015102                JSR     PC,FXTST        ;DISPLAY TEST #
```

```
1637  012220  012705  026161              MOV     #MSG28,R5       ;SETUP THE ADDRESS OF THE ERROR MESSAGE
1638  012224  004737  014532              JSR     PC,ERMES        ;DISPLAY THE ERROR MESSAGE
1639  012230  000137  012350              JMP     T0030           ;AND GO ON TO THE NEXT TEST
1640  012234  020027  000146      48:     CMP     R0,#146         ;GONE ALL THE WAY YET?
1641  012240  001343                      BNE     2$              ;IF NOT GO BACK AND PAN SOME MORE
1642  012242  162777  000004  166560      SUB     #4,@IDTP        ;ADD ONE LINE TO THE TOP OF THE DISPLAY LIST
1643  012250  012777  000157  166550      MOV     #157,@DCSR      ;SET PAN TO 0
1644  012256  027727  166546  012324      CMP     @IDTP,#6$       ;DISPLAY SAME AS INITIAL?
1645  012264  001003                      BNE     5$              ;IF NOT LEAVE IT ALONE
1646  012266  012777  012334  166534      MOV     #7$,@IDTP       ;IF SO RESET IT SO WE DONT RUN OUT OF ROOM
1647  012274  012700  000160      5$:     MOV     #160,R0
1648  012300  005337  000766              DEC     TEMP
1649  012304  001316                      BNE     2$
1650  012306  004737  014136              JSR     PC,CLRTUB       ;CLEAR OUT THE REGULAR DISPLAY TABLE
1651  012312  012777  015760  166510      MOV     #DISTRL,@IDTP   ;DISPLAY FROM IT
1652  012320  000137  012350              JMP     T0030           ;GO ON TO THE NEXT TEST
1653  012324  003660              6$:     3660
1654  012326  024021                      MSG12
1655  012330  003761              9$:     3761
1656  012332  024611                      MSG17
1657  012334  003660              7$:     3660
1658  012336  024021                      MSG12
1659  012340  003761                      3761
1660  012342  024611                      MSG17
1661  012344  100000                      100000
1662  012346  012324              8$:     6$
1663
1664
1665
1666
1667                          .SBTTL  PUT A GRID ON THE SCREEN TO CHECK FOR PROPPER ALIGNMENT
1668                          ;T0030  PUT A GRID ON THE SCREEN TO CHECK FOR PROPPER
1669                          ;       ALIGNMENT AND LINIERITY
1670  012350  000004          T0030;  SCOPE
1671  012352  004737  015102              JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
1672  012356  012737  012422  000606      MOV     #1$,$LPADR      ;IF ERROR, AND SWR<13>=0, THEN LOOP BACK TO 1$
1673  012364  012777  022342  166442      MOV     #PLUSS,@DCP     ;ADDRESS OF THE "+" FOR CODE 21
1674  012372  012777  100146  166426      MOV     #100146,@DCSR   ;LOAD CHARACTER SET
1675  012400  012705  022412              MOV     #ENDPLS,R5      ;SETUP END OF CHAR SET ADDRESS
1676  012404  004737  015254              JSR     PC,TSTLOD       ;WAIT FOR LOADING TO BE DONE
1677  012410  004737  015002              JSR     PC,LSTALL       ;WAIT, SO RESULTS CAN BE SEEN
1678  012414  012777  012432  166406      MOV     #3$,@IDTP       ;DISPLAY GRID
1679  012422  004737  015002      1$:     JSR     PC,LSTALL       ;AND WAIT SO OPERATOR CAN SEE IT
1680  012426  000137  012442              JMP     T0031
1681  012432  003660              3$:     3660
1682  012434  026570                      MSG34
1683  012436  100000                      100000
1684  012440  012432                      3$
1685
1686
1687                          .SBTTL  PUT ANOTHER GRID ON THE SCREEN TO CHECK FOR PROPPER ALIGNMENT
1688                          ;T0031  THIS IS ANOTHER GRID EXCEPT WITH LARGER BOXES
1689                          ;       IN THIS TEST A GRID IS DISPLAYED. ALL OF THE BOXES
1690                          ;       SHOULD HAVE APPROXAMATLY THE SAME SIZE AND SHOULD HAVE
1691                          ;       REASONABLY STRAIGHT SIDES
1692  012442  000004          T0031:  SCOPE
```

```
1693  012444  004737  015102                  JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
1694  012450  012737  012514  000606          MOV     #1$,$LPADR      ;IF ERROR, AND SWR<13>=0, THEN LOOP BACK TO 1$
1695  012456  012777  022342  166350          MOV     #PLUSS,@DCP     ;ADDRESS OF THE "+" FOR CODE 21
1696  012464  012777  100146  166334          MOV     #100146,@DCSR   ;LOAD CHARACTER SET
1697  012472  012705  022412                  MOV     #ENDPIS,R5      ;SETUP END OF CHAR SET ADDRESS
1698  012476  004737  015254                  JSR     PC,TSTLOD       ;WAIT FOR LOADING TO BE DONE
1699  012502  004737  015002                  JSR     PC,LSTALL       ;WAIT, SO RESULTS CAN BE SEEN
1700  012506  012777  012524  166314          MOV     #3$,@IDTP       ;DISPLAY GRID
1701  012514  004737  015002          1$:     JSR     PC,ISTALL       ;AND WAIT SO OPERATOR CAN SEE IT
1702  012520  000137  012554                  JMP     T0032
1703  012524  003657                  3$:     3657
1704  012526  026570                          MSG34
1705  012530  003657                          3657
1706  012532  030206                          MSG43
1707  012534  003657                          3657
1708  012536  030206                          MSG43
1709  012540  003657                          3657
1710  012542  030206                          MSG43
1711  012544  003657                          3657
1712  012546  030206                          MSG43
1713  012550  100000                          100000
1714  012552  012524                          3$
1715
1716
1717
1718                                  .SBTTL  END OF PASS INDICATING
1719  012554  000004                  T0032:  SCOPE
1720  012556  004737  015102                  JSR     PC,FXTST        ;CHANGE TEST # TO ASCII IN CASE IT IS NEEDED
1721  012562  012777  016100  166240          MOV     #TBL23,@IDTP    ;NEW POINTER TO NOTHING
1722  012570  004737  015172                  JSR     PC,MSTALL       ;WAIT FOR A NOTHING DISPLAY
1723  012574  012777  000046  166224          MOV     #46,@DCSR       ;TURN OFF THE DISPLAY
1724  012602  004737  015172                  JSR     PC,MSTALL       ;NO GLITCHES
1725  012606  000005                          RESET                   ;TURN EVERYTHING OFF
1726  012610  013737  000612  000766          MOV     $ERTTL,TEMP     ;GET ERROR TOTAL FOR ALL PASSES
1727  012616  012705  025745                  MOV     #MSG25Y,R5      ;POINTER TO ASCII
1728  012622  004737  015504                  JSR     PC,BIOCT        ;CONVERT TOTAL # TO ASCII
1729  012626  013737  001056  000766          MOV     ERRPAS,TEMP     ;GET # OF ERRORS FOR THIS PASS
1730  012634  012705  025706                  MOV     #MSG25,R5       ;SETUP ADDRESS OF 6 BYTES
1731  012640  004737  015504                  JSR     PC,BIOCT        ;CHANGE # OF ERRORS TO ASCII
1732  012644  012705  025734                  MOV     #MSX25,R5       ;SETUP ADDR TO PUT ASCII OF PASS COUNT
1733  012650  013737  000600  000766          MOV     $PASS,TEMP      ;GET PASS COUNT #
1734  012656  004737  015504                  JSR     PC,BIOCT        ;CONVERT IT INTO ASCII
1735  012662  012777  013012  166140          MOV     #EOPTBL,@IDTP   ;SETUP POINTER TO DISPLAY TABLE
1736  012670  012777  000146  166130          MOV     #146,@DCSR      ;TURN ON THE DISPLAY
1737  012676  012705  025706                  MOV     #MSG25,R5       ;GET ADDR OF END OF PASS MESSAGE
1738  012702  004737  014722                  JSR     PC,TTYOUT       ;DISPLAY IT
1739  012706  012777  016110  166120          MOV     #CHARS,@DCP     ;GET THE ADDRESS OF THE CHARACTER SET
1740  012714  012777  100146  166104          MOV     #100146,@DCSR   ;START LOADING IT
1741  012722  004737  015002                  JSR     PC,LSTALL       ;GIVE IT TIME TO LOAD
1742
1743
1744                                  .SBTTL  END OF PASS ROUTINE
1745
1746                                  ;;**********************************************************************
1747                                  ;*INCREMENT THE PASS NUMBER ($PASS)
1748                                  ;*IF THERES A MONITOR GO TO IT
```

```
1749                                        ;*IF THERE ISN'T JUMP TO T0000
1750
1751  012726                                $EOP:
1752  012726  000004                               SCOPE
1753  012730  005037  000602                       CLR     $TSTNM          ;;ZERO THE TEST NUMBER
1754  012734  005037  000722                       CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
1755  012740  005237  000600                       INC     $PASS           ;;INCREMENT THE PASS NUMBER
1756  012744  042737  100000  000600               BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
1757  012752  005327                               DEC     (PC)+           ;;LOOP?
1758  012754  000001                $EOPCT: .WORD   1
1759  012756  003013                               BGT     $DOAGN          ;;YES
1760  012760  012737                               MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
1761  012762  000001                $ENDCT: .WORD   1
1762  012764  012754                               $EOPCT
1763  012766  013700  000042       $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
1764  012772  001405                               BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
1765  012774  000005                               RESET                   ;;CLEAR THE WORLD
1766  012776  004710                $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
1767  013000  000240                               NOP                     ;;SAVE ROOM
1768  013002  000240                               NOP                     ;;FOR
1769  013004  000240                               NOP                     ;;ACT11
1770  013006                                $DOAGN:
1771  013006  000137                               JMP     @(PC)+          ;;RETURN
1772  013010  001514                $RTNAD: .WORD   T0000
1773  013012  003700                $EOPTBL: 3700
1774  013014  025726                               MSG25
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
```

```
                                        SPECIAL TEST FOR SCOPE LOOPING

                                        .SBTTL  SPECIAL TEST FOR SCOPE LOOPING
                                        ;T0036  WRITE A CHARACTER, IF "XDISP" EQUALS 0(DEFAULT), WRITE
                                        ;       THE CHARACTER OVER AND OVER AND OVER, DO NOTHING BUT WAIT FOR IT TO BE  FINISHED
                                        ;       IF "XDISP" IS NOT EQUAL TO 0, THE CHARACTER WILL BE DISPLAYED AFTER IT IS LOADED
                                        ;       BEFORE STARTING SET LOCATION "XCODE" TO THE VALUE OF THE CODE YOU WISH TO WRITE
                                        ;       THE DEFAULT IS "A"(CODE 101)
013016                                  T0036:
013016  106427  000340                 XSTART: MTPS    #340                    ;NO INTERRUPTS
013022  012706  001000                         MOV     #1000,SP                ;SETUP STACK POINTER
013026  004737  013654                         JSR     PC,GITCOD               ;GET ADDRESS OF STUFF TO LOAD

                                        ;HERE IS THE ACTUAL WRITE LOOP
013032  012777  013602  165774         1$:     MOV     #SINCHR,@DCP            ;SETUP ADDRESS OF CHARACTER TO LOAD
013040  012777  100006  165760                 MOV     #100006,@DCSR           ;SET LOAD BIT IN CONTROL REG
013046  027727  165762  013652         2$:     CMP     @DCP,#ENDSIN            ;POINTER AT END OF CHAR DESCRIBING WORDS YET?
013054  001374                                 BNE     2$                      ;IF NOT, WAIT TILL IT IS
013056  005737  013112                         TST     XDISP                   ;SHOULD WE NOW DISPLAY THE CHARACTER?
013062  001763                                 BEQ     1$                      ;NOT IF THE "XDISP" SWITCH IS 0

013064  012777  013104  165736                 MOV     #XDSPTR,@IDTP           ;POINT POINTER AT THE DISPLAY TABLE
013072  012777  000146  165726                 MOV     #146,@DCSR              ;START UP THE DISPLAY

013100  000137  013100                 3$:     JMP     3$                      ;JUST DISPLAY, DO NOTHING ELSE
                                                                               ;WE ARE NOW DISPLAYING THE SPECIFIED CHARACTER,
                                                                               ;PLUS AN END OF SCREEN CHARACTER


013104  003775                         XDSPTR: 3775                            ;2 CHARACTERS
013106  013110                                 XCODE                           ;POINTS AT THE SPECIFIED CODE

                                        ;USER SETTABLE SWITCHES
013110  000101                         XCODE:  000101                          ;CODE # TO WRITE.(DEFAULT "A")
013112  000000                         XDISP:  000000                          ;SET THIS TO A 1 TO MAKE THE CHARACTER BE DISPLAYED
                                                                               ;LEAVE IT 0 TO MAKE THE CHARACTER BE LOADED OVER AND OVE
```

```
1853
1854
```

```
                                        .SBTTL  SPECIAL TEST FOR SCOPE LOOPING
                                        ;T0037  YOU CAN DISPLAY ANY CHARACTER UP TO 80 TIMES PER LINE, AND UP TO 24 LINES PER SC
                                        ;       TO DO THIS SIMPLY SET LOCATION "YCODE" TO THE CHARACTER CODE YOU WISH TO DISPLAY
                                        ;       THEN SET LOCATION "YBLOCK" TO THE # OF TIMES THE CHARACTER SHOULD APPEAR ON EACH
                                        ;       THEN SET LOCATION "YLINES" TO THE # OF LINES YOU WANT DISPLAYED
013114  000101                          YCODE:  000101                  ;CODE, DEFAULT IS "A"
013116  000001                          YBLOCK: 000001                  ;# OF CHARS PER LINE, DEFAULT IS 1
013120  000002                          YLINES: 000002                  ;# OF LINES, DEFAULT IS 2

                                        ;       AFTER SETTING THE ABOVE, START HERE
013122                                  T0037:
013122  106427  000340                  YSTART: MTPS    #340            ;NO INTERRUPTS
013126  012706  000600                          MOV     #600,SP         ;SETUP THE STACK POINTER
013132  013737  013114  013110                  MOV     YCODE,XCODE
013140  004737  013654                          JSR     PC,GITCOD       ;GET ADDRESS OF CHARACTER TO LOAD
013144  012777  013602  165662                  MOV     #SINCHR,@DCP    ;SET ADDR OF CHAR
013152  012777  100006  165646                  MOV     #100006,@DCSR   ;START IT LOADING
013160  027727  165650  013652          1$:     CMP     @DCP,#ENDSIN    ;LOADED YET?
013166  001374                                  BNE     1$              ;WAIT TILL IT IS
                                        ;THE CHARACTER HAS BEEN LOADED.

                                        ;NOW BUILD TEXT AT LOCATION "MSGY"
013170  013700  013116                          MOV     YBLOCK,R0
013174  012701  013312                          MOV     #MSGY,R1
013200  112721  013114                  3$:     MOVB    YCODE,(R1)+     ;ADD A CHAR TO THE TEXT BLOCK
013204  005300                                  DEC     R0
013206  001374                                  BNE     3$              ;LONG ENOUGH BLOCK YET? IF NOT, GO BACK AND ADD MORE
013210  112721  000001                          MOVB    #1,(R1)+        ;END THE BLOCK WITH A EOL CHAR
                                        ;THE TEXT THAT WE WILL DISPLAY ON EACH LINE IS NOW SITTING AT LOCATION "MSGY"

                                        ;NOW BUILD A DISPLAY TABLE AT LOCATION "YDSPTB"
013214  013700  013116                          MOV     YBLOCK,R0       ;GET # OF CHARS/LINE
013220  005200                                  INC     R0              ;ADD 1 FOR EOL
013222  005100                                  COM     R0              ;MAKE IT NEGATIVE
013224  042700  174000                          BIC     #174000,R0      ;CLEAR OUT THE MODE BITS
013230  012704  000031                          MOV     #25.,R4         ;COUNT IN CASE "YLINES" IS TOO BIG
013234  013701  013120                          MOV     YLINES,R1       ;LINE COUNT
013240  012705  013436                          MOV     #YDSPTBL,R5     ;POINTER AT THE DISPLAY TABLE
013244  010025                          5$:     MOV     R0,(R5)+        ;PUT A COUNT INTO THE DISPLAY TABLE
013246  012725  013312                          MOV     #MSGY,(R5)+     ;PUT A POINTER TO THE TEXT IN ALSO
013252  005304                                  DEC     R4              ;MORE THAN 24 LINES SPECIFIED?
013254  001406                                  BEQ     7$              ;OH NO YOU DONT.
013256  005301                                  DEC     R1              ;DONE SPECIFIED # OF LINES YET?
013260  001371                                  BNE     5$              ;IF NOT, GO BACK AND PUT ANOTHER IN THE DISPLAY TABLE
013262  012725  003776                          MOV     #3776,(R5)+     ;YES. POLISH OFF THE TABLE
013266  012725  013434                          MOV     #MSGYA,(R5)+    ;WITH A EOS CHAR
                                        ;THE DISPLAY TABLE IS NOW SET UP

                                        ;ALL THER IS LEFT TO DO NOW IS TO START THE DISPLAY
013272  012777  013436  165530          7$:     MOV     #YDSPTB,@IDTP   ;POINTER AT DISPLAY TABLE
013300  012777  000146  165520                  MOV     #146,@DCSR      ;START UP THE DISPLAY
                                        ;WE SHOULD BE DISPLAYING ANY MOMENT NOW.

                                        ;GO INTO A DO NOTHING LOOP WHILE THE DISPLAY RUNS
013306  000137  013306                  9$:     JMP     9$              ;JUMP HERE FOREVER, OR TILL MANUALY STOPPED
```

```
1911
1912  013312  020040  020040  020040  MSGY:   .ASCII  /
1913  013320  020040  020040  020040
1914  013326  020040  020040  020040
1915  013334  020040  020040  020040
1916  013342  020040  020040  020040
1917  013350  020040  020040  020040
1918  013356  020040  020040  020040
1919  013364  020040  020040  020040
1920  013372  020040  020040  020040
1921  013400  020040  020040  020040
1922  013406  020040  020040  020040
1923  013414  020040  020040  020040
1924  013422  020040  020040  020040
1925  013430  020040    040
1926          013434                .EVEN
1927  013434  000000                MSGYA:  .WORD   000000                ;THIS IS THE END OF SCREEN TEXT
1928  013436  000031                YDSPTB:
1929  013436  000000                        000000                  ;COUNT
1930  013440  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1931  013442  000000                        000000                  ;COUNT
1932  013444  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1933  013446  000000                        000000                  ;COUNT
1934  013450  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1935  013452  000000                        000000                  ;COUNT
1936  013454  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1937  013456  000000                        000000                  ;COUNT
1938  013460  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1939  013462  000000                        000000                  ;COUNT
1940  013464  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1941  013466  000000                        000000                  ;COUNT
1942  013470  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1943  013472  000000                        000000                  ;COUNT
1944  013474  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1945  013476  000000                        000000                  ;COUNT
1946  013500  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1947  013502  000000                        000000                  ;COUNT
1948  013504  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1949  013506  000000                        000000                  ;COUNT
1950  013510  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1951  013512  000000                        000000                  ;COUNT
1952  013514  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1953  013516  000000                        000000                  ;COUNT
1954  013520  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1955  013522  000000                        000000                  ;COUNT
1956  013524  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1957  013526  000000                        000000                  ;COUNT
1958  013530  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1959  013532  000000                        000000                  ;COUNT
1960  013534  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1961  013536  000000                        000000                  ;COUNT
1962  013540  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1963  013542  000000                        000000                  ;COUNT
1964  013544  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
1965  013546  000000                        000000                  ;COUNT
1966  013550  013312                        MSGY                    ;POINTS TO TEXT BLOCKS
```

```
1967  013552  000000                        000000          ;COUNT
1968  013554  013312                        MSGY            ;POINTS TO TEXT BLOCKS
1969  013556  000000                        000000          ;COUNT
1970  013560  013312                        MSGY            ;POINTS TO TEXT BLOCKS
1971  013562  000000                        000000          ;COUNT
1972  013564  013312                        MSGY            ;POINTS TO TEXT BLOCKS
1973  013566  000000                        000000          ;COUNT
1974  013570  013312                        MSGY            ;POINTS TO TEXT BLOCKS
1975  013572  000000                        000000          ;COUNT
1976  013574  013312                        MSGY            ;POINTS TO TEXT BLOCKS
1977  013576  000000                        000000          ;COUNT
1978  013600  013312                        MSGY            ;POINTS TO TEXT BLOCKS
1979
1980
1981  013602  000000                SINCHR: .WORD   0
1982  013604  000000                        .WORD   0
1983  013606  000000                        .WORD   0
1984  013610  000000                        .WORD   0
1985  013612  000000                        .WORD   0
1986  013614  000000                        .WORD   0
1987  013616  000000                        .WORD   0
1988  013620  000000                        .WORD   0
1989  013622  000000                        .WORD   0
1990  013624  000000                        .WORD   0
1991  013626  000000                        .WORD   0
1992  013630  000000                        .WORD   0
1993  013632  000000                        .WORD   0
1994  013634  000000                        .WORD   0
1995  013636  000000                        .WORD   0
1996  013640  000000                        .WORD   0
1997  013642  000000                        .WORD   0
1998  013644  000000                        .WORD   0
1999  013646  000000                        .WORD   0
2000  013650  000000                        .WORD   0
2001  013652  000000                ENDSIN: .WORD   0
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015  013654  012700  016110          GITCOD: MOV     #CHARS,R0
2016  013660  012037  000766          1$:     MOV     (R0)+,TEMP
2017  013664  123737  013110  000767          CMPB    XCODE,TEMP+1
2018  013672  001406                          BEQ     2$
2019  013674  020027  022340                  CMP     R0,#ENDCHR
2020  013700  001367                          BNE     1$
2021  013702  012700  016730                  MOV     #SPCHAR,R0
2022  013706  000402                          BR      3$
```

```
2023  013710  162700  000002          2$:     SUB     #2,R0
2024  013714  012701  000024          3$:     MOV     #20.,R1
2025  013720  012702  013602                  MOV     #SINCHR,R2
2026  013724  112022                  4$:     MOVB    (R0)+,(R2)+     ;XFR 1
2027  013726  113722  013110                  MOVB    XCODE,(R2)+     ;XFR CODE
2028  013732  005200                          INC     R0
2029  013734  005301                          DEC     R1
2030  013736  001372                          BNE     4$
2031  013740  005022                          CLR     (R2)+
2032  013742  105037  013111                  CLRB    XCODE+1         ;EOS CHAR
2033  013746  000207                          RTS     PC
2034
```

```
                                        .SBTTL  TUBE OUTPUT SUBROUTINE
                                        ;SUBROUTINE TO TAKE TEXT, COUNT CHARACTERS & HAVE EACH LINE
                                        ;INSERTED INTO THE DISPLAY QUEUE
                                        ;CALL WITH ADDRESS OF MESSAGE IN R5
013750  012777  015760  165052  TUBOUT: MOV     #DISTBL,@IDTP   ;SETUP DISPLAY TABLE ADDRESS
013756  012777  000146  165042          MOV     #146,@DCSR      ;MAKE SURE DISPLAY IS GOING
013764  010546                          MOV     R5,-(SP)        ;SAVE ADDR OF MESSAGE
013766  010537  001046          1$:     MOV     R5,TUBTM1       ;MAKE A WORKING COPY
013772  012737  003777  001036          MOV     #3777,CHRCNT    ;ZERO THE CHARACTER COUNT
014000  112537  001044          2$:     MOVB    (R5)+,TUBTM0    ;GET A CHARACTER
014004  001012                          BNE     4$              ;IS FINAL END OF MESSAGE?
014006  023727  001036  003777          CMP     CHRCNT,#3777    ;YES, FIND OUT HOW MANY CHARACTERS ARE IN IT
014014  001402                          BEQ     3$              ;ANY AT ALL?
014016  004737  014056                  JSR     PC,INSERT       ;YES, INSERT THEM INTO THE BUFFER
014022  012605                  3$:     MOV     (SP)+,R5        ;RESTORE THE ADDRESS OF THE MESSAGE
014024  005037  000754                  CLR     SPMODE          ;CLEAR OUT SPECIAL MODE BITS
014030  000207                          RTS     PC              ;RETURN
014032  023727  001044  000012  4$:     CMP     TUBTM0,#12      ;END OF A LINE?
014040  001003                          BNE     5$              ;IF NOT, GO FIX THE CHARACTER COUNT
014042  004737  014056                  JSR     PC,INSERT       ;IF IT IS THE END PUT THE LINE INTO THE MESSAGE BUFFER
014046  000747                          BR      1$              ;AND SETUP TO DO THE SAME FOR THE OTHER LINES IN THE MES
014050  005337  001036          5$:     DEC     CHRCNT          ;ADD 1 TO THE CHARACTER COUNT
014054  000751                          BR      2$              ;GO COUNT THE REST OF THE CHARACTERS


                                        .SBTTL  DISPLAY TABLE ENRTY INSERTER ROUTINE
                                        ;SUBROUTINE TO ADD A NEW ENTRY TO THE DISPLAY QUEUE
                                        ;CALL WITH CHARACTER COUNT IN CHRCNT
                                        ;AND WITH THE TEXT ADDRESS IN TUBTM1
014056  010546                  INSERT: MOV     R5,-(SP)        ;COMMANDEER 3 REGISTERS
014060  010446                          MOV     R4,-(SP)        ;FIRST SAVE THIER CONTENTS
014062  010346                          MOV     R3,-(SP)        ;SO THAT WE CAN RESTORE THEM LATER
014064  053737  000754  001036          BIS     SPMODE,CHRCNT   ;SET ANY SPECIFIED MODE BITS
014072  012705  000023                  MOV     #19.,R5         ;SETUP A COUNT OF 19 SHIFTS
014076  012704  015760                  MOV     #DISTBL,R4      ;DISPLAY TABLE ADDRESS IN R4
014102  012703  015764                  MOV     #DISTBL+4,R3    ;ADDRESS OF 2ND ENTRY OF DISPLAY TABLE
014106  012324                  1$:     MOV     (R3)+,(R4)+     ;SHIFT THE COUNT BY 1 ENTRY
014110  012324                          MOV     (R3)+,(R4)+     ;SHIFT THE ADDRESS BY 1 ENTRY
014112  005305                          DEC     R5              ;DONE 10 ENTRYS YET?
014114  001374                          BNE     1$              ;IF NOT DO SOME MORE
014116  013724  001036                  MOV     CHRCNT,(R4)+    ;YES WE HAVE DONE 10, PUT NEW CHAR COUNT INTO TABLE
014122  013724  001046                  MOV     TUBTM1,(R4)+    ;PUT NEW TEXT POINTER INTO THE TABLE
014126  012603                          MOV     (SP)+,R3        ;RESTORE THE REGISTERS
014130  012604                          MOV     (SP)+,R4        ;THAT WE USED SO THAT WE
014132  012605                          MOV     (SP)+,R5        ;DONT CONFUSE OTHER ROUTINES
014134  000207                  2$:     RTS     PC              ;RETURN


                                        .SBTTL  DISPLAY TABLE CLEARING ROUTINE
                                        ;SUBROUTINE TO CLEAR OUT THE DISPLAY TABLE
014136  012777  000146  164662  CLRTUB: MOV     #146,@DCSR      ;TELL THE DISPLAY TO GO
014144  004737  015226                  JSR     PC,SSTALL       ;WAIT FOR IT TO STOP
014150  010046                          MOV     R0,-(SP)        ;SAVE R0
014152  012700  015760                  MOV     #DISTBL,R0      ;SET ADDR OF DISPLAY TABLE
014156  012720  003740          1$:     MOV     #3740,(R0)+     ;PUT COUNT IN THE TABLE
```

```
2091  014162  012720  027552                  MOV     #MSG37,(R0)+    ;PUT MESSAGE ADDRESS IN THE TABLE
2092  014166  020027  016074                  CMP     R0,#TBL22       ;ALMOST FULL?
2093  014172  001371                          BNE     1$              ;IF NOT GO BACK AND FILL IT SUMORE
2094  014174  012720  003766                  MOV     #3766,(R0)+     ;ALMOST FULL
2095  014200  012720  027552                  MOV     #MSG37,(R0)+    ;THIS SHOULD FILL IT
2096  014204  012600                          MOV     (R6)+,R0        ;RESTORE R0
2097  014206  000207                          RTS     PC              ;RETURN
2098                                  .SBTTL  SCOPE HANDLER ROUTINE
2099
2100                                  ;;**********************************************************************
2101                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS, IT WILL INCREMENT
2102                                  ;*AND LOAD THE TEST NUMBER(STSTNM) INTO THE DISPLAY REG,(DISPLAY<7:0>)
2103                                  ;*AND LOAD THE ERROR FLAG (SERFLG) INTO DISPLAY<15:08>
2104                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
2105                                  ;*SW14=1        LOOP ON TEST
2106                                  ;*SW11=1        INHIBIT ITERATIONS
2107                                  ;*SW09=1        LOOP ON ERROR
2108                                  ;*SW08=1        LOOP ON TEST IN SWR<7:0>
2109                                  ;*CALL
2110                                  ;*      SCOPE           ;;SCOPE=IOT
2111
2112  014210                          $SCOPE:
2113  014210  032777  010000  164422          BIT     #10000P,@#SWR   ;FIND OUT IF SWITCH REG BIT 12 IS SET
2114  014216  001402                          BEQ     14$             ;IF NOT DONT STALL
2115  014220  004737  015002                  JSR     PC,LSTALL       ;IT IS SET, STALL FOR A SECOND OR SO
2116  014224  105777  164414          14$:    TSTB    @#TKS           ;CHAR IN ADDITIONAL KEY BUFFER?
2117  014230  100013                          BPL     77$             ;IF NOT
2118  014232  017746  164410                  MOV     @#TKB,-(SP)     ;GET CHARS VALUE
2119  014236  042716  177600                  BIC     #177600,(SP)    ;CLEAR ITS PARITY BIT
2120  014242  022627  000022                  CMP     (SP)+,#22       ;IS A ^R CODE IN THE ADITIONAL KEYBOARD BUFFER?
2121  014246  001004                          BNE     77$             ;IF NOT, JUST GO ABOUT OUR BUSINESS
2122  014250  005037  000602                  CLR     $TSTNM          ;ZERO THE TEST #
2123  014254  000137  001514                  JMP     T0000           ;YES THERE IS A ^R THERE, RESTART
2124  014260  000240                  77$:    NOP
2125  014262  032777  040000  164350  1$:     BIT     #BIT14,@#SWR    ;;LOOP ON PRESENT TEST?
2126  014270  001111                          BNE     $OVER           ;;YES IF SW14=1
2127                                  ;#####START OF CODE FOR THE XOR TESTER#####
2128  014272  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
2129                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
2130  014274  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
2131  014300  012737  014320  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
2132  014306  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
2133  014312  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
2134  014316  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
2135  014320  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
2136  014322  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
2137  014326  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
2138  014330                          6$:;#####END OF CODE FOR THE XOR TESTER#####
2139  014330  032777  000400  164302          BIT     #BIT08,@#SWR    ;;LOOP ON SPEC. TEST?
2140  014336  001404                          BEQ     2$              ;;BR IF NO
2141  014340  127737  164274  000602          CMPB    @#SWR,$TSTNM    ;;ON THE RIGHT TEST?    SWR<7:0>
2142  014346  001462                          BEQ     $OVER           ;;BR IF YES
2143  014350  105737  000603          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
2144  014354  001421                          BEQ     3$              ;;BR IF NO
2145  014356  123737  000615  000603          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
2146  014364  101015                          BHI     3$              ;;BR IF NO
```

```
2147  014366  032777  001000  164244          BIT     $BIT09,@SWR     ;;LOOP ON ERROR?
2148  014374  001404                          BEQ     4$              ;;BR IF NO
2149  014376  013737  000610  000606  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
2150  014404  000443                          BR      $OVER
2151  014406  105037  000603          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
2152  014412  005037  000722                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
2153  014416  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
2154  014420  032777  004000  164212  3$:     BIT     $BIT11,@SWR     ;;INHIBIT ITERATIONS?
2155  014426  001011                          BNE     1$              ;;BR IF YES
2156  014430  005737  000600                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
2157  014434  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
2158  014436  005237  000604                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
2159  014442  023737  000722  000604          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
2160  014450  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
2161  014452  012737  000001  000604  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
2162  014460  013737  014530  000722          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
2163  014466  105237  000602          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
2164  014472  011637  000606                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
2165  014476  011637  000610                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
2166  014502  005037  000724                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
2167  014506  112737  000001  000615          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
2168  014514  013777  000602  164120  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
2169  014522  013716  000606                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
2170  014526  000002                          RTI                     ;;FIXES PS
2171  014530  000001                  $MXCNT: 1                       ;;MAX. NUMBER OF ITERATIONS
2172                                  .EVEN
2173                                  .SBTTL  SUBROUTINE FOR REPORTING ERROR MESSAGES
2174                                  ;THIS SUBROUTINE DISPLAYS ERROR MESSAGES
2175                                  ;UNLESS THE SOFTWARE SWITCH REGISTER BIT  13 IS SET, IT PUTS THE
2176                                  ;ERROR MESSAGE POINTED TO BY R5 INTO THE REGULAR
2177                                  ;DISPLAY TABLE, THEN SENDS THE MESSAGE TO ANY ADDITIONAL TERMINAL,
2178                                  ;IF SOFTWARE SWITCH REGISTER BIT 10 IS SET, IT ALSO RINGS THE ADDITIONAL
2179                                  ;TERMINALS BELL, AND CAUSES THE VT71 BUZZER TO BUZZ
2180  014532  010546                  ERRMES: MOV     R5,-(SP)        ;SAVE R5 CONTENTS
2181  014534  105237  000603          1$:     INCB    $ERFLG          ;SET ERROR FLAG
2182  014540  001775                          BEQ     1$              ;MAKE SURE IT IS NOT 0
2183  014542  005237  000612                  INC     $ERTTL          ;ADD 1 TO THE ERROR COUNT
2184  014546  032777  002000  164064          BIT     #BIT10,@SWR     ;IS BIT 10 SET
2185  014554  001406                          BEQ     2$              ;IF NOT DONT RING BELL ON ERROR
2186  014556  012705  000726                  MOV     #$BELL,R5       ;IF BIT10 IS SET
2187  014562  004737  014722                  JSR     PC,TTYOUT       ;RING THE BELL
2188  014566  004737  014662                  JSR     PC,BUZZ         ;MAKE THE VT71 BUZZER BUZZ
2189  014572  032777  020000  164040  2$:     BIT     #20000,@SWR     ;INHIBIT ERROR TYPEOUTS?
2190  014600  001402                          BEQ     3$              ;
2191  014602  012605                          MOV     (SP)+,R5        ;YES. RESTORE R5
2192  014604  000207                          RTS     PC              ;AND RETURN
2193  014606  016637  000002  000766  3$:     MOV     2(SP),TEMP      ;GET ERROR PC
2194  014614  012705  030611                  MOV     #M$G54A,R5      ;SETUP ADDRESS TO PUT ASCII VALUE OF ERROR PC
2195  014620  004737  015504                  JSR     PC,BIOCT        ;CONVERT ERROR PC VALUE TO ASCII
2196  014624  012705  030602                  MOV     #M$G54,R5       ;SETUP ADDR OF PC PRINTOUT
2197  014630  004737  014752                  JSR     PC,MSGOUT       ;DISPLAY ERROR PC VALUE
2198  014634  012605                          MOV     (SP)+,R5        ;GET ERROR MESSAGE ADDRESS
2199  014636  004737  014752                  JSR     PC,MSGOUT       ;DISPLAY ERROR MESSAGE
2200  014642  004737  015002                  JSR     PC,LSTALL       ;GIVE TIME FOR IT TO BE SEEN
2201  014646  032777  100000  163764          BIT     #BIT15,@SWR     ;HALT ON ERROR BIT SET?
2202  014654  001401                          BEQ     CONER           ;IF NOT, DONT HALT!
```

```
2203  014656  000000                  HALTER: HALT                    ;IT IS SET SO WE  DO  HALT
2204  014660  000207                  CONER:  RTS     PC
2205
2206                                  .SBTTL  BUZZ NOISE MAKING SUBROUTINE
2207                                  ;THIS SUBROUTINE CAUSES THE VT71 BUZZER TO EMIT A BUZZ
2208  014662  010046                  BUZZ:   MOV     R0,-(SP)        ;SAVE R0
2209  014664  005077  164114                  CLR     @LCSR           ;DISABLE LED INTERRUPTS
2210  014670  012700  177600                  MOV     #177600,R0      ;SETUP A COUNT FOR 200 CLICKS
2211  014674  012777  100000  164104  1$:     MOV     #100000,@LBUF   ;DO A CLICK
2212  014702  022777  000200  164074  2$:     CMP     #200,@LCSR      ;IS LED READY BIT SET?
2213  014710  001374                        BNE     2$              ;IF NOT GO BACK AND TEST IT AGAIN
2214  014712  005300                        DEC     R0              ;CHALK UP ANOTHER CLICK
2215  014714  001367                        BNE     1$              ;IF WE HAVEN'T DONE 200 GO BACK AND DO SOME MORE
2216  014716  012600                        MOV     (SP)+,R0        ;IF WE HAVE DONE 200, RESTORE R0
2217  014720  000002                        RTI                     ;AND RETURN
2218
2219
2220                                  .SBTTL  TTY OUTPUT SUBROUTINE
2221                                  ;CALL WITH A 'JSR PC'
2222                                  ;WITH ADDRESS OF THE MESSAGE IN R5
2223                                  ;MESSAGE SHOULD BE IN 8BIT ASCII PACKED
2224                                  ;1 CHARACTER PER WORD WITH A NULL
2225                                  ;CHARACTER ACTING AS A 'END OF MESSAGE' FLAG
2226  014722  005737  001042          TTYOUT: TST     TTYAVA          ;IS A TTY AVAILABLE?
2227  014726  001410                        BEQ     RTNTT           ;IF NOT RETURN RIGHT AWAY
2228  014730  010546                        MOV     R5,-(SP)        ;SAVE R5 IF THERE IS A TTY
2229  014732  105777  164036          TTOUT:  TSTB    @TPS            ;IS THE PRINTER READY?
2230  014736  100375                        BPL     TTOUT           ;IF NOT TEST IT AGAIN
2231  014740  112577  164032                  MOVB    (R5)+,@TPB      ;PRINT A CHARACTER
2232  014744  001372                        BNE     TTOUT           ;IF ITS NOT A NULL LOOP BACK AND PRINT ANOTHER
2233  014746  012605                        MOV     (SP)+,R5        ;RESTORE R5
2234  014750  000207          RTNTT:  RTS     PC                      ;IF IT IS A NULL RETURN
2235
2236
2237                                  .SBTTL  REGULAR MESSAGE ROUTINE, SCREEN AND TTY
2238                                  ;THIS IS THE MESSAGE OUTPUT ROUTINE
2239                                  ;IT IS CALLED WITH A'JSR PC' INSTRUCTION
2240                                  ;WITH R5 SET TO POINT TO THE ASCII TEXT OF THE MESSAGE
2241  014752  005737  001042          MSGOUT: TST     TTYAVA          ;IS A EXTRA TERMINAL HOOKED UP?
2242  014756  001004                        BNE     1$              ;IF SO NO EXTRA STALLING TIME IS NEEDED
2243  014760  004737  015002                  JSR     PC,LSTALL       ;MORE TIME PLEASE
2244  014764  004737  015002                  JSR     PC,LSTALL       ;IF NOT, WASTE TIME SO THAT THE DISPLAY
2245                                                                  ;DOESNT CHANGE TOO FAST FOR HUMAN EYES
2246  014770  004737  013750          1$:     JSR     PC,TUBOUT       ;DISPLAY THE MESSAGE ON THE VT71 SCREEN
2247  014774  004737  014722                  JSR     PC,TTYOUT       ;DISPLAY MESSAGE ON THE TELLETYPPE IF AVAILABLE
2248  015000  000207                        RTS     PC              ;RETURN
2249
2250                                  .SBTTL  1.5 SECOND TIME WASTING SUBROUTINE
2251                                  ;THIS IS A TIME WASTING ROUTINE
2252                                  ;IT IS USED ANYWHERE A STALL OR WAIT IS  NEEDED.
2253                                  ;THIS ROUTINE PROVIDES ABOUT 1.5 SECONDS OF WAITING TIME
2254  015002  005037  000772          LSTALL: CLR     STLCNT          ;CLEAR OUT THE STALL COUNTER
2255  015006  005737  000772          2$:     TST     STLCNT          ;WASTE SOME TIME
2256  015012  005237  000772                  INC     STLCNT          ;TICK - INCREMENT COUNT
2257  015016  001373                        BNE     2$              ;KEEP WAITING UNTIL THE COUNT REACHES 0
2258  015020  000207                        RTS     PC              ;COUNT=0 - DONE WAITING - RETURN
```

```
2259
2260
2261
2262                                    .SBTTL  KEYBOARD INPOUT CHECKING ROUTINE
2263                                    ;THIS ROUTINE CHECKS THE ADDITIONAL KEYBOARD FOR SPECIAL CHARACTERS
2264                                    ;AND ACTS OPON THEM ACORDINGLY
2265  015022  000240                    SPCHK:  NOP
2266  015024  105777  163614                    TSTB    @STKS           ;IS A CHAR WAITING IN THE ADITIONAL KEYBOARD BUFFER
2267  015030  100022                            BPL     2$              ;IF NOT, SKIP OVER THE NEXT STUFF
2268  015032  017737  163610  000766            MOV     @STKB,TEMP      ;RESCUE THE CHAR FROM THE BUFFER
2269  015040  042737  000200  000766            BIC     #200,TEMP       ;CLEAR OUT THE PARITY BIT IFIT IS SET
2270  015046  022737  000022  000766            CMP     #22,TEMP        ;IS THE CHAR  ^R  ?
2271  015054  001004                            BNE     1$              ;IF NOT ^R,TRY ^A
2272  015056  012706  000600                    MOV     #600,SP         ;IT IS  ^R, RESET THE STACK
2273  015062  000137  001062                    JMP     START           ;AND RESTART THE PROGRAM
2274  015066  022737  000001  000766  1$:       CMP     #1,TEMP         ;IS CHAR ^A ?
2275  015074  001000                            BNE     2$              ;IF NOT IGNORE IT
2276  015076  000240                    2$:     NOP
2277  015100  000207                            RTS     PC
2278
2279
2280                                    .SBTTL  TEST INITIALIZATION ROUTINE
2281                                    ;THIS SUBROUTINE IS CALLED AT THE BEGINING OF EACH TEST.
2282                                    ;IT DISPLAYS THE TEST # IN THE VT71'S LEDS, IT PRINTS THE TEST
2283                                    ;# ON THE ADDITIONAL TERMINAL, AND IT LEAVES ASCII TEXT FOR THE
2284                                    ;CURRENT TEST # AT LOCATION "MSG17", SO THAT INDIVIDUAL TESTS CAN EASILY
2285                                    ;DISPLAY THE TEST #
2286  015102  005077  163700            FXTST:  CLR     @LBUF           ;0 TO LIGHTS
2287  015106  013737  000602  000766            MOV     $TSTNM,TEMP     ;GET TEST #
2288  015114  105037  000767                    CLRB    TEMP+1          ;DO NOT INCLUDE ERROR COUNT
2289  015120  013777  000766  163660            MOV     TEMP,@LBUF      ;DISPLAY THE TEST # IN THE LEDS
2290  015126  012705  026247                    MOV     #MSG29A,R5      ;SETUP MSG29A OF ASCII
2291  015132  004737  015504                    JSR     PC,BIOCT        ;CONVERT DATA INTO ASCII
2292  015136  012705  026240                    MOV     #MSG29,R5       ;SETUP ADDRESS OF TEST # MESSAGE
2293  015142  004737  014722                    JSR     PC,TTYOUT       ;DISPLAY TEST # ON TTY ONLY(IF AVAILABLE)
2294  015146  013737  000602  000766            MOV     $TSTNM,TEMP     ;GET TEST #
2295  015154  105037  000767                    CLRB    TEMP+1          ;DO NOT INCLUDE ERROR COUNT
2296  015160  012705  024620                    MOV     #MSG17A,R5      ;SETUP MSG17A OF ASCII
2297  015164  004737  015504                    JSR     PC,BIOCT        ;CONVERT DATA INTO ASCII
2298  015170  000207                            RTS     PC
2299
2300
2301                                    .SBTTL  .1 SECOND OF TIME WASTING SUBROUTINE
2302                                    ;THIS IS THE MEDIUM STALL ROUTINE
2303                                    ;IT SATLLS FOR ABOUT A TENTH OF A SECOND
2304  015172  012737  170000  000772    MSTALL: MOV     #170000,STLCNT
2305  015200  005237  000772            1$:     INC     STLCNT          ;INC COUNTER TILL IT REACHES 0
2306  015204  001375                            BNE     1$              ;IF IT HAS NOT REACHED 0 YET GO BACK
2307  015206  000207                            RTS     PC              ;IF IT HAS REACHED 0, RETURN
2308                                    .SBTTL  .02 SECOND OF TIME WASTING SUBROUTINE
2309                                    ;THIS IS THE MEDIUM STALL ROUTINE
2310                                    ;IT SATLLS FOR ABOUT TWO HUNDRETHS OF A SECOND
2311  015210  012737  176000  000772    XSTALL: MOV     #176000,STLCNT
2312  015216  005237  000772            1$:     INC     STLCNT          ;INC COUNTER TILL IT REACHES 0
2313  015222  001375                            BNE     1$              ;IF IT HAS NOT REACHED 0 YET GO BACK
2314  015224  000207                            RTS     PC              ;IF IT HAS REACHED 0, RETURN
```

```
                                    .SBTTL  400US STALL ROUTINE
                                    ;THIS IS ANOTHER TIME WASTING SUBROUTING
                                    ;IT STALLS FOR ONLY A FEW HUNDRED MICROSECONDS OR SO
015226  012737  177740  000772    SSTALL: MOV     #177740,STLCNT
015234  005237  000772              1$:     INC     STLCNT          ;INC COUNTER TILL IT REACHES 0
015240  001375                            BNE     1$              ;IF IT HAS NOT REACHED 0 YET GO BACK
015242  000207                            RTS     PC              ;IF IT HAS REACHED 0, RETURN


                                    .SBTTL  LED AND KEYBOARD INTERRUPT SERVICE ROUTINES
                                    ;THIS IS A "JUST IN CASE" SERVICE ROUTINE, FOR LED AND KEYBOARD INTERRUPTS
                                    ;IT JUST RETURNS CONTROL BACK TO THE INTERRUPTED SECTION
015244                              KBSRV:
015244  000002                      LDSRV:  RTI                     ;RETURN


                                    .SBTTL  DISPLAY INTERRUPT SERVICE ROUTINE
                                    ;THIS IS THE DISPLAY INTERRUPT SERVICE ROUTINE,
                                    ;IT JUST ADDS 1 TO A INTERRUPT COUNTER LOCATION, THAN IT
                                    ;RETURNS USING "RTI"
015246  005237  001050              DIHAN:  INC     INTCNT          ;ADD 1 TO INTERRUPT COUNTER
015252  000002                            RTI                     ;RETURN


                                    .SBTTL  SUBROUTINE TO WAIT ON CHARACTER SET LOADING.
                                    ;SUBROUTINE TO WAIT ON CHARACTER SET LOADING,
                                    ;IF CHAR SET DOES NOT LOAD WITHIN 2 SECONDS AN ERROR MESSAGE IS
                                    ;SENT TO ANY ADDITIONAL TELLETYPE, ALSO IT IS DISPLAYED ON THE VT71
                                    ;SCREEN IF LOCATION "TUBCNT" IS SET TO A NON-ZERO VALUE BEFORE CALLING
                                    ;THIS SUBROUTINE.
                                    ;BEFORE CALLING  R5 SHOULD BE EQUAL TO THE ADDRESS OF THE CHARACTER SETS END
015254  010046                      TSTLOD: MOV     R0,-(SP)        ;SAVE A REG SO WE CAN USE IT IN THIS SUBROUTINE
015256  012700  170000                      MOV     #170000,R0      ;PREPARE IT FOR USE
015262  020577  163546              1$:     CMP     R5,@DCP         ;DONE LOADING?
015266  001450                            BEQ     5$              ;IF SO.
015270  004737  015226                      JSR     PC,SSTALL       ;NO. WASTE SOME TIME
015274  005200                            INC     R0              ;WASTED 2 SECONDS WAITING YET?
015276  001371                            BNE     1$              ;IF NOT GO  BACK AND WASTE SOME MORE
015300  010537  000766                      MOV     R5,TEMP         ;ADDRESS OF GOOD
015304  012705  030567                      MOV     #MSG45B,R5
015310  004737  015504                      JSR     PC,BIOCT        ;CONVERT S/B DATA TO ASCII
015314  017737  163514  000766            MOV     @DCP,TEMP       ;GET WAS DATA
015322  012705  030531                      MOV     #MSG45A,R5
015326  004737  015504                      JSR     PC,BIOCT        ;CONVERT BAD DATA TO ASCII
015332  004737  014136                      JSR     PC,CLRTUB       ;CLEAR OUT THE DISPLAY TABLE
015336  012777  015760  163464            MOV     #DISTRL,@IDTP
015344  032777  020000  163266            BIT     #20000,@SWR     ;IS THE INHIBIT ERROR MESSAGES BIT SET?
015352  001016                            BNE     5$              ;IF SO, DO NO TYPEOUTS
015354  012705  030453                      MOV     #MSG45,R5       ;ADDRESS OF ERROR MESSAGE
```

```
                        SUBROUTINE TO WAIT ON CHARACTER SET LOADING.

2371  015360  004737  014722          JSR     PC,TTYOUT       ;SEND MESSAGE TO THE TTY
2372  015364  005737  000756          TST     TUBSWT          ;ALSO TO THE VT71 SCREEN?
2373  015370  001007                  BNE     5$              ;IF NOT, JUST GO AND RETURN
2374  015372  012777  000146  163426  MOV     #146,@DCSR      ;DISPLAY REGULAR MODE
2375  015400  012705  030453          MOV     #MSG45,R5       ;YES TO THE SCREEN, GET MESSAGE ADDRESS
2376  015404  004737  013750          JSR     PC,TUBOUT       ;DISPLAY MESSAGE
2377  015410  012600                  5$:     MOV     (SP)+,R0        ;RESTORE R0 TO ITS FORMER GLORY
2378  015412  005037  000756          CLR     TUBSWT          ;INIT THE TUBE DISPLAY SWITCH
2379  015416  000207                  RTS     PC              ;RETURN
2380
2381
2382
2383
2384                                .SBTTL  TRAP TO VECTOR 4 HANDLING ROUTINE
2385                                ;IF A TRAP TO LOC 4 OCCURS WHEN WE DO NOT EXPECT ONE, WE END UP HERE
2386                                ;A ERROR MESSAGE WILL BE PRINTED OUT, AND A RTT INSTRUCTION EXECUTED
2387  015420  005077  163402          TRAPER: CLR     @DCSR           ;STOP THE DISPLAY MOMENTARALY
2388  015424  004737  015002          JSR     PC,LSTALL       ;GIVE IT TIME TO STOP
2389  015430  004737  014136          JSR     PC,CLRTUB       ;CLEAR OUT DISPLAY TABLE
2390  015434  012737  015760  001030  MOV     #DISTRL,IDTP    ;GIVE THE DISPLAY PROCESSOR ITS ADDRESS
2391  015442  012777  100146  163356  MOV     #100146,@DCSR   ;AND RESTART THE DISPLAY
2392  015450  012705  026453          MOV     #MSX32,R5       ;SETUP ADDRESS OF PLACE TO PUT ASCII OF TRAP PC CONTENTS
2393  015454  011637  000766          MOV     (SP),TEMP       ;GET TRAP PC
2394  015460  162737  000002  000766  SUB     #2,TEMP         ;MAKE IT RIGHT
2395  015466  004737  015504          JSR     PC,BIOCT        ;CHANGE TRAP PC INTO ASCII
2396  015472  012705  026421          MOV     #MSG32,R5       ;GET ADDR OF THE ASCII ERROR MESSAGE
2397  015476  004737  014752          JSR     PC,MSGOUT       ;DISPLAY THE 'TRAP ERROR' MESSAGE
2398  015502  000006                  RTT                     ;TRY TO CONTINUE
2399
2400                                .SBTTL  BINARY TO ASCII CONVERT SUBROUTINE.
2401                                ;BINARY TO ASCII CONVERT SUBROUTINE.
2402                                ;CALL USING A 'JSR PC'
2403                                ;DERIVES ASCII CHARACTERS REPRESENTING THE CONTENTS
2404                                ;OF LOCATION 'TEMP', AND PUTS THEM INTO THE 6 BYTES POINTED TO
2405                                ;BY R5
2406                                ;THIS IS A STOLLEN ROUTINE. IT IS ROTTENLY WRITEN
2407
2408  015504  010446                  BIOCT:  MOV     R4,-(SP)        ;SAVE R4
2409  015506  012704  031477          MOV     #BIOTMP,R4      ;SETUP POINTER TO TEMP TEXT SPACE
2410  015512  113764  000766  000005  MOVB    TEMP,5(R4)
2411  015520  006037  000766          ROR     TEMP
2412  015524  113764  000767  000002  MOVB    TEMP+1,2(R4)
2413  015532  006037  000766          ROR     TEMP
2414  015536  006037  000766          ROR     TEMP
2415  015542  113764  000766  000004  MOVB    TEMP,4(R4)
2416  015550  006037  000766          ROR     TEMP
2417  015554  113764  000767  000001  MOVB    TEMP+1,1(R4)
2418  015562  006037  000766          ROR     TEMP
2419  015566  006037  000766          ROR     TEMP
2420  015572  113764  000766  000003  MOVB    TEMP,3(R4)
2421  015600  006037  000766          ROR     TEMP
2422  015604  113714  000767          MOVB    TEMP+1,(R4)
2423  015610  142714  000376          BICB    #376,(R4)
2424  015614  142764  000370  000001  BICB    #370,1(R4)
2425  015622  142764  000370  000002  BICB    #370,2(R4)
2426  015630  142764  000370  000003  BICB    #370,3(R4)
```

```
2427  015636  142764  000370  000004          BICB    #370,4(R4)
2428  015644  142764  000370  000005          BICB    #370,5(R4)
2429  015652  152714  000060                  BISB    #60,(R4)
2430  015656  152764  000060  000001          BISB    #60,1(R4)
2431  015664  152764  000060  000002          BISB    #60,2(R4)
2432  015672  152764  000060  000003          BISB    #60,3(R4)
2433  015700  152764  000060  000004          BISB    #60,4(R4)
2434  015706  152764  000060  000005          BISB    #60,5(R4)
2435  015714  111415                          MOVB    (R4),(R5)
2436  015716  116465  000001  000001          MOVB    1(R4),1(R5)
2437  015724  116465  000002  000002          MOVB    2(R4),2(R5)
2438  015732  116465  000003  000003          MOVB    3(R4),3(R5)
2439  015740  116465  000004  000004          MOVB    4(R4),4(R5)
2440  015746  116465  000005  000005          MOVB    5(R4),5(R5)
2441  015754  012604                          MOV     (SP)+,R4        ;RESTORE R4
2442  015756  000207                          RTS     PC              ;YEAH
2443
2444
2445
2446
2447  015760  003740                  DISTBL: .WORD   3740
2448  015762  030624                          .WORD   MSG57
2449  015764  003740                          .WORD   3740
2450  015766  030624                          .WORD   MSG57
2451  015770  003740                          .WORD   3740
2452  015772  030624                          .WORD   MSG57
2453  015774  003740                          .WORD   3740
2454  015776  030624                          .WORD   MSG57
2455  016000  003740                          .WORD   3740
2456  016002  030624                          .WORD   MSG57
2457  016004  003740                          .WORD   3740
2458  016006  030624                          .WORD   MSG57
2459  016010  003740                          .WORD   3740
2460  016012  030624                          .WORD   MSG57
2461  016014  003740                          .WORD   3740
2462  016016  030624                          .WORD   MSG57
2463  016020  003740                          .WORD   3740
2464  016022  030624                          .WORD   MSG57
2465  016024  003740                          .WORD   3740
2466  016026  030624                          .WORD   MSG57
2467  016030  003740                          .WORD   3740
2468  016032  030624                          .WORD   MSG57
2469  016034  003740                          .WORD   3740
2470  016036  030624                          .WORD   MSG57
2471  016040  003740                          .WORD   3740
2472  016042  030624                          .WORD   MSG57
2473  016044  003740                          .WORD   3740
2474  016046  030624                          .WORD   MSG57
2475  016050  003740                          .WORD   3740
2476  016052  030624                          .WORD   MSG57
2477  016054  003740                          .WORD   3740
2478  016056  030624                          .WORD   MSG57
2479  016060  003740                          .WORD   3740
2480  016062  030624                          .WORD   MSG57
2481  016064  003740                          .WORD   3740
2482  016066  030624                          .WORD   MSG57
```

```
2482  016070  003770                    .WORD   3770
2483  016072  030624                    .WORD   MSG57
2484  016074  003766          TBL22:    .WORD   3766
2485  016076  030624                    .WORD   MSG57
2486  016100  003774          TBL23:    .WORD   3774
2487  016102  030064                    .WORD   MSG41
2488  016104  100000                    100000
2489  016106  015760                    DISTBL
2490                          ;END OF DISPLAY TABLE
2491
2492
2493
2494
2495
2496                                    .SBTTL  CHARACTER SET
2497  016110      340     064   CHARS:    .BYTE 340,064           ;NUMBER '4'
2498  016112      300     064             .BYTE 300,064
2499  016114      340     064             .BYTE 340,064
2500  016116      302     064             .BYTE 302,064
2501  016120      360     064             .BYTE 360,064
2502  016122      302     064             .BYTE 302,064
2503  016124      344     064             .BYTE 344,064
2504  016126      302     064             .BYTE 302,064
2505  016130      366     064             .BYTE 366,064
2506  016132      312     064             .BYTE 312,064
2507  016134      340     064             .BYTE 340,064
2508  016136      302     064             .BYTE 302,064
2509  016140      340     064             .BYTE 340,064
2510  016142      302     064             .BYTE 302,064
2511  016144      340     064             .BYTE 340,064
2512  016146      302     064             .BYTE 302,064
2513  016150      340     064             .BYTE 340,064
2514  016152      300     064             .BYTE 300,064
2515  016154      340     064             .BYTE 340,064
2516  016156      300     064             .BYTE 300,064
2517
2518
2519
2520  016160      340     065             .BYTE 340,065           ;NUMBER '5'
2521  016162      300     065             .BYTE 300,065
2522  016164      364     065             .BYTE 364,065
2523  016166      312     065             .BYTE 312,065
2524  016170      344     065             .BYTE 344,065
2525  016172      300     065             .BYTE 300,065
2526  016174      344     065             .BYTE 344,065
2527  016176      300     065             .BYTE 300,065
2528  016200      364     065             .BYTE 364,065
2529  016202      312     065             .BYTE 312,065
2530  016204      340     065             .BYTE 340,065
2531  016206      310     065             .BYTE 310,065
2532  016210      344     065             .BYTE 344,065
2533  016212      310     065             .BYTE 310,065
2534  016214      350     065             .BYTE 350,065
2535  016216      302     065             .BYTE 302,065
2536  016220      340     065             .BYTE 340,065
2537  016222      300     065             .BYTE 300,065
2538  016224      340     065             .BYTE 340,065
```

```
2539  016226  300  065                .BYTE 300,065
2540
2541
2542  016230  340  001                .BYTE 340,01          ;END OF LINE
2543  016232  300  001                .BYTE 300,01
2544  016234  340  001                .BYTE 340,01
2545  016236  300  001                .BYTE 300,01
2546  016240  340  001                .BYTE 340,01
2547  016242  300  001                .BYTE 300,01
2548  016244  340  001                .BYTE 340,01
2549  016246  300  001                .BYTE 300,01
2550  016250  340  001                .BYTE 340,01
2551  016252  300  001                .BYTE 300,01
2552  016254  340  001                .BYTE 340,01
2553  016256  300  001                .BYTE 300,01
2554  016260  340  001                .BYTE 340,01
2555  016262  300  001                .BYTE 300,01
2556  016264  340  001                .BYTE 340,01
2557  016266  300  001                .BYTE 300,01
2558  016270  340  001                .BYTE 340,01
2559  016272  300  001                .BYTE 300,01
2560  016274  340  001                .BYTE 340,01
2561  016276  300  001                .BYTE 300,01
2562  016300  340  000                .BYTE 340,00          ;END OF SCREEN
2563  016302  300  000                .BYTE 300,00
2564  016304  340  000                .BYTE 340,00
2565  016306  300  000                .BYTE 300,00
2566  016310  340  000                .BYTE 340,00
2567  016312  300  000                .BYTE 300,00
2568  016314  340  000                .BYTE 340,00
2569  016316  300  000                .BYTE 300,00
2570  016320  340  000                .BYTE 340,00
2571  016322  300  000                .BYTE 300,00
2572  016324  340  000                .BYTE 340,00
2573  016326  300  000                .BYTE 300,00
2574  016330  340  000                .BYTE 340,00
2575  016332  300  000                .BYTE 300,00
2576  016334  340  000                .BYTE 340,00
2577  016336  300  000                .BYTE 300,00
2578  016340  340  000                .BYTE 340,00
2579  016342  300  000                .BYTE 300,00
2580  016344  340  000                .BYTE 340,00
2581  016346  300  000                .BYTE 300,00
2582  016350  377  377                .BYTE 377,377         ;CODE 377
2583  016352  337  377                .BYTE 337,377
2584  016354  377  377                .BYTE 377,377
2585  016356  337  377                .BYTE 337,377
2586  016360  377  377                .BYTE 377,377
2587  016362  337  377                .BYTE 337,377
2588  016364  377  377                .BYTE 377,377
2589  016366  337  377                .BYTE 337,377
2590  016370  377  377                .BYTE 377,377
2591  016372  337  377                .BYTE 337,377
2592  016374  377  377                .BYTE 377,377
2593  016376  337  377                .BYTE 337,377
2594  016400  377  377                .BYTE 377,377
```

```
2595  016402  337  377          .BYTE 337,377
2596  016404  377  377          .BYTE 377,377
2597  016406  337  377          .BYTE 337,377
2598  016410  377  377          .BYTE 377,377
2599  016412  337  377          .BYTE 337,377
2600  016414  377  377          .BYTE 377,377
2601  016416  337  377          .BYTE 337,377
2602  016420  340  202          .BYTE 340,202          ;END OF LINE
2603  016422  300  202          .BYTE 300,202
2604  016424  340  202          .BYTE 340,202
2605  016426  300  202          .BYTE 300,202
2606  016430  340  202          .BYTE 340,202
2607  016432  300  202          .BYTE 300,202
2608  016434  340  202          .BYTE 340,202
2609  016436  300  202          .BYTE 300,202
2610  016440  340  202          .BYTE 340,202
2611  016442  300  202          .BYTE 300,202
2612  016444  340  202          .BYTE 340,202
2613  016446  300  202          .BYTE 300,202
2614  016450  340  202          .BYTE 340,202
2615  016452  300  202          .BYTE 300,202
2616  016454  340  202          .BYTE 340,202
2617  016456  300  202          .BYTE 300,202
2618  016460  340  202          .BYTE 340,202
2619  016462  300  202          .BYTE 300,202
2620  016464  340  202          .BYTE 340,202
2621  016466  300  202          .BYTE 300,202
2622  016470  340  002          .BYTE 340,02           ;END OF LINE
2623  016472  300  002          .BYTE 300,02
2624  016474  340  002          .BYTE 340,02
2625  016476  300  002          .BYTE 300,02
2626  016500  340  002          .BYTE 340,02
2627  016502  300  002          .BYTE 300,02
2628  016504  340  002          .BYTE 340,02
2629  016506  300  002          .BYTE 300,02
2630  016510  340  002          .BYTE 340,02
2631  016512  300  002          .BYTE 300,02
2632  016514  340  002          .BYTE 340,02
2633  016516  300  002          .BYTE 300,02
2634  016520  340  002          .BYTE 340,02
2635  016522  300  002          .BYTE 300,02
2636  016524  340  002          .BYTE 340,02
2637  016526  300  002          .BYTE 300,02
2638  016530  340  002          .BYTE 340,02
2639  016532  300  002          .BYTE 300,02
2640  016534  340  002          .BYTE 340,02
2641  016536  300  002          .BYTE 300,02
2642  016540  340  003          .BYTE 340,03           ;END OF LINE
2643  016542  300  003          .BYTE 300,03
2644  016544  340  003          .BYTE 340,03
2645  016546  300  003          .BYTE 300,03
2646  016550  340  003          .BYTE 340,03
2647  016552  300  003          .BYTE 300,03
2648  016554  340  003          .BYTE 340,03
2649  016556  300  003          .BYTE 300,03
2650  016560  340  003          .BYTE 340,03
```

```
2651  016562  300  003                .BYTE 300,03
2652  016564  340  003                .BYTE 340,03
2653  016566  300  003                .BYTE 300,03
2654  016570  340  003                .BYTE 340,03
2655  016572  300  003                .BYTE 300,03
2656  016574  340  003                .BYTE 340,03
2657  016576  300  003                .BYTE 300,03
2658  016600  340  003                .BYTE 340,03
2659  016602  300  003                .BYTE 300,03
2660  016604  340  003                .BYTE 340,03
2661  016606  300  003                .BYTE 300,03
2662  016610  340  201                .BYTE 340,201          ;END OF LINE
2663  016612  300  201                .BYTE 300,201
2664  016614  340  201                .BYTE 340,201
2665  016616  300  201                .BYTE 300,201
2666  016620  340  201                .BYTE 340,201
2667  016622  300  201                .BYTE 300,201
2668  016624  340  201                .BYTE 340,201
2669  016626  300  201                .BYTE 300,201
2670  016630  340  201                .BYTE 340,201
2671  016632  300  201                .BYTE 300,201
2672  016634  340  201                .BYTE 340,201
2673  016636  300  201                .BYTE 300,201
2674  016640  340  201                .BYTE 340,201
2675  016642  300  201                .BYTE 300,201
2676  016644  340  201                .BYTE 340,201
2677  016646  300  201                .BYTE 300,201
2678  016650  340  201                .BYTE 340,201
2679  016652  300  201                .BYTE 300,201
2680  016654  340  201                .BYTE 340,201
2681  016656  300  201                .BYTE 300,201
2682  016660  340  203                .BYTE 340,203          ;END OF LINE
2683  016662  300  203                .BYTE 300,203
2684  016664  340  203                .BYTE 340,203
2685  016666  300  203                .BYTE 300,203
2686  016670  340  203                .BYTE 340,203
2687  016672  300  203                .BYTE 300,203
2688  016674  340  203                .BYTE 340,203
2689  016676  300  203                .BYTE 300,203
2690  016700  340  203                .BYTE 340,203
2691  016702  300  203                .BYTE 300,203
2692  016704  340  203                .BYTE 340,203
2693  016706  300  203                .BYTE 300,203
2694  016710  340  203                .BYTE 340,203
2695  016712  300  203                .BYTE 300,203
2696  016714  340  203                .BYTE 340,203
2697  016716  300  203                .BYTE 300,203
2698  016720  340  203                .BYTE 340,203
2699  016722  300  203                .BYTE 300,203
2700  016724  340  203                .BYTE 340,203
2701  016726  300  203                .BYTE 300,203
2702  016730  340  040  SPCHAR:       .BYTE 340,40           ;SPACE
2703  016732  300  040                .BYTE 300,40
2704  016734  340  040                .BYTE 340,40
2705  016736  300  040                .BYTE 300,40
2706  016740  340  040                .BYTE 340,40
```

```
2707  016742  300  040          .BYTE 300,40
2708  016744  340  040          .BYTE 340,40
2709  016746  300  040          .BYTE 300,40
2710  016750  340  040          .BYTE 340,40
2711  016752  300  040          .BYTE 300,40
2712  016754  340  040          .BYTE 340,40
2713  016756  300  040          .BYTE 300,40
2714  016760  340  040          .BYTE 340,40
2715  016762  300  040          .BYTE 300,40
2716  016764  340  040          .BYTE 340,40
2717  016766  300  040          .BYTE 300,40
2718  016770  340  040          .BYTE 340,40
2719  016772  300  040          .BYTE 300,40
2720  016774  340  040          .BYTE 340,40
2721  016776  300  040          .BYTE 300,40
2722  017000  340  200          .BYTE 340,200          ;END OF SCREEN
2723  017002  300  200          .BYTE 300,200
2724  017004  340  200          .BYTE 340,200
2725  017006  300  200          .BYTE 300,200
2726  017010  340  200          .BYTE 340,200
2727  017012  300  200          .BYTE 300,200
2728  017014  340  200          .BYTE 340,200
2729  017016  300  200          .BYTE 300,200
2730  017020  340  200          .BYTE 340,200
2731  017022  300  200          .BYTE 300,200
2732  017024  340  200          .BYTE 340,200
2733  017026  300  200          .BYTE 300,200
2734  017030  340  200          .BYTE 340,200
2735  017032  300  200          .BYTE 300,200
2736  017034  340  200          .BYTE 340,200
2737  017036  300  200          .BYTE 300,200
2738  017040  340  200          .BYTE 340,200
2739  017042  300  200          .BYTE 300,200
2740  017044  340  200          .BYTE 340,200
2741  017046  300  200          .BYTE 300,200
2742  017050  340  000          .BYTE 340,000          ;END OF SCREEN
2743  017052  300  000          .BYTE 300,000
2744  017054  340  000          .BYTE 340,000
2745  017056  300  000          .BYTE 300,000
2746  017060  340  000          .BYTE 340,000
2747  017062  300  000          .BYTE 300,000
2748  017064  340  000          .BYTE 340,000
2749  017066  300  000          .BYTE 300,000
2750  017070  340  000          .BYTE 340,000
2751  017072  300  000          .BYTE 300,000
2752  017074  340  000          .BYTE 340,000
2753  017076  300  000          .BYTE 300,000
2754  017100  340  000          .BYTE 340,000
2755  017102  300  000          .BYTE 300,000
2756  017104  340  000          .BYTE 340,000
2757  017106  300  000          .BYTE 300,000
2758  017110  340  000          .BYTE 340,000
2759  017112  300  000          .BYTE 300,000
2760  017114  340  000          .BYTE 340,000
2761  017116  300  000          .BYTE 300,000
2762  017120  340  040          .BYTE 340,40           ;SPACE
```

```
2763  017122  300  040          .BYTE 300,40
2764  017124  340  040          .BYTE 340,40
2765  017126  300  040          .BYTE 300,40
2766  017130  340  040          .BYTE 340,40
2767  017132  300  040          .BYTE 300,40
2768  017134  340  040          .BYTE 340,40
2769  017136  300  040          .BYTE 300,40
2770  017140  340  040          .BYTE 340,40
2771  017142  300  040          .BYTE 300,40
2772  017144  340  040          .BYTE 340,40
2773  017146  300  040          .BYTE 300,40
2774  017150  340  040          .BYTE 340,40
2775  017152  300  040          .BYTE 300,40
2776  017154  340  040          .BYTE 340,40
2777  017156  300  040          .BYTE 300,40
2778  017160  340  040          .BYTE 340,40
2779  017162  300  040          .BYTE 300,40
2780  017164  340  040          .BYTE 340,40
2781  017166  300  040          .BYTE 300,40
2782
2783
2784  017170  340  015          .BYTE 340,15          ;CARRIAGE RETURN
2785  017172  300  015          .BYTE 300,15
2786  017174  340  015          .BYTE 340,15
2787  017176  300  015          .BYTE 300,15
2788  017200  340  015          .BYTE 340,15
2789  017202  300  015          .BYTE 300,15
2790  017204  340  015          .BYTE 340,15
2791  017206  300  015          .BYTE 300,15
2792  017210  340  015          .BYTE 340,15
2793  017212  300  015          .BYTE 300,15
2794  017214  340  015          .BYTE 340,15
2795  017216  300  015          .BYTE 300,15
2796  017220  340  015          .BYTE 340,15
2797  017222  300  015          .BYTE 300,15
2798  017224  340  015          .BYTE 340,15
2799  017226  300  015          .BYTE 300,15
2800  017230  340  015          .BYTE 340,15
2801  017232  300  015          .BYTE 300,15
2802  017234  340  015          .BYTE 340,15
2803  017236  300  015          .BYTE 300,15
2804
2805
2806  017240  340  012          .BYTE 340,12          ;LINE FEED
2807  017242  300  012          .BYTE 300,12
2808  017244  340  012          .BYTE 340,12
2809  017246  300  012          .BYTE 300,12
2810  017250  340  012          .BYTE 340,12
2811  017252  300  012          .BYTE 300,12
2812  017254  340  012          .BYTE 340,12
2813  017256  300  012          .BYTE 300,12
2814  017260  340  012          .BYTE 340,12
2815  017262  300  012          .BYTE 300,12
2816  017264  340  012          .BYTE 340,12
2817  017266  300  012          .BYTE 300,12
2818  017270  340  012          .BYTE 340,12
```

```
2818  017272  300  012                .BYTE 300,12
2819  017274  340  012                .BYTE 340,12
2820  017276  300  012                .BYTE 300,12
2821  017300  340  012                .BYTE 340,12
2822  017302  300  012                .BYTE 300,12
2823  017304  340  012                .BYTE 340,12
2824  017306  300  012                .BYTE 300,12
2825
2826
2827  017310  340  066                .BYTE 340,066          ;NUMBER '6'
2828  017312  300  066                .BYTE 300,066
2829  017314  350  066                .BYTE 350,066
2830  017316  305  066                .BYTE 305,066
2831  017320  344  066                .BYTE 344,066
2832  017322  300  066                .BYTE 300,066
2833  017324  344  066                .BYTE 344,066
2834  017326  300  066                .BYTE 300,066
2835  017330  364  066                .BYTE 364,066
2836  017332  312  066                .BYTE 312,066
2837  017334  344  066                .BYTE 344,066
2838  017336  310  066                .BYTE 310,066
2839  017340  344  066                .BYTE 344,066
2840  017342  310  066                .BYTE 310,066
2841  017344  350  066                .BYTE 350,066
2842  017346  305  066                .BYTE 305,066
2843  017350  340  066                .BYTE 340,066
2844  017352  300  066                .BYTE 300,066
2845  017354  340  066                .BYTE 340,066
2846  017356  300  066                .BYTE 300,066
2847
2848
2849  017360  340  067                .BYTE 340,067          ;NUMBER '7'
2850  017362  300  067                .BYTE 300,067
2851  017364  364  067                .BYTE 364,067
2852  017366  312  067                .BYTE 312,067
2853  017370  340  067                .BYTE 340,067
2854  017372  310  067                .BYTE 310,067
2855  017374  340  067                .BYTE 340,067
2856  017376  304  067                .BYTE 304,067
2857  017400  340  067                .BYTE 340,067
2858  017402  302  067                .BYTE 302,067
2859  017404  340  067                .BYTE 340,067
2860  017406  301  067                .BYTE 301,067
2861  017410  360  067                .BYTE 360,067
2862  017412  300  067                .BYTE 300,067
2863  017414  350  067                .BYTE 350,067
2864  017416  300  067                .BYTE 300,067
2865  017420  340  067                .BYTE 340,067
2866  017422  300  067                .BYTE 300,067
2867  017424  340  067                .BYTE 340,067
2868  017426  300  067                .BYTE 300,067
2869
2870
2871  017430  340  070                .BYTE 340,070          ;NUMBER '8'
2872  017432  300  070                .BYTE 300,070
2873  017434  350  070                .BYTE 350,070
```

```
2875  017436  305  070                .BYTE 305,070
2876  017440  344  070                .BYTE 344,070
2877  017442  310  070                .BYTE 310,070
2878  017444  344  070                .BYTE 344,070
2879  017446  310  070                .BYTE 310,070
2880  017450  350  070                .BYTE 350,070
2881  017452  305  070                .BYTE 305,070
2882  017454  344  070                .BYTE 344,070
2883  017456  310  070                .BYTE 310,070
2884  017460  344  070                .BYTE 344,070
2885  017462  310  070                .BYTE 310,070
2886  017464  350  070                .BYTE 350,070
2887  017466  305  070                .BYTE 305,070
2888  017470  340  070                .BYTE 340,070
2889  017472  300  070                .BYTE 300,070
2890  017474  340  070                .BYTE 340,070
2891  017476  300  070                .BYTE 300,070
2892
2893
2894  017500  340  071                .BYTE 340,071          ;NUMBER '9'
2895  017502  300  071                .BYTE 300,071
2896  017504  350  071                .BYTE 350,071
2897  017506  305  071                .BYTE 305,071
2898  017510  344  071                .BYTE 344,071
2899  017512  310  071                .BYTE 310,071
2900  017514  344  071                .BYTE 344,071
2901  017516  310  071                .BYTE 310,071
2902  017520  364  071                .BYTE 364,071
2903  017522  312  071                .BYTE 312,071
2904  017524  340  071                .BYTE 340,071
2905  017526  310  071                .BYTE 310,071
2906  017530  340  071                .BYTE 340,071
2907  017532  310  071                .BYTE 310,071
2908  017534  360  071                .BYTE 360,071
2909  017536  305  071                .BYTE 305,071
2910  017540  340  071                .BYTE 340,071
2911  017542  300  071                .BYTE 300,071
2912  017544  340  071                .BYTE 340,071
2913  017546  300  071                .BYTE 300,071
2914
2915
2916  017550  340  060                .BYTE 340,060          ;NUMBER '0'
2917  017552  300  060                .BYTE 300,060
2918  017554  360  060                .BYTE 360,060
2919  017556  305  060                .BYTE 305,060
2920  017560  344  060                .BYTE 344,060
2921  017562  314  060                .BYTE 314,060
2922  017564  344  060                .BYTE 344,060
2923  017566  312  060                .BYTE 312,060
2924  017570  344  060                .BYTE 344,060
2925  017572  311  060                .BYTE 311,060
2926  017574  364  060                .BYTE 364,060
2927  017576  310  060                .BYTE 310,060
2928  017600  354  060                .BYTE 354,060
2929  017602  310  060                .BYTE 310,060
2930  017604  350  060                .BYTE 350,060
```

```
2931  017606  305  060          .BYTE 305,060
2932  017610  340  060          .BYTE 340,060
2933  017612  300  060          .BYTE 300,060
2934  017614  340  060          .BYTE 340,060
2935  017616  300  060          .BYTE 300,060
2936
2937
2938
2939
2940  017620  340  043          .BYTE 340,043          ;SPECIAL CHARACTER '8'
2941  017622  300  043          .BYTE 300,043
2942  017624  360  043          .BYTE 360,043
2943  017626  304  043          .BYTE 304,043
2944  017630  360  043          .BYTE 360,043
2945  017632  304  043          .BYTE 304,043
2946  017634  364  043          .BYTE 364,043
2947  017636  315  043          .BYTE 315,043
2948  017640  360  043          .BYTE 360,043
2949  017642  304  043          .BYTE 304,043
2950  017644  364  043          .BYTE 364,043
2951  017646  315  043          .BYTE 315,043
2952  017650  360  043          .BYTE 360,043
2953  017652  304  043          .BYTE 304,043
2954  017654  360  043          .BYTE 360,043
2955  017656  304  043          .BYTE 304,043
2956  017660  340  043          .BYTE 340,043
2957  017662  300  043          .BYTE 300,043
2958  017664  340  043          .BYTE 340,043
2959  017666  300  043          .BYTE 300,043
2960
2961
2962
2963  017670  340  101          .BYTE 340,101          ;UPPER CASE 'A'
2964  017672  300  101          .BYTE 300,101
2965  017674  360  101          .BYTE 360,101
2966  017676  302  101          .BYTE 302,101
2967  017700  350  101          .BYTE 350,101
2968  017702  304  101          .BYTE 304,101
2969  017704  344  101          .BYTE 344,101
2970  017706  310  101          .BYTE 310,101
2971  017710  344  101          .BYTE 344,101
2972  017712  310  101          .BYTE 310,101
2973  017714  364  101          .BYTE 364,101
2974  017716  312  101          .BYTE 312,101
2975  017720  344  101          .BYTE 344,101
2976  017722  310  101          .BYTE 310,101
2977  017724  344  101          .BYTE 344,101
2978  017726  310  101          .BYTE 310,101
2979  017730  340  101          .BYTE 340,101
2980  017732  300  101          .BYTE 300,101
2981  017734  340  101          .BYTE 340,101
2982  017736  300  101          .BYTE 300,101
2983
2984
2985  017740  340  102          .BYTE 340,102          ;UPPER CASE 'B'
2986  017742  300  102          .BYTE 300,102
```

```
 2987  017744  352  102          .BYTE 352,102
 2988  017746  305  102          .BYTE 305,102
 2989  017750  344  102          .BYTE 344,102
 2990  017752  310  102          .BYTE 310,102
 2991  017754  344  102          .BYTE 344,102
 2992  017756  310  102          .BYTE 310,102
 2993  017760  354  102          .BYTE 354,102
 2994  017762  305  102          .BYTE 305,102
 2995  017764  344  102          .BYTE 344,102
 2996  017766  310  102          .BYTE 310,102
 2997  017770  344  102          .BYTE 344,102
 2998  017772  310  102          .BYTE 310,102
 2999  017774  352  102          .BYTE 352,102
 3000  017776  305  102          .BYTE 305,102
 3001  020000  340  102          .BYTE 340,102
 3002  020002  300  102          .BYTE 300,102
 3003  020004  340  102          .BYTE 340,102
 3004  020006  300  102          .BYTE 300,102
 3005
 3006
 3007  020010  340  103          .BYTE 340,103          ;UPPER CASE 'C'
 3008  020012  300  103          .BYTE 300,103
 3009  020014  350  103          .BYTE 350,103
 3010  020016  305  103          .BYTE 305,103
 3011  020020  344  103          .BYTE 344,103
 3012  020022  310  103          .BYTE 310,103
 3013  020024  344  103          .BYTE 344,103
 3014  020026  300  103          .BYTE 300,103
 3015  020030  344  103          .BYTE 344,103
 3016  020032  300  103          .BYTE 300,103
 3017  020034  344  103          .BYTE 344,103
 3018  020036  300  103          .BYTE 300,103
 3019  020040  344  103          .BYTE 344,103
 3020  020042  310  103          .BYTE 310,103
 3021  020044  350  103          .BYTE 350,103
 3022  020046  305  103          .BYTE 305,103
 3023  020050  340  103          .BYTE 340,103
 3024  020052  300  103          .BYTE 300,103
 3025  020054  340  103          .BYTE 340,103
 3026  020056  300  103          .BYTE 300,103
 3027
 3028
 3029  020060  340  104          .BYTE 340,104          ;UPPER CASE 'D'
 3030  020062  300  104          .BYTE 300,104
 3031  020064  352  104          .BYTE 352,104
 3032  020066  305  104          .BYTE 305,104
 3033  020070  344  104          .BYTE 344,104
 3034  020072  310  104          .BYTE 310,104
 3035  020074  344  104          .BYTE 344,104
 3036  020076  310  104          .BYTE 310,104
 3037  020100  344  104          .BYTE 344,104
 3038  020102  310  104          .BYTE 310,104
 3039  020104  344  104          .BYTE 344,104
 3040  020106  310  104          .BYTE 310,104
 3041  020110  344  104          .BYTE 344,104
 3042  020112  310  104          .BYTE 310,104
```

```
3043  020114  352  104                .BYTE 352,104
3044  020116  305  104                .BYTE 305,104
3045  020120  340  104                .BYTE 340,104
3046  020122  300  104                .BYTE 300,104
3047  020124  340  104                .BYTE 340,104
3048  020126  300  104                .BYTE 300,104
3049
3050
3051  020130  340  105                .BYTE 340,105          ;UPPER CASE 'E'
3052  020132  300  105                .BYTE 300,105
3053  020134  364  105                .BYTE 364,105
3054  020136  312  105                .BYTE 312,105
3055  020140  344  105                .BYTE 344,105
3056  020142  300  105                .BYTE 300,105
3057  020144  344  105                .BYTE 344,105
3058  020146  300  105                .BYTE 300,105
3059  020150  364  105                .BYTE 364,105
3060  020152  302  105                .BYTE 302,105
3061  020154  344  105                .BYTE 344,105
3062  020156  300  105                .BYTE 300,105
3063  020160  344  105                .BYTE 344,105
3064  020162  300  105                .BYTE 300,105
3065  020164  364  105                .BYTE 364,105
3066  020166  312  105                .BYTE 312,105
3067  020170  340  105                .BYTE 340,105
3068  020172  300  105                .BYTE 300,105
3069  020174  340  105                .BYTE 340,105
3070  020176  300  105                .BYTE 300,105
3071
3072
3073  020200  340  106                .BYTE 340,106          ;UPPER CASE 'F'
3074  020202  300  106                .BYTE 300,106
3075  020204  364  106                .BYTE 364,106
3076  020206  312  106                .BYTE 312,106
3077  020210  344  106                .BYTE 344,106
3078  020212  300  106                .BYTE 300,106
3079  020214  344  106                .BYTE 344,106
3080  020216  300  106                .BYTE 300,106
3081  020220  364  106                .BYTE 364,106
3082  020222  302  106                .BYTE 302,106
3083  020224  344  106                .BYTE 344,106
3084  020226  300  106                .BYTE 300,106
3085  020230  344  106                .BYTE 344,106
3086  020232  300  106                .BYTE 300,106
3087  020234  344  106                .BYTE 344,106
3088  020236  300  106                .BYTE 300,106
3089  020240  340  106                .BYTE 340,106
3090  020242  300  106                .BYTE 300,106
3091  020244  340  106                .BYTE 340,106
3092  020246  300  106                .BYTE 300,106
3093
3094
3095  020250  340  107                .BYTE 340,107          ;UPPER CASE 'G'
3096  020252  300  107                .BYTE 300,107
3097  020254  350  107                .BYTE 350,107
3098  020256  305  107                .BYTE 305,107
```

```
3099  020262  344  107          .BYTE 344,107
3100  020262  310  107          .BYTE 310,107
3101  020264  344  107          .BYTE 344,107
3102  020266  300  107          .BYTE 300,107
3103  020270  344  107          .BYTE 344,107
3104  020272  300  107          .BYTE 300,107
3105  020274  344  107          .BYTE 344,107
3106  020276  312  107          .BYTE 312,107
3107  020300  344  107          .BYTE 344,107
3108  020302  310  107          .BYTE 310,107
3109  020304  350  107          .BYTE 350,107
3110  020306  305  107          .BYTE 305,107
3111  020310  340  107          .BYTE 340,107
3112  020312  300  107          .BYTE 300,107
3113  020314  340  107          .BYTE 340,107
3114  020316  300  107          .BYTE 300,107
3115
3116
3117  020320  340  110          .BYTE 340,110          ;UPPER CASE 'H'
3118  020322  300  110          .BYTE 300,110
3119  020324  344  110          .BYTE 344,110
3120  020326  310  110          .BYTE 310,110
3121  020330  344  110          .BYTE 344,110
3122  020332  310  110          .BYTE 310,110
3123  020334  344  110          .BYTE 344,110
3124  020336  310  110          .BYTE 310,110
3125  020340  364  110          .BYTE 364,110
3126  020342  312  110          .BYTE 312,110
3127  020344  344  110          .BYTE 344,110
3128  020346  310  110          .BYTE 310,110
3129  020350  344  110          .BYTE 344,110
3130  020352  310  110          .BYTE 310,110
3131  020354  344  110          .BYTE 344,110
3132  020356  310  110          .BYTE 310,110
3133  020360  340  110          .BYTE 340,110
3134  020362  300  110          .BYTE 300,110
3135  020364  340  110          .BYTE 340,110
3136  020366  300  110          .BYTE 300,110
3137
3138
3139  020370  340  111          .BYTE 340,111          ;UPPER CASE 'I'
3140  020372  300  111          .BYTE 300,111
3141  020374  364  111          .BYTE 364,111
3142  020376  312  111          .BYTE 312,111
3143  020400  340  111          .BYTE 340,111
3144  020402  301  111          .BYTE 301,111
3145  020404  340  111          .BYTE 340,111
3146  020406  301  111          .BYTE 301,111
3147  020410  340  111          .BYTE 340,111
3148  020412  301  111          .BYTE 301,111
3149  020414  340  111          .BYTE 340,111
3150  020416  301  111          .BYTE 301,111
3151  020420  340  111          .BYTE 340,111
3152  020422  301  111          .BYTE 301,111
3153  020424  364  111          .BYTE 364,111
3154  020426  312  111          .BYTE 312,111
```

```
3155  020430  340  111          .BYTE 340,111
3156  020432  300  111          .BYTE 300,111
3157  020434  340  111          .BYTE 340,111
3158  020436  300  111          .BYTE 300,111
3159
3160
3161  020440  340  112          .BYTE 340,112          ;UPPER CASE 'J'
3162  020442  300  112          .BYTE 300,112
3163  020444  364  112          .BYTE 364,112
3164  020446  312  112          .BYTE 312,112
3165  020450  340  112          .BYTE 340,112
3166  020452  302  112          .BYTE 302,112
3167  020454  340  112          .BYTE 340,112
3168  020456  302  112          .BYTE 302,112
3169  020460  340  112          .BYTE 340,112
3170  020462  302  112          .BYTE 302,112
3171  020464  340  112          .BYTE 340,112
3172  020466  302  112          .BYTE 302,112
3173  020470  344  112          .BYTE 344,112
3174  020472  302  112          .BYTE 302,112
3175  020474  350  112          .BYTE 350,112
3176  020476  301  112          .BYTE 301,112
3177  020500  340  112          .BYTE 340,112
3178  020502  300  112          .BYTE 300,112
3179  020504  340  112          .BYTE 340,112
3180  020506  300  112          .BYTE 300,112
3181
3182
3183  020510  340  113          .BYTE 340,113          ;UPPER CASE 'K'
3184  020512  300  113          .BYTE 300,113
3185  020514  344  113          .BYTE 344,113
3186  020516  310  113          .BYTE 310,113
3187  020520  344  113          .BYTE 344,113
3188  020522  304  113          .BYTE 304,113
3189  020524  344  113          .BYTE 344,113
3190  020526  302  113          .BYTE 302,113
3191  020530  354  113          .BYTE 354,113
3192  020532  301  113          .BYTE 301,113
3193  020534  344  113          .BYTE 344,113
3194  020536  302  113          .BYTE 302,113
3195  020540  344  113          .BYTE 344,113
3196  020542  304  113          .BYTE 304,113
3197  020544  344  113          .BYTE 344,113
3198  020546  310  113          .BYTE 310,113
3199  020550  340  113          .BYTE 340,113
3200  020552  300  113          .BYTE 300,113
3201  020554  340  113          .BYTE 340,113
3202  020556  300  113          .BYTE 300,113
3203
3204
3205  020560  340  114          .BYTE 340,114          ;UPPER CASE 'L'
3206  020562  300  114          .BYTE 300,114
3207  020564  344  114          .BYTE 344,114
3208  020566  300  114          .BYTE 300,114
3209  020570  344  114          .BYTE 344,114
3210  020572  300  114          .BYTE 300,114
```

```
3211  020574  344  114          .BYTE 344,114
3212  020576  300  114          .BYTE 300,114
3213  020600  344  114          .BYTE 344,114
3214  020602  300  114          .BYTE 300,114
3215  020604  344  114          .BYTE 344,114
3216  020606  300  114          .BYTE 300,114
3217  020610  344  114          .BYTE 344,114
3218  020612  300  114          .BYTE 300,114
3219  020614  354  114          .BYTE 354,114
3220  020616  305  114          .BYTE 305,114
3221  020620  340  114          .BYTE 340,114
3222  020622  300  114          .BYTE 300,114
3223  020624  340  114          .BYTE 340,114
3224  020626  300  114          .BYTE 300,114
3225
3226
3227  020630  340  115          .BYTE 340,115          ;UPPER CASE "M"
3228  020632  300  115          .BYTE 300,115
3229  020634  344  115          .BYTE 344,115
3230  020636  310  115          .BYTE 310,115
3231  020640  354  115          .BYTE 354,115
3232  020642  314  115          .BYTE 314,115
3233  020644  364  115          .BYTE 364,115
3234  020646  312  115          .BYTE 312,115
3235  020650  344  115          .BYTE 344,115
3236  020652  311  115          .BYTE 311,115
3237  020654  344  115          .BYTE 344,115
3238  020656  310  115          .BYTE 310,115
3239  020660  344  115          .BYTE 344,115
3240  020662  310  115          .BYTE 310,115
3241  020664  344  115          .BYTE 344,115
3242  020666  310  115          .BYTE 310,115
3243  020670  340  115          .BYTE 340,115
3244  020672  300  115          .BYTE 300,115
3245  020674  340  115          .BYTE 340,115
3246  020676  300  115          .BYTE 300,115
3247
3248
3249  020700  340  116          .BYTE 340,116          ;UPPER CASE " N "
3250  020702  300  116          .BYTE 300,116
3251  020704  344  116          .BYTE 344,116
3252  020706  310  116          .BYTE 310,116
3253  020710  354  116          .BYTE 354,116
3254  020712  310  116          .BYTE 310,116
3255  020714  364  116          .BYTE 364,116
3256  020716  310  116          .BYTE 310,116
3257  020720  344  116          .BYTE 344,116
3258  020722  311  116          .BYTE 311,116
3259  020724  344  116          .BYTE 344,116
3260  020726  312  116          .BYTE 312,116
3261  020730  344  116          .BYTE 344,116
3262  020732  314  116          .BYTE 314,116
3263  020734  344  116          .BYTE 344,116
3264  020736  310  116          .BYTE 310,116
3265  020740  340  116          .BYTE 340,116
3266  020742  300  116          .BYTE 300,116
```

```
3267  020744  340  116          .BYTE 340,116
3268  020746  300  116          .BYTE 300,116
3269
3270
3271  020750  340  117          .BYTE 340,117          ;UPPER CASE 'O'
3272  020752  300  117          .BYTE 300,117
3273  020754  350  117          .BYTE 350,117
3274  020756  305  117          .BYTE 305,117
3275  020760  344  117          .BYTE 344,117
3276  020762  310  117          .BYTE 310,117
3277  020764  344  117          .BYTE 344,117
3278  020766  310  117          .BYTE 310,117
3279  020770  344  117          .BYTE 344,117
3280  020772  310  117          .BYTE 310,117
3281  020774  344  117          .BYTE 344,117
3282  020776  310  117          .BYTE 310,117
3283  021000  344  117          .BYTE 344,117
3284  021002  310  117          .BYTE 310,117
3285  021004  350  117          .BYTE 350,117
3286  021006  305  117          .BYTE 305,117
3287  021010  340  117          .BYTE 340,117
3288  021012  300  117          .BYTE 300,117
3289  021014  340  117          .BYTE 340,117
3290  021016  300  117          .BYTE 300,117
3291
3292
3293  021020  340  120          .BYTE 340,120          ;UPPER CASE 'P'
3294  021022  300  120          .BYTE 300,120
3295  021024  354  120          .BYTE 354,120
3296  021026  305  120          .BYTE 305,120
3297  021030  344  120          .BYTE 344,120
3298  021032  310  120          .BYTE 310,120
3299  021034  344  120          .BYTE 344,120
3300  021036  310  120          .BYTE 310,120
3301  021040  354  120          .BYTE 354,120
3302  021042  305  120          .BYTE 305,120
3303  021044  344  120          .BYTE 344,120
3304  021046  300  120          .BYTE 300,120
3305  021050  344  120          .BYTE 344,120
3306  021052  300  120          .BYTE 300,120
3307  021054  344  120          .BYTE 344,120
3308  021056  300  120          .BYTE 300,120
3309  021060  340  120          .BYTE 340,120
3310  021062  300  120          .BYTE 300,120
3311  021064  340  120          .BYTE 340,120
3312  021066  300  120          .BYTE 300,120
3313
3314
3315  021070  340  121          .BYTE 340,121          ;UPPER CASE 'Q'
3316  021072  300  121          .BYTE 300,121
3317  021074  350  121          .BYTE 350,121
3318  021076  305  121          .BYTE 305,121
3319  021100  344  121          .BYTE 344,121
3320  021102  310  121          .BYTE 310,121
3321  021104  344  121          .BYTE 344,121
3322  021106  310  121          .BYTE 310,121
```

```
3323  021110  344  121          .BYTE 344,121
3324  021112  312  121          .BYTE 312,121
3325  021114  344  121          .BYTE 344,121
3326  021116  314  121          .BYTE 314,121
3327  021120  344  121          .BYTE 344,121
3328  021122  310  121          .BYTE 310,121
3329  021124  350  121          .BYTE 350,121
3330  021126  325  121          .BYTE 325,121
3331  021130  340  121          .BYTE 340,121
3332  021132  300  121          .BYTE 300,121
3333  021134  340  121          .BYTE 340,121
3334  021136  300  121          .BYTE 300,121
3335
3336
3337  021140  340  122          .BYTE 340,122          ;UPPER CASE 'R'
3338  021142  300  122          .BYTE 300,122
3339  021144  354  122          .BYTE 354,122
3340  021146  305  122          .BYTE 305,122
3341  021150  344  122          .BYTE 344,122
3342  021152  310  122          .BYTE 310,122
3343  021154  344  122          .BYTE 344,122
3344  021156  310  122          .BYTE 310,122
3345  021160  354  122          .BYTE 354,122
3346  021162  305  122          .BYTE 305,122
3347  021164  344  122          .BYTE 344,122
3348  021166  302  122          .BYTE 302,122
3349  021170  344  122          .BYTE 344,122
3350  021172  304  122          .BYTE 304,122
3351  021174  344  122          .BYTE 344,122
3352  021176  310  122          .BYTE 310,122
3353  021200  340  122          .BYTE 340,122
3354  021202  300  122          .BYTE 300,122
3355  021204  340  122          .BYTE 340,122
3356  021206  300  122          .BYTE 300,122
3357
3358
3359  021210  340  123          .BYTE 340,123          ;UPPER CASE 'S'
3360  021212  300  123          .BYTE 300,123
3361  021214  350  123          .BYTE 350,123
3362  021216  305  123          .BYTE 305,123
3363  021220  344  123          .BYTE 344,123
3364  021222  310  123          .BYTE 310,123
3365  021224  344  123          .BYTE 344,123
3366  021226  300  123          .BYTE 300,123
3367  021230  350  123          .BYTE 350,123
3368  021232  305  123          .BYTE 305,123
3369  021234  340  123          .BYTE 340,123
3370  021236  310  123          .BYTE 310,123
3371  021240  344  123          .BYTE 344,123
3372  021242  310  123          .BYTE 310,123
3373  021244  350  123          .BYTE 350,123
3374  021246  305  123          .BYTE 305,123
3375  021250  340  123          .BYTE 340,123
3376  021252  300  123          .BYTE 300,123
3377  021254  340  123          .BYTE 340,123
3378  021256  300  123          .BYTE 300,123
```

```
3379
3380
3381  021260  340  124  .BYTE 340,124  ;UPPER CASE 'T'
3382  021262  300  124  .BYTE 300,124
3383  021264  364  124  .BYTE 364,124
3384  021266  312  124  .BYTE 312,124
3385  021270  340  124  .BYTE 340,124
3386  021272  301  124  .BYTE 301,124
3387  021274  340  124  .BYTE 340,124
3388  021276  301  124  .BYTE 301,124
3389  021300  340  124  .BYTE 340,124
3390  021302  301  124  .BYTE 301,124
3391  021304  340  124  .BYTE 340,124
3392  021306  301  124  .BYTE 301,124
3393  021310  340  124  .BYTE 340,124
3394  021312  301  124  .BYTE 301,124
3395  021314  340  124  .BYTE 340,124
3396  021316  301  124  .BYTE 301,124
3397  021320  340  124  .BYTE 340,124
3398  021322  300  124  .BYTE 300,124
3399  021324  340  124  .BYTE 340,124
3400  021326  300  124  .BYTE 300,124
3401
3402
3403  021330  340  125  .BYTE 340,125  ;UPPER CASE 'U'
3404  021332  300  125  .BYTE 300,125
3405  021334  344  125  .BYTE 344,125
3406  021336  310  125  .BYTE 310,125
3407  021340  344  125  .BYTE 344,125
3408  021342  310  125  .BYTE 310,125
3409  021344  344  125  .BYTE 344,125
3410  021346  310  125  .BYTE 310,125
3411  021350  344  125  .BYTE 344,125
3412  021352  310  125  .BYTE 310,125
3413  021354  344  125  .BYTE 344,125
3414  021356  310  125  .BYTE 310,125
3415  021360  344  125  .BYTE 344,125
3416  021362  310  125  .BYTE 310,125
3417  021364  350  125  .BYTE 350,125
3418  021366  305  125  .BYTE 305,125
3419  021370  340  125  .BYTE 340,125
3420  021372  300  125  .BYTE 300,125
3421  021374  340  125  .BYTE 340,125
3422  021376  300  125  .BYTE 300,125
3423
3424
3425  021400  340  126  .BYTE 340,126  ;UPPER CASE 'V'
3426  021402  300  126  .BYTE 300,126
3427  021404  344  126  .BYTE 344,126
3428  021406  310  126  .BYTE 310,126
3429  021410  344  126  .BYTE 344,126
3430  021412  310  126  .BYTE 310,126
3431  021414  350  126  .BYTE 350,126
3432  021416  304  126  .BYTE 304,126
3433  021420  350  126  .BYTE 350,126
3434  021422  304  126  .BYTE 304,126
```

```
3435  021424  360  126  .BYTE 360,126
3436  021426  302  126  .BYTE 302,126
3437  021430  340  126  .BYTE 340,126
3438  021432  301  126  .BYTE 301,126
3439  021434  340  126  .BYTE 340,126
3440  021436  301  126  .BYTE 301,126
3441  021440  340  126  .BYTE 340,126
3442  021442  300  126  .BYTE 300,126
3443  021444  340  126  .BYTE 340,126
3444  021446  300  126  .BYTE 300,126
3445
3446
3447  021450  340  127  .BYTE 340,127          ;UPPER CASE 'W'
3448  021452  300  127  .BYTE 300,127
3449  021454  344  127  .BYTE 344,127
3450  021456  310  127  .BYTE 310,127
3451  021460  344  127  .BYTE 344,127
3452  021462  310  127  .BYTE 310,127
3453  021464  344  127  .BYTE 344,127
3454  021466  310  127  .BYTE 310,127
3455  021470  344  127  .BYTE 344,127
3456  021472  311  127  .BYTE 311,127
3457  021474  344  127  .BYTE 344,127
3458  021476  311  127  .BYTE 311,127
3459  021500  364  127  .BYTE 364,127
3460  021502  312  127  .BYTE 312,127
3461  021504  350  127  .BYTE 350,127
3462  021506  304  127  .BYTE 304,127
3463  021510  340  127  .BYTE 340,127
3464  021512  300  127  .BYTE 300,127
3465  021514  340  127  .BYTE 340,127
3466  021516  300  127  .BYTE 300,127
3467
3468
3469  021520  340  130  .BYTE 340,130          ;UPPER CASE 'X'
3470  021522  300  130  .BYTE 300,130
3471  021524  344  130  .BYTE 344,130
3472  021526  310  130  .BYTE 310,130
3473  021530  350  130  .BYTE 350,130
3474  021532  304  130  .BYTE 304,130
3475  021534  360  130  .BYTE 360,130
3476  021536  302  130  .BYTE 302,130
3477  021540  340  130  .BYTE 340,130
3478  021542  301  130  .BYTE 301,130
3479  021544  360  130  .BYTE 360,130
3480  021546  302  130  .BYTE 302,130
3481  021550  350  130  .BYTE 350,130
3482  021552  304  130  .BYTE 304,130
3483  021554  344  130  .BYTE 344,130
3484  021556  310  130  .BYTE 310,130
3485  021560  340  130  .BYTE 340,130
3486  021562  300  130  .BYTE 300,130
3487  021564  340  130  .BYTE 340,130
3488  021566  300  130  .BYTE 300,130
3489
3490
```

```
3491  021570  340  131  .BYTE 340,131  ;UPPER CASE 'Y'
3492  021572  300  131  .BYTE 300,131
3493  021574  344  131  .BYTE 344,131
3494  021576  310  131  .BYTE 310,131
3495  021600  350  131  .BYTE 350,131
3496  021602  304  131  .BYTE 304,131
3497  021604  360  131  .BYTE 360,131
3498  021606  302  131  .BYTE 302,131
3499  021610  340  131  .BYTE 340,131
3500  021612  301  131  .BYTE 301,131
3501  021614  340  131  .BYTE 340,131
3502  021616  301  131  .BYTE 301,131
3503  021620  340  131  .BYTE 340,131
3504  021622  301  131  .BYTE 301,131
3505  021624  340  131  .BYTE 340,131
3506  021626  301  131  .BYTE 301,131
3507  021630  340  131  .BYTE 340,131
3508  021632  300  131  .BYTE 300,131
3509  021634  340  131  .BYTE 340,131
3510  021636  300  131  .BYTE 300,131
3511
3512
3513  021640  340  132  .BYTE 340,132  ;UPPER CASE 'Z'
3514  021642  300  132  .BYTE 300,132
3515  021644  364  132  .BYTE 364,132
3516  021646  312  132  .BYTE 312,132
3517  021650  340  132  .BYTE 340,132
3518  021652  304  132  .BYTE 304,132
3519  021654  340  132  .BYTE 340,132
3520  021656  302  132  .BYTE 302,132
3521  021660  340  132  .BYTE 340,132
3522  021662  301  132  .BYTE 301,132
3523  021664  360  132  .BYTE 360,132
3524  021666  300  132  .BYTE 300,132
3525  021670  350  132  .BYTE 350,132
3526  021672  300  132  .BYTE 300,132
3527  021674  364  132  .BYTE 364,132
3528  021676  312  132  .BYTE 312,132
3529  021700  340  132  .BYTE 340,132
3530  021702  300  132  .BYTE 300,132
3531  021704  340  132  .BYTE 340,132
3532  021706  300  132  .BYTE 300,132
3533
3534
3535  021710  340  061  .BYTE 340,061  ;NUMBER '1'
3536  021712  300  061  .BYTE 300,061
3537  021714  340  061  .BYTE 340,061
3538  021716  301  061  .BYTE 301,061
3539  021720  360  061  .BYTE 360,061
3540  021722  301  061  .BYTE 301,061
3541  021724  350  061  .BYTE 350,061
3542  021726  301  061  .BYTE 301,061
3543  021730  340  061  .BYTE 340,061
3544  021732  301  061  .BYTE 301,061
3545  021734  340  061  .BYTE 340,061
3546  021736  301  061  .BYTE 301,061
```

```
3547  021740  340  061          .BYTE 340,061
3548  021742  301  061          .BYTE 301,061
3549  021744  350  061          .BYTE 350,061
3550  021746  305  061          .BYTE 305,061
3551  021750  340  061          .BYTE 340,061
3552  021752  300  061          .BYTE 300,061
3553  021754  340  061          .BYTE 340,061
3554  021756  300  061          .BYTE 300,061
3555
3556
3557  021760  340  062          .BYTE 340,062          ;NUMBER "2"
3558  021762  300  062          .BYTE 300,062
3559  021764  350  062          .BYTE 350,062
3560  021766  305  062          .BYTE 305,062
3561  021770  344  062          .BYTE 344,062
3562  021772  310  062          .BYTE 310,062
3563  021774  340  062          .BYTE 340,062
3564  021776  310  062          .BYTE 310,062
3565  022000  360  062          .BYTE 360,062
3566  022002  312  062          .BYTE 312,062
3567  022004  350  062          .BYTE 350,062
3568  022006  300  062          .BYTE 300,062
3569  022010  344  062          .BYTE 344,062
3570  022012  300  062          .BYTE 300,062
3571  022014  364  062          .BYTE 364,062
3572  022016  312  062          .BYTE 312,062
3573  022020  340  062          .BYTE 340,062
3574  022022  300  062          .BYTE 300,062
3575  022024  340  062          .BYTE 340,062
3576  022026  300  062          .BYTE 300,062
3577
3578
3579  022030  340  042          .BYTE   340,042        ;DIAMOND
3580  022032  300  042          .BYTE   300,042
3581  022034  341  042          .BYTE   341,042
3582  022036  300  042          .BYTE   300,042
3583  022040  342  042          .BYTE   342,042
3584  022042  320  042          .BYTE   320,042
3585  022044  344  042          .BYTE   344,042
3586  022046  310  042          .BYTE   310,042
3587  022050  350  042          .BYTE   350,042
3588  022052  304  042          .BYTE   304,042
3589  022054  360  042          .BYTE   360,042
3590  022056  302  042          .BYTE   302,042
3591  022060  350  042          .BYTE   350,042
3592  022062  304  042          .BYTE   304,042
3593  022064  344  042          .BYTE   344,042
3594  022066  310  042          .BYTE   310,042
3595  022070  342  042          .BYTE   342,042
3596  022072  320  042          .BYTE   320,042
3597  022074  341  042          .BYTE   341,042
3598  022076  300  042          .BYTE   300,042
3599
3600  022100  341  044          .BYTE   341,044        ;BIG TRIANGLE
3601  022102  300  044          .BYTE   300,044
3602  022104  341  044          .BYTE   341,044
```

```
3603  022106  300  044          .BYTE  300,044
3604  022110  343  044          .BYTE  343,044
3605  022112  320  044          .BYTE  320,044
3606  022114  343  044          .BYTE  343,044
3607  022116  320  044          .BYTE  320,044
3608  022120  347  044          .BYTE  347,044
3609  022122  336  044          .BYTE  336,044
3610  022124  347  044          .BYTE  347,044
3611  022126  336  044          .BYTE  336,044
3612  022130  357  044          .BYTE  357,044
3613  022132  337  044          .BYTE  337,044
3614  022134  357  044          .BYTE  357,044
3615  022136  334  044          .BYTE  334,044
3616  022140  377  044          .BYTE  377,044
3617  022142  336  044          .BYTE  336,044
3618  022144  340  044          .BYTE  340,044
3619  022146  300  044          .BYTE  300,044
3620
3621  022150  377  045          .BYTE  377,045          ;SQUARE
3622  022152  337  045          .BYTE  337,045
3623  022154  377  045          .BYTE  377,045
3624  022156  337  045          .BYTE  337,045
3625  022160  377  045          .BYTE  377,045
3626  022162  337  045          .BYTE  337,045
3627  022164  377  045          .BYTE  377,045
3628  022166  337  045          .BYTE  337,045
3629  022170  377  045          .BYTE  377,045
3630  022172  337  045          .BYTE  337,045
3631  022174  377  045          .BYTE  377,045
3632  022176  337  045          .BYTE  337,045
3633  022200  377  045          .BYTE  377,045
3634  022202  337  045          .BYTE  337,045
3635  022204  377  045          .BYTE  377,045
3636  022206  337  045          .BYTE  337,045
3637  022210  377  045          .BYTE  377,045
3638  022212  337  045          .BYTE  337,045
3639  022214  377  045          .BYTE  377,045
3640  022216  337  045          .BYTE  337,045
3641
3642  022220  340  063          .BYTE 340,063          ;NUMBER '3'
3643  022222  300  063          .BYTE 300,063
3644  022224  364  063          .BYTE 364,063
3645  022226  312  063          .BYTE 312,063
3646  022230  340  063          .BYTE 340,063
3647  022232  304  063          .BYTE 304,063
3648  022234  340  063          .BYTE 340,063
3649  022236  302  063          .BYTE 302,063
3650  022240  340  063          .BYTE 340,063
3651  022242  305  063          .BYTE 305,063
3652  022244  340  063          .BYTE 340,063
3653  022246  310  063          .BYTE 310,063
3654  022250  344  063          .BYTE 344,063
3655  022252  310  063          .BYTE 310,063
3656  022254  360  063          .BYTE 360,063
3657  022256  305  063          .BYTE 305,063
3658  022260  340  063          .BYTE 340,063
```

```
3659  022262  300  063                .BYTE 300,063
3660  022264  340  063                .BYTE 340,063
3661  022266  300  063                .BYTE 300,063
3662  022270  340  041  FAKE:         .BYTE   340,041          ;SMALL TRIANGLE
3663  022272  300  041                .BYTE   300,041
3664  022274  340  041                .BYTE   340,041
3665  022276  300  041                .BYTE   300,041
3666  022300  340  041                .BYTE   340,041
3667  022302  320  041                .BYTE   320,041
3668  022304  341  041                .BYTE   341,041
3669  022306  330  041                .BYTE   330,041
3670  022310  343  041                .BYTE   343,041
3671  022312  334  041                .BYTE   334,041
3672  022314  347  041                .BYTE   347,041
3673  022316  336  041                .BYTE   336,041
3674  022320  357  041                .BYTE   357,041
3675  022322  337  041                .BYTE   337,041
3676  022324  340  041                .BYTE   340,041
3677  022326  300  041                .BYTE   300,041
3678  022330  340  041                .BYTE   340,041
3679  022332  300  041                .BYTE   300,041
3680  022334  340  041                .BYTE   340,041
3681  022336  300  041                .BYTE   300,041
3682
3683  022340  000000           ENDCHR: .WORD 0
3684
3685  022342  000005           PLUSS:
3686  022342  340  041         .BYTE   340,041
3687  022344  301  041         .BYTE   301,041
3688  022346  340  041         .BYTE   340,041
3689  022350  301  041         .BYTE   301,041
3690  022352  340  041         .BYTE   340,041
3691  022354  301  041         .BYTE   301,041
3692  022356  340  041         .BYTE   340,041
3693  022360  301  041         .BYTE   301,041
3694  022362  340  041         .BYTE   340,041
3695  022364  301  041         .BYTE   301,041
3696  022366  377  041         .BYTE   377,041
3697  022370  337  041         .BYTE   337,041
3698  022372  340  041         .BYTE   340,041
3699  022374  301  041         .BYTE   301,041
3700  022376  340  041         .BYTE   340,041
3701  022400  301  041         .BYTE   301,041
3702  022402  340  041         .BYTE   340,041
3703  022404  301  041         .BYTE   301,041
3704  022406  340  041         .BYTE   340,041
3705  022410  301  041         .BYTE   301,041
3706  022412  000000           ENDPLS: 000000
3707
3708
3709  022414  377  001  EOLDIS: .BYTE   377,001          ;END OF  LINE CHAR, DISPLAYABLE AS A SQUARE
3710  022416  337  001         .BYTE   337,001
3711  022420  377  001         .BYTE   377,001
3712  022422  337  001         .BYTE   337,001
3713  022424  377  001         .BYTE   377,001
3714  022426  337  001         .BYTE   337,001
```

```
3715  022430  377  001          .BYTE  377,001
3716  022432  337  001          .BYTE  337,001
3717  022434  377  001          .BYTE  377,001
3718  022436  337  001          .BYTE  337,001
3719  022440  377  001          .BYTE  377,001
3720  022442  337  001          .BYTE  337,001
3721  022444  377  001          .BYTE  377,001
3722  022446  337  001          .BYTE  337,001
3723  022450  377  001          .BYTE  377,001
3724  022452  337  001          .BYTE  337,001
3725  022454  377  001          .BYTE  377,001
3726  022456  337  001          .BYTE  337,001
3727  022460  377  001          .BYTE  377,001
3728  022462  337  001          .BYTE  337,001
3729  022464  377  002          .BYTE  377,002      ;END OF  LINE CHAR, DISPLAYABLE AS A SQUARE
3730  022466  337  002          .BYTE  337,002
3731  022470  377  002          .BYTE  377,002
3732  022472  337  002          .BYTE  337,002
3733  022474  377  002          .BYTE  377,002
3734  022476  337  002          .BYTE  337,002
3735  022500  377  002          .BYTE  377,002
3736  022502  337  002          .BYTE  337,002
3737  022504  377  002          .BYTE  377,002
3738  022506  337  002          .BYTE  337,002
3739  022510  377  002          .BYTE  377,002
3740  022512  337  002          .BYTE  337,002
3741  022514  377  002          .BYTE  377,002
3742  022516  337  002          .BYTE  337,002
3743  022520  377  002          .BYTE  377,002
3744  022522  337  002          .BYTE  337,002
3745  022524  377  002          .BYTE  377,002
3746  022526  337  002          .BYTE  337,002
3747  022530  377  002          .BYTE  377,002
3748  022532  337  002          .BYTE  337,002
3749  022534  377  003          .BYTE  377,003      ;END OF  LINE CHAR, DISPLAYABLE AS A SQUARE
3750  022536  337  003          .BYTE  337,003
3751  022540  377  003          .BYTE  377,003
3752  022542  337  003          .BYTE  337,003
3753  022544  377  003          .BYTE  377,003
3754  022546  337  003          .BYTE  337,003
3755  022550  377  003          .BYTE  377,003
3756  022552  337  003          .BYTE  337,003
3757  022554  377  003          .BYTE  377,003
3758  022556  337  003          .BYTE  337,003
3759  022560  377  003          .BYTE  377,003
3760  022562  337  003          .BYTE  337,003
3761  022564  377  003          .BYTE  377,003
3762  022566  337  003          .BYTE  337,003
3763  022570  377  003          .BYTE  377,003
3764  022572  337  003          .BYTE  337,003
3765  022574  377  003          .BYTE  377,003
3766  022576  337  003          .BYTE  337,003
3767  022600  377  003          .BYTE  377,003
3768  022602  337  003          .BYTE  337,003
3769  022604  377  201          .BYTE  377,201      ;END OF  LINE CHAR, DISPLAYABLE AS A SQUARE
3770  022606  337  201          .BYTE  337,201
```

```
3771  022610  377  201          .BYTE  377,201
3772  022612  337  201          .BYTE  337,201
3773  022614  377  201          .BYTE  377,201
3774  022616  337  201          .BYTE  337,201
3775  022620  377  201          .BYTE  377,201
3776  022622  337  201          .BYTE  337,201
3777  022624  377  201          .BYTE  377,201
3778  022626  337  201          .BYTE  337,201
3779  022630  377  201          .BYTE  377,201
3780  022632  337  201          .BYTE  337,201
3781  022634  377  201          .BYTE  377,201
3782  022636  337  201          .BYTE  337,201
3783  022640  377  201          .BYTE  377,201
3784  022642  337  201          .BYTE  337,201
3785  022644  377  201          .BYTE  377,201
3786  022646  337  201          .BYTE  337,201
3787  022650  377  201          .BYTE  377,201
3788  022652  337  201          .BYTE  337,201
3789  022654  377  202          .BYTE  377,202      ;END OF  LINE CHAR, DISPLAYABLE AS A SQUARE
3790  022656  337  202          .BYTE  337,202
3791  022660  377  202          .BYTE  377,202
3792  022662  337  202          .BYTE  337,202
3793  022664  377  202          .BYTE  377,202
3794  022666  337  202          .BYTE  337,202
3795  022670  377  202          .BYTE  377,202
3796  022672  337  202          .BYTE  337,202
3797  022674  377  202          .BYTE  377,202
3798  022676  337  202          .BYTE  337,202
3799  022700  377  202          .BYTE  377,202
3800  022702  337  202          .BYTE  337,202
3801  022704  377  202          .BYTE  377,202
3802  022706  337  202          .BYTE  337,202
3803  022710  377  202          .BYTE  377,202
3804  022712  337  202          .BYTE  337,202
3805  022714  377  202          .BYTE  377,202
3806  022716  337  202          .BYTE  337,202
3807  022720  377  202          .BYTE  377,202
3808  022722  337  202          .BYTE  337,202
3809  022724  377  203          .BYTE  377,203      ;END OF  LINE CHAR, DISPLAYABLE AS A SQUARE
3810  022726  337  203          .BYTE  337,203
3811  022730  377  203          .BYTE  377,203
3812  022732  337  203          .BYTE  337,203
3813  022734  377  203          .BYTE  377,203
3814  022736  337  203          .BYTE  337,203
3815  022740  377  203          .BYTE  377,203
3816  022742  337  203          .BYTE  337,203
3817  022744  377  203          .BYTE  377,203
3818  022746  337  203          .BYTE  337,203
3819  022750  377  203          .BYTE  377,203
3820  022752  337  203          .BYTE  337,203
3821  022754  377  203          .BYTE  377,203
3822  022756  337  203          .BYTE  337,203
3823  022760  377  203          .BYTE  377,203
3824  022762  337  203          .BYTE  337,203
3825  022764  377  203          .BYTE  377,203
3826  022766  337  203          .BYTE  337,203
```

```
                                CHARACTER SET

3827  022770  377     203               .BYTE   377,203
3828  022772  337     203               .BYTE   337,203
3829  022774  000000              ENDLIN: 000000
3830  022776  000024              NUMBER:
3831  022776  000000                      .WORD   0
3832  023000  000000                      .WORD   0
3833  023002  000000                      .WORD   0
3834  023004  000000                      .WORD   0
3835  023006  000000                      .WORD   0
3836  023010  000000                      .WORD   0
3837  023012  000000                      .WORD   0
3838  023014  000000                      .WORD   0
3839  023016  000000                      .WORD   0
3840  023020  000000                      .WORD   0
3841  023022  000000                      .WORD   0
3842  023024  000000                      .WORD   0
3843  023026  000000                      .WORD   0
3844  023030  000000                      .WORD   0
3845  023032  000000                      .WORD   0
3846  023034  000000                      .WORD   0
3847  023036  000000                      .WORD   0
3848  023040  000000                      .WORD   0
3849  023042  000000                      .WORD   0
3850  023044  000000                      .WORD   0
3851  023046  000000              ENDNUM: 000000
3852
3853
3854
3855
3856
3857
3858
3859
3860  023050  001770              TSTLST: T0001
3861  023052  001770                      T0001
3862  023054  002072                      T0002
3863  023056  002152                      T0003
3864  023060  002226                      T0004
3865  023062  003064                      T0005
3866  023064  007352                      T0006
3867  023066  007464                      T0007
3868  023070  007566                      T0010
3869  023072  007654                      T0011
3870  023074  007746                      T0012
3871  023076  010120                      T0013
3872  023100  010226                      T0014
3873  023102  010312                      T0015
3874  023104  010424                      T0016
3875  023106  010514                      T0017
3876  023110  010772                      T0020
3877  023112  011160                      T0021
3878  023114  011266                      T0022
3879  023116  011362                      T0023
3880  023120  011456                      T0024
3881  023122  011552                      T0025
3882  023124  011646                      T0026
```

```
3883  023126  012102                          T0027
3884  023130  012350                          T0030
3885  023132  012442                          T0031
3886  023134  012554                          T0032
3887  023136  001770                          T0001
3888  023140  001770                          T0001
3889  023142  001770                          T0001
3890  023144  013016                          T0036
3891  023146  013122                          T0037
3892
3893
3894  023150  044124  051511  044440  MSG0:   .ASCIZ  /THIS IS THE VT71 CONTROL-VIDEO TEST/<1><15><12>
3895  023156  020123  044124  020105
3896  023164  052126  030467  041440
3897  023172  047117  051124  046117
3898  023200  053055  042111  047505
3899  023206  052040  051505  000524
3900  023214  005015     000
3901  023217     105  051122  051117  MSG1:   .ASCII  /ERROR/<1><15><12>
3902  023224  006401     012
3903  023227     120  047101  047440  MSG2:   .ASCII  /PAN OFFSET BITS DID NOT SET CORRECTLY/<1><15><12>
3904  023234  043106  042523  020124
3905  023242  044502  051524  042040
3906  023250  042111  047040  052117
3907  023256  051440  052105  041440
3908  023264  051117  042522  052103
3909  023272  054514  006401     012
3910  023277     127  051105  020105  MSG2A:  .ASCII  /WERE /
3911  023304  030060  030060  030060  MSG2B:  .ASCII  /000000 SHOULD HAVE BEEN /
3912  023312  051440  047510  046125
3913  023320  020104  040510  042526
3914  023326  041040  042505  020116
3915  023334  030060  030060  030060  MSG2C:  .ASCIZ  /000000/<1><15><12>
3916  023342  006401  000012
3917  023346  050123  041505  040511  MSG3:   .ASCIZ  /SPECIAL CHARACTER ENABLE BIT DID NOT CLEAR/<1><15><12>
3918  023354  020114  044103  051101
3919  023362  041501  042524  020122
3920  023370  047105  041101  042514
3921  023376  041040  052111  042040
3922  023404  042111  047040  052117
3923  023412  041440  042514  051101
3924  023420  006401  000012
3925  023424  050123  041505  040511  MSG4:   .ASCIZ  /SPECIAL CHARACTER ENABLE BIT DID NOT SET/<1><15><12>
3926  023432  020114  044103  051101
3927  023440  041501  042524  020122
3928  023446  047105  041101  042514
3929  023454  041040  052111  042040
3930  023462  042111  047040  052117
3931  023470  051440  052105  006401
3932  023476  000012
3933  023500  044504  050123  040514  MSG5:   .ASCIZ  /DISPLAY ENABLE BIT DID NOT SET/<1><15><12>
3934  023506  020131  047105  041101
3935  023514  042514  041040  052111
3936  023522  042040  042111  047040
3937  023530  052117  051440  052105
3938  023536  006401  000012
```

```
023542 044504 050123 040514 MSG6:   .ASCIZ  /DISPLAY ENABLE BIT DID NOT CLEAR/<1><15><12>
023550 020131 047105 041101
023556 042514 041040 052111
023564 042040 042111 047040
023572 052117 041440 042514
023600 051101 006401 000012
023606 047514 042101 041440 MSG7:   .ASCIZ  /LOAD CHARACTER GENERATOR BIT WILL NOT SET/<1><15><12>
023614 040510 040522 052103
023622 051105 043440 047105
023630 051105 052101 051117
023636 041040 052111 053440
023644 046111 020114 047516
023652 020124 042523 000524
023660 005015    000
023663    104 051503 020122 MSG10:  .ASCII  /DCSR BIT 15 DOES NOT CLEAR WHEN ROMS ARE FINISHED LOADING/<1><15><12>
023670 044502 020124 032461
023676 042040 042517 020123
023704 047516 020124 046103
023712 040505 020122 044127
023720 047105 051040 046517
023726 020123 051101 020105
023734 044506 044516 044123
023742 042105 046040 040517
023750 044504 043516 006401
023756    012
023757    105 042116 047440 MSG11:  .ASCII  /END OF TEXT CHAR NOT RECOGNIZED/<1><15><12>
023764 020106 042524 052130
023772 041440 040510 020122
024000 047516 020124 042522
024006 047503 047107 055111
024014 042105 006401    012
024021    042 021042 021042 MSG12:  .ASCIZ  /"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
024026 021042 021042 021042
024034 021042 021042 021042
024042 021042 021042 021042
024050 021042 021042 021042
024056 021042 021042 021042
024064 021042 021042 021042
024072 021042 021042 021042
024100 021042 021042 021042
024106 021042 021042 021042
024114 021042 021042 021042
024122 021042 021042 021042
024130 021042 021042 021042
024136 000442 005015 005012
024144 000012

024146 047514 040503 044524 MSG13:  .ASCIZ  /LOCATION/
024154 047117    000
024157    041 020441 020441 MSG14:  .ASCIZ  /!!!!!!!!!!!!!!!!!!!!!!!!!! ALTERNATE CROSSES AND TRIANGLES !!!!!!!!!!!!!!!
024164 020441 020441 020441
024172 020441 020441 020441
024200 020441 020441 020441
024206 020041 046101 042524
024214 047122 052101 020105
024222 051103 051517 042523
```

```
  3995  024230  020123  047101  020104
  3996  024236  051124  040511  043516
  3997  024244  042514  020123  020441
  3998  024252  020441  020441  020441
  3999  024260  020441  020441  020441
  4000  024266  020441  020441  020441
  4001  024274  020441  020441  020441
  4002  024302  020441  020441  000441
  4003  024310  045015  005015      000
  4004
  4005  024315      105  041501  020110  MSG15:  .ASCII  /EACH 1 /<1>
  4006  024322  020061      001
  4007  024325      127  051117  020104          .ASCII  /WORD 2 /<2>
  4008  024332  020062      002
  4009  024335      117  020106  020063          .ASCII  /OF 3 /<3>
  4010  024342      003
  4011  024343      124  044510  020123          .ASCII  /THIS 201 /<201>
  4012  024350  030062  020061      201
  4013  024355      115  051505  040523          .ASCII  /MESSAGE 202 /<202>
  4014  024362  042507  031040  031060
  4015  024370  101040
  4016  024372  044123  052517  042114          .ASCII  /SHOULD 203 /<203>
  4017  024400  031040  031460  101440
  4018  024406  042502  031040  031060          .ASCII  /BE 202 /<202>
  4019  024414  101040
  4020  024416  047117  031040  030460          .ASCII  /ON 201 /<201>
  4021  024424  100440
  4022  024426  020101  020063      003          .ASCII  /A 3 /<3>
  4023  024433      123  050105  051101          .ASCII  /SEPARATE 2 /<2>
  4024  024440  052101  020105  020062
  4025  024446      002
  4026  024447      114  047111  020105          .ASCII  /LINE 1 /<1>
  4027  024454  020061      001
  4028  024457      040  020040  000040  MSG16:  .ASCII  /    /<000>
  4029  024464  047105  020104  043117  MSG16A: .ASCIZ  /END OF TEXT CHAR(000) NOT RECOGNIZED/<1><15><12>
  4030  024472  052040  054105  020124
  4031  024500  044103  051101  030050
  4032  024506  030060  020051  047516
  4033  024514  020124  042522  047503
  4034  024522  047107  055111  042105
  4035  024530  006401  000012
  4036  024534  020040  020040      200  MSG16B: .ASCII  /    /<200>
  4037  024541      105  042116  047440  MSG16D: .ASCIZ  /END OF TEXT CHAR(200) NOT RECOGNIZED/<1><15><12>
  4038  024546  020106  042524  052130
  4039  024554  041440  040510  024122
  4040  024562  030062  024460  047040
  4041  024570  052117  051040  041505
  4042  024576  043517  044516  042532
  4043  024604  000504  005015      000
  4044  024611      124  051505  020124  MSG17:  .ASCII  /TEST 0 /
  4045  024616  020043
  4046  024620  030060  030060  030060  MSG17A: .ASCIZ  /000000/<1><15><12>
  4047  024626  006401  000012
  4048  024632  047125  042504  046122  MSG18:  .ASCIZ  /UNDERLINE MODE/<1>
  4049  024640  047111  020105  047515
  4050  024646  042504  000001
```

KL

```
 4051  024652  047502  042114  046440  MSG19:  .ASCIZ  /BOLD MODE/<1>
 4052  024660  042117  000505     000
 4053  024665     102  040514  045516  MSG20:  .ASCIZ  /BLANK MODE ERROR/<1>
 4054  024672  046440  042117  020105
 4055  024700  051105  047522  000522
 4056  024706     000
 4057  024707     122  053105  051105  MSG21:  .ASCIZ  /REVERSE VIDEO MODE/<1>
 4058  024714  042523  053040  042111
 4059  024722  047505  046440  042117
 4060  024730  000505     000
 4061  024733     124  040522  050120  MSG22:  .ASCIZ  /TRAPPED TO LOC 4 TRYING TO ACCESS DCSR/<1><15><12>
 4062  024740  042105  052040  020117
 4063  024746  047514  020103  020064
 4064  024754  051124  044531  043516
 4065  024762  052040  020117  041501
 4066  024770  042503  051523  042040
 4067  024776  051503  000522  005015
 4068  025004     000
 4069          025006                  .EVEN
 4070  025006  040501  040501  040501  MSG23A: .ASCII  /AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 4071  025014  040501  040501  040501
 4072  025022  040501  040501  040501
 4073  025030  040501  040501  040501
 4074  025036  040501  040501  040501
 4075  025044  040501  040501  040501
 4076  025052  040501  040501  040501
 4077  025060  040501  040501  040501
 4078  025066  040501  040501  040501
 4079  025074  040501  040501  040501
 4080  025102  040501  040501  040501
 4081  025110  040501  040501  040501
 4082  025116  040501  040501  040501
 4083  025124  040501     101
 4084          025130                  .EVEN
 4085  025130  041102  041102  041102  MSG23B: .ASCII  /BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
 4086  025136  041102  041102  041102
 4087  025144  041102  041102  041102
 4088  025152  041102  041102  041102
 4089  025160  041102  041102  041102
 4090  025166  041102  041102  041102
 4091  025174  041102  041102  041102
 4092  025202  041102  041102  041102
 4093  025210  041102  041102  041102
 4094  025216  041102  041102  041102
 4095  025224  041102  041102  041102
 4096  025232  041102  041102  041102
 4097  025240  041102  041102  041102
 4098  025246  041102     102
 4099          025252                  .EVEN
 4100  025252  041503  041503  041503  MSG23C: .ASCII  /CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
 4101  025260  041503  041503  041503
 4102  025266  041503  041503  041503
 4103  025274  041503  041503  041503
 4104  025302  041503  041503  041503
 4105  025310  041503  041503  041503
 4106  025316  041503  041503  041503
```

```
        025324  041503  041503  041503
        025332  041503  041503  041503
        025340  041503  041503  041503
        025346  041503  041503  041503
        025354  041503  041503  041503
        025362  041503  041503  041503
        025370  041503     103
                025374                  .EVEN
        025374  042104  042104  042104  MSG23D: .ASCII  /DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
        025402  042104  042104  042104
        025410  042104  042104  042104
        025416  042104  042104  042104
        025424  042104  042104  042104
        025432  042104  042104  042104
        025440  042104  042104  042104
        025446  042104  042104  042104
        025454  042104  042104  042104
        025462  042104  042104  042104
        025470  042104  042104  042104
        025476  042104  042104  042104
        025504  042104  042104  042104
        025512  042104
        025514  042524  052123  021440  MSG24:  .ASCII  /TEST 000011 ALL 1/<1>
        025522  030060  030060  030461
        025530  040440  046114  030440
        025536     001
        025537     127  051117  051504          .ASCII  /WORDS 2/<2>
        025544  031040     002
        025547     123  047510  046125          .ASCII  /SHOULD 3/<3>
        025554  020104  001463
        025560  042502  031040  030460          .ASCII  /BE 201/ <201>
        025566     201
        025567     117  020116  030062          .ASCII  /ON 202/<202>
        025574  101062
        025576  044124  020105  030062          .ASCII  /THE 203/<203>
        025604  101463
        025606  040523  042515  046040          .ASCII  /SAME LINE                                                /
        025614  047111  020105  020040
        025622  020040  020040  020040
        025630  020040  020040  020040
        025636  020040  020040  020040
        025644  020040  020040  020040
        025652  020040  020040  020040
        025660  020040  020040  020040
        025666  020040  020040  020040
        025674  020040  020040  020040
        025702  020040  020040
        025706  030060  030060  030060  MSG25:  .ASCII  /000000 ERRORS ON PASS /
        025714  042440  051122  051117
        025722  020123  047117  050040
        025730  051501  020123
        025734  030060  030060  030060  MSX25:  .ASCII  /000000/<1><15><12>
        025742  006401     012
        025745     060  030060  030060  MSG25Y: .ASCIZ  /000000 ERRORS ON ALL PASSES/<1><15><12>
        025752  020060  051105  047522
        025760  051522  047440  020116
```

5

M6

```
 4163  025766  046101  020114  040520
 4164  025774  051523  051505  006401
 4165  026002  000012
 4166  026004  051105  047522  020122  MSG26:  .ASCII  /ERROR - ZERO BLOCK COUNT NOT RECOGNIZED AS SUCH/<1><15><12><200>
 4167  026012  020055  042532  047522
 4168  026020  041040  047514  045503
 4169  026026  041440  052517  052116
 4170  026034  047040  052117  051040
 4171  026042  041505  043517  044516
 4172  026050  042532  020104  051501
 4173  026056  051440  041525  000510
 4174  026064  005015     200
 4175  026067     102  052111  030440  MSG27:  .ASCIZ  /BIT 15 OF A DISPLAY PARAMETER WORD DID NOT CAUSE A JUMP/<1><200>
 4176  026074  020065  043117  040440
 4177  026102  042040  051511  046120
 4178  026110  054501  050040  051101
 4179  026116  046501  052105  051105
 4180  026124  053440  051117  020104
 4181  026132  044504  020104  047516
 4182  026140  020124  040503  051525
 4183  026146  020105  020101  052512
 4184  026154  050115  100001     000
 4185  026161     104  051511  046120  MSG28:  .ASCIZ  /DISPLAY DID NOT INTERRUPT FOR A FULL SECOND/<1><15><12>
 4186  026166  054501  042040  042111
 4187  026174  047040  052117  044440
 4188  026202  052116  051105  052522
 4189  026210  052120  043040  051117
 4190  026216  040440  043040  046125
 4191  026224  020114  042523  047503
 4192  026232  042116  006401  000012
 4193  026240  042524  052123  021440  MSG29:  .ASCII  /TEST # /
 4194  026246     040
 4195  026247     060  030060  030060  MSG29A: .ASCIZ  /000000/<1><15><12>
 4196  026254  000460  005015     000
 4197  026261     044  022044  022044  MSG30:  .ASCIZ  /$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
 4198  026266  022044  022044  022044
 4199  026274  022044  022044  022044
 4200  026302  022044  022044  022044
 4201  026310  022044  022044  022044
 4202  026316  022044  022044  022044
 4203  026324  022044  022044  022044
 4204  026332  022044  022044  022044
 4205  026340  022044  022044  022044
 4206  026346  022044  022044  022044
 4207  026354  022044  022044  022044
 4208  026362  022044  022044  022044
 4209  026370  022044  022044  022044
 4210  026376  000444  006412     000
 4211  026403     122  043505  046125  MSG31:  .ASCIZ  /REGULAR MODE/<1>
 4212  026410  051101  046440  042117
 4213  026416  000505     000
 4214  026421     124  040522  050120  MSG32:  .ASCII  /TRAPPED TO LOC 4 FROM LOC /
 4215  026426  042105  052040  020117
 4216  026434  047514  020103  020064
 4217  026442  051106  046517  046040
 4218  026450  041517     040
```

6

A7

```
026453     060  030060  030060  MSX32:  .ASCIZ  /000000/<1><15><12>
026460  000460  005015     000
        026466                          .EVEN
026466  003703                          MSTB0:  3703
026470  026472                          MSTB1:  MSG33
026472  044504  050123  040514  MSG33:  .ASCII  /DISPLAY TABLE NOW AT LOC /
026500  020131  040524  046102
026506  020105  047516  020127
026514  052101  046040  041517
026522     040
026523     060  030060  030060  MSG33T: .ASCII  /000000/<1><15><12>
026530  000460  005015
026534  051501  044503  020111          .ASCII  /ASCIT NOW AT LOC /
026542  047516  020127  052101
026550  046040  041517     040
026555     060  030060  030060  MSG33A: .ASCIZ  /000000/<1><200><12><15>
026562  000460  005200  000015
026570  020441  020441  020441  MSG34:  .ASCIZ  /!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
026576  020441  020441  020441
026604  020441  020441  020441
026612  020441  020441  020441
026620  020441  020441  020441
026626  020441  020441  020441
026634  020441  020441  020441
026642  020441  020441  020441
026650  020441  020441  020441
026656  020441  020441  020441
026664  020441  020441  020441
026672  020441  020441  020441
026700  020441  020441  020441
026706  000441  006412  005012
026714     000
026715     000  001001  002003  MSG35A: .ASCII  <0><1><2><3><4><5><6><7><10><11><12>
026722  003005  004007  005011
026730  006013  007015  010017  MSG35B: .ASCII  <13><14><15><16><17><20><21><22><23><24><25>
026736  011021  012023     025
026743     026  014027  015031  MSG35C: .ASCII  <26><27><30><31><32><33><34><35><36><37><40>
026750  016033  017035  020037
026756  021041  022043  023045  MSG35D: .ASCII  <41><42><43><44><45><46><47><50><51><52><53>
026764  024047  025051     053
026771     054  027055  030057  MSG35E: .ASCII  <54><55><56><57><60><61><62><63><64><65><66>
026776  031061  032063  033065
027004  034067  035071  036073  MSG35F: .ASCII  <67><70><71><72><73><74><75><76><77><100><101>
027012  037075  040077     101
027017     102  042103  043105  MSG35G: .ASCII  <102><103><104><105><106><107><110><111><112><113><114>
027024  044107  045111  046113
027032  047115  050117  051121  MSG35H: .ASCII  <115><116><117><120><121><122><123><124><125><126><127>
027040  052123  053125     127
027045     130  055131  056133  MSG35I: .ASCII  <130><131><132><133><134><135><136><137><140><141><142>
027052  057135  060137  061141
027060  062143  063145  064147  MSG35J: .ASCII  <143><144><145><146><147><150><151><152><153><154><155>
027066  065151  066153     155
027073     156  070157  071161  MSG35K: .ASCII  <156><157><160><161><162><163><164><165><166><167><170>
027100  072163  073165  074167
027106  075171  076173  077175  MSG35L: .ASCII  <171><172><173><174><175><176><177><200><201><202><203>
027114  100177  101201     203
```

```
4275  027121    204  103205  104207  MSG35M: .ASCII  <204><205><206><207><210><211><212><213><214><215><216>
4276  027126 105211  106213  107215
4277  027134 110217  111221  112223  MSG35N: .ASCII  <217><220><221><222><223><224><225><226><227><230><231>
4278  027142 113225  114227    231
4279  027147    232  116233  117235  MSG350: .ASCII  <232><233><234><235><236><237><240><241><242><243><244>
4280  027154 120237  121241  122243
4281  027162 123245  124247  125251  MSG35P: .ASCII  <245><246><247><250><251><252><253><254><255><256><257>
4282  027170 126253  127255    257
4283  027175    260  131261  132263  MSG350: .ASCII  <260><261><262><263><264><265><266><267><270><271><272>
4284  027202 133265  134267  135271
4285  027210 136273  137275  140277  MSG35P: .ASCII  <273><274><275><276><277><300><301><302><303><304><305>
4286  027216 141301  142303    305
4287  027223    306  144307  145311  MSG358: .ASCII  <306><307><310><311><312><313><314><315><316><317><320>
4288  027230 146313  147315  150317
4289  027236 151321  152323  153325  MSG35T: .ASCII  <321><322><323><324><325><326><327><330><331><332><333>
4290  027244 154327  155331    333
4291  027251    334  157335  160337  MSG35U: .ASCII  <334><335><336><337><340><341><342><343><344><345><346>
4292  027256 161341  162343  163345
4293  027264 164347  165351  166353  MSG35V: .ASCII  <347><350><351><352><353><354><355><356><357><360><361>
4294  027272 167355  170357    361
4295  027277    362  172363  173365  MSG35W: .ASCII  <362><363><364><365><366><367><370><371><372><373><374>
4296  027304 174367  175371  176373
4297  027312 177375  172764  173766  MSG35X: .ASCII  <375><376><364><365><366><367><370><371><372><373><374>
4298  027320 174770  175772    374
4299
4300  027325    012  005012  005012  MSG35Y: .ASCII  <12><12><12><12><12><12><12><12><12><12><12>
4301  027332 005012  005012  005012
4302  027340 005012  005012  005012          .ASCII  <12><12><12><12><12><12><12><12><12><12><12><12><12>
4303  027346 005012  005012  005012
4304  027354    012
4305  027355    012  005012  005012          .ASCII  <12><12><12><12><12><12><12><12><12><12><12><12><12>
4306  027362 005012  005012  005012
4307  027370 005012
4308  027372 005012  005012  005012          .ASCII  <12><12><12><12><12><12><12><12><12><12><12><12><12><12>
4309  027400 005012  005012  005012
4310  027406 005012
4311  027410 005012  005012  005012          .ASCII  <12><12><12><12><12><12><12><12><12><12><12><12><12><12>
4312  027416 005012  005012  005012
4313  027424    012
4314  027425    012  005012  005012          .ASCIZ  <12><12><12><12><12>
4315  027432    000
4316  027433    132  051105  020117  MSG36:  .ASCII  /ZERO CHARACTER COUNT TEST/<1><15><12>
4317  027440 044103  051101  041501
4318  027446 042524  020122  047503
4319  027454 047125  020124  042524
4320  027462 052123  006401    012
4321  027467    123  051103  042505          .ASCIZ  /SCREEN SHOULD GO TOTALY BLANK FOR A FEW SECONDS/<1><15><12>
4322  027474 020116  044123  052517
4323  027502 042114  043440  020117
4324  027510 047524  040524  054514
4325  027516 041040  040514  045516
4326  027524 043040  051117  040440
4327  027532 043040  053505  051440
4328  027540 041505  047117  051504
4329  027546 006401  000012
4330  027552 020040  020040  020040  MSG37:  .ASCIZ  /                                                  /<1><15><12>
```

```
4331  027560  020040  020040  020040
4332  027566  020040  020040  020040
4333  027574  020040  020040  020040
4334  027602  020040  020040  020040
4335  027610  020040  020040  020040
4336  027616  020040  020040  020040
4337  027624  020040  020040  020040
4338  027632  006401  000012
4339  027636  044504  050123  040514  MSG38:  .ASCIZ  /DISPLAY INTERRUPT HAPPENED WITH PRIORITY SET TOO HIGH/<1><12><15>
4340  027644  020131  047111  042524
4341  027652  051122  050125  020124
4342  027660  040510  050120  047105
4343  027666  042105  053440  052111
4344  027674  020110  051120  047511
4345  027702  044522  054524  051440
4346  027710  052105  052040  047517
4347  027716  044040  043511  000510
4348  027724  006412  000
4349  027727  000123  051103  042505  MSG39:  .ASCIT  /SCREEN WILL GO BLANK FOR A FEW SECONDS/<1><15><12>
4350  027734  020116  044527  046114
4351  027742  043440  020117  046102
4352  027750  047101  020113  047506
4353  027756  020122  020101  042506
4354  027764  020127  042523  047503
4355  027772  042116  000523  005015
4356  030000  047504  052116  050040          .ASCIZ  /DONT PANIC/<1><15><12>
4357  030006  047101  041511  006401
4358  030014  000012
4359  030016  044124  051511  052040  MSG40:  .ASCII  /THIS TEXT SHOULD ALL BE/<1>
4360  030024  054105  020124  044123
4361  030032  052517  042114  040440
4362  030040  046114  041040  000505
4363  030046  047117  047440  042516          .ASCIZ  /ON ONE LINE/<1><200>
4364  030054  046040  047111  000505
4365  030062  000200
4366  030064  000200                  MSG41:  .ASCIZ  <200>
4367  030066  044124  051511  046440  MSG42:  .ASCIZ  /THIS MESSAGE SHOULD APPEAR IN 24 PLACES ON THE SCREEN/
4368  030074  051505  040523  042507
4369  030102  051440  047510  046125
4370  030110  020104  050101  042520
4371  030116  051101  044440  020116
4372  030124  032062  050040  040514
4373  030132  042503  020123  047117
4374  030140  052040  042510  051440
4375  030146  051103  042505  000116
4376  030154  020040  020040  020040          .ASCII  /                              /<200>
4377  030162  020040  020040  020040
4378  030170  020040  020040  020040
4379  030176  020040  020040  020040
4380  030204  100040
4381  030206  020041  020040  020040  MSG43:  .ASCIZ/!       !       !       !       !       !       !       !       !       !
4382  030214  020040  020041  020040
4383  030222  020040  020040  020041
4384  030230  020040  020040  020040
4385  030236  020041  020040  020040
4386  030244  020040  020041  020040
```

```
4387  030252  020040  020040  020041
4388  030260  020040  020040  020040
4389  030266  020041  020040  020040
4390  030274  020040  020041  020040
4391  030302  020040  020040  020041
4392  030310  020040  020040  020040
4393  030316  020041  020040  020040
4394  030324  020040  000441  005015
4395  030332     000
4396  030333     132  051105  020117  MSG44:  .ASCII  /ZERO BLOCK COUNT TEST    THE BAR TO THE RIGHT SHOULD BE STRAIGHT AND SO
4397  030340  046102  041517  020113
4398  030346  047503  047125  020124
4399  030354  042524  052123  020040
4400  030362  020040  044124  020105
4401  030370  040502  020122  047524
4402  030376  052040  042510  051040
4403  030404  043511  052110  051440
4404  030412  047510  046125  020104
4405  030420  042502  051440  051124
4406  030426  044501  044107  020124
4407  030434  047101  020104  047523
4408  030442  044514  020104  020040
4409  030450  020040     045
4410
4411  030453     103  040510  040522  MSG45:  .ASCII  /CHARACTER SET DID NOT LOAD PROPERLY/<1><15><12>
4412  030460  052103  051105  051440
4413  030466  052105  042040  042111
4414  030474  047040  052117  046040
4415  030502  040517  020104  051120
4416  030510  050117  051105  054514
4417  030516  006401     012
4418  030521     104  050103  053440          .ASCII  /DCP WAS /
4419  030526  051501     040
4420  030531     060  030060  030060  MSG45A: .ASCII  /000000/<1><15><12>
4421  030536  000460  005015
4422  030542  041504  020120  044123          .ASCII  /DCP SHOULD HAVE BEEN /
4423  030550  052517  042114  044040
4424  030556  053101  020105  042502
4425  030564  047105     040
4426  030567     060  030060  030060  MSG45B: .ASCIZ  /000000 /<1><15><12>
4427  030574  020060  006401  000012
4428  030602  050040  020103  052101  MSG54:  .ASCII  / PC AT /
4429  030610     040
4430  030611     060  030060  030060  MSG54A: .ASCIZ  /000000 /<1><15><12>
4431  030616  020060  006401  000012
4432  030624  020040  020040  020040  MSG57:  .ASCIZ  /                                                    /<1><15><12>
4433  030632  020040  020040  020040
4434  030640  020040  020040  020040
4435  030646  020040  020040  020040
4436  030654  020040  020040  020040
4437  030662  020040  020040  020040
4438  030670  020040  020040  006401
4439  030676  000012
4440  030700  044502  020124  032461  MSG70:  .ASCIZ  /BIT 15 IS SET = HALT ON ERROR/<1><15><12>
4441  030706  044440  020123  042523
4442  030714  020124  020055  040510
```

```
4443  030722  052114  047440  020116
4444  030730  051105  047522  000522
4445  030736  005015      000
4446
4447  030741      102  052111  030440  MSG71:  .ASCIZ  /BIT 14 IS SET - LOOP ON CURRENT TEST/<1><15><12>
4448  030746  020064  051511  051440
4449  030754  052105  026440  046040
4450  030762  047517  020120  047117
4451  030770  041440  051125  042522
4452  030776  052116  052040  051505
4453  031004  000524  005015      000
4454  031011      102  052111  030440  MSG72:  .ASCIZ  /BIT 13 IS SET - INHIBIT ERROR MESSAGES/<1><15><12>
4455  031016  020063  051511  051440
4456  031024  052105  026440  044440
4457  031032  044116  041111  052111
4458  031040  042440  051122  051117
4459  031046  046440  051505  040523
4460  031054  042507  000523  005015
4461  031062      000
4462
4463  031063      102  052111  030440  MSG73:  .ASCIZ  /BIT 12 IS SET -/<1><15><12>
4464  031070  020062  051511  051440
4465  031076  052105  026440  006401
4466  031104  000012
4467  031106  044502  020124  030461  MSG74:  .ASCIZ  /BIT 11 IS SET -/<1><15><12>
4468  031114  044440  020123  042523
4469  031122  020124  000455  005015
4470  031130      000
4471  031131      102  052111  034440  MSG76:  .ASCIZ  /BIT 9 IS SET - LOOP ON ERROR TEST/<1><15><12>
4472  031136  044440  020123  042523
4473  031144  020124  020055  047514
4474  031152  050117  047440  020116
4475  031160  051105  047522  020122
4476  031166  042524  052123  006401
4477  031174  000012
4478  031176  044502  051524  033440  MSG77:  .ASCII  /BITS 7-0  -  GO DIRECTLY TO TEST /
4479  031204  030055  020040  020055
4480  031212  043440  020117  044504
4481  031220  042522  052103  054514
4482  031226  052040  020117  042524
4483  031234  052123      040
4484  031237      060  030060  030060  MSG77A: .ASCIZ  /000000/<1><15><12>
4485  031244  000460  005015      000
4486  031251      102  052111  034040  MSG78:  .ASCII  /BIT 8 IS SET - LOOP ON TEST /
4487  031256  044440  020123  042523
4488  031264  020124  020055  047514
4489  031272  050117  047440  020116
4490  031300  042524  052123      040
4491  031305      060  030060  030060  MSG78A: .ASCIZ  /000000/<1><15><12>
4492  031312  000460  005015      000
4493  031317      101  046114  041040  MSG79:  .ASCIZ  /ALL BITS IN THE SOFTWARE SWITCH REGISTER ARE CLEAR/<1><15><12>
4494  031324  052111  020123  047111
4495  031332  052040  042510  051440
4496  031340  043117  053524  051101
4497  031346  020105  053523  052111
4498  031354  044103  051040  043505
```

```
 4499  031362  051511  042524  020122
 4500  031370  051101  020105  046103
 4501  031376  040505  000522  005015
 4502  031404     000
 4503  031405     124  042510  043040  MSG80:  .ASCIZ  /THE FOLLOWING SOFTWARE SWITCH REGISTER BITS ARE SET.../<1><15><12>
 4504  031412  046117  047514  044527
 4505  031420  043516  051440  043117
 4506  031426  053524  051101  020105
 4507  031434  053523  052111  044103
 4508  031442  051040  043505  051511
 4509  031450  042524  020122  044502
 4510  031456  051524  040440  042522
 4511  031464  051440  052105  027056
 4512  031472  000456  005015     000
 4513
 4514  031477     130  054130  054130  BIOTMP: .ASCIZ  /XXXXXXXX/
 4515  031504  054130  000130
 4516          000200                  .END 200
```

```
BIOCT   015504
BIOTMP  031477
BIT0  = 000001
BIT00 = 000001
BIT01 = 000002
BIT02 = 000004
BIT03 = 000010
BIT04 = 000020
BIT05 = 000040
BIT06 = 000100
BIT07 = 000200
BIT08 = 000400
BIT09 = 001000
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BM0     000746
BM1     000750
BM2     000752
BPTVEC= 000014
BUZZ    014662
CDTP    001032
CHARS   016110
CHPCNT  001036
CLRTUB  014136
CNCHAR  000740
CNERRO  000744
CNRECV  000742
CNXFER  000736
CONER   014660
CR    = 000015
CRLF  = 000200
DCP     001034
DCSR    001026
DDISP = 177570
DIHAN   015246
DISPLA  000642
DISPRE  000174
DISTBL  015760
DOWNFA  000764
DSVAD1  001020
DSVAD2  001022
DSWR  = 177570
EMTVEC= 000030
ENDCHR  022340
ENDLIN  022774
ENDNUM  023046
ENDPLS  022412
ENDSIN  013652
EOLDIS  022414
EOPTBL  013012
ERMES   014532
ERRPAS  001056
ERRVEC= 000004
FAKE    022270
FAKEY   001052
FINDTT  001306
FXTST   015102
GITCOD  013654
HALTER  014656
HT    = 000011
IDTP    001030
INSERT  014056
INTCNT  001050
IOTVEC= 000020
KBID0   001054
KBSR    001000
KBSRV   015244
KBUF    001002
KBVAD1  001010
KBVAD2  001012
LBUF    001006
LCSR    001004
LDSRV   015244
LDVAD1  001014
LDVAD2  001016
LEAVEC  000760
LF    = 000012
LINCNT  000770
LSTALL  015002
MAXBLK  001060
MSGOUT  014752
MSGY    013312
MSGYA   013434
MSG0    023150
MSG1    023217
MSG10   023663
MSG11   023757
MSG12   024021
MSG13   024146
MSG14   024157
MSG15   024315
MSG16   024457
MSG16A  024464
MSG16B  024534
MSG16D  024541
MSG17   024611
MSG17A  024620
MSG18   024632
MSG19   024652
MSG2    023227
MSG2A   023277
MSG2B   023304
MSG2C   023334
MSG2P   024665
MSG21   024707
MSG22   024733
MSG23A  025006
MSG23B  025130
MSG23C  025252
MSG23D  025374
MSG24   025514
MSG25   025706
MSG25Y  025745
MSG26   026004
MSG27   026067
MSG28   026161
MSG29   026240
MSG29A  026247
MSG3    023346
MSG30   026261
MSG31   026403
MSG32   026421
MSG33   026472
MSG33A  026555
MSG33T  026523
MSG34   026570
MSG35A  026715
MSG35B  026730
MSG35C  026743
MSG35D  026756
MSG35E  026771
MSG35F  027004
MSG35G  027017
MSG35H  027032
MSG35I  027045
MSG35J  027060
MSG35K  027073
MSG35L  027106
MSG35M  027121
MSG35N  027134
MSG35O  027147
MSG35P  027162
MSG35Q  027175
MSG35R  027210
MSG35S  027223
MSG35T  027236
MSG35U  027251
MSG35V  027264
MSG35W  027277
MSG35X  027312
MSG35Y  027325
MSG36   027433
MSG37   027552
MSG38   027636
MSG39   027727
MSG4    023424
MSG40   030016
MSG41   030064
MSG42   030066
MSG43   030206
MSG44   030333
MSG45   030453
MSG45A  030531
MSG45B  030567
MSG5    023500
MSG54   030602
MSG54A  030611
MSG57   030624
MSG6    023542
MSG7    023606
MSG70   030700
MSG71   030741
MSG72   031011
MSG73   031063
MSG74   031106
MSG76   031131
MSG77   031176
MSG77A  031237
MSG78   031251
MSG78A  031305
MSG79   031317
MSG80   031405
MSTALL  015172
MSTB0   026466
MSTB1   026470
MSX25   025734
MSX32   026453
NUMBER  022776
PASCNT  001040
PIRQ  = 177772
PIRQVE= 000240
PLUSS   022342
PR0   = 000000
PR1   = 000040
PR2   = 000100
PR3   = 000140
PR4   = 000200
PR5   = 000240
PR6   = 000300
PR7   = 000340
PS    = 177776
PSW   = 177776
PWRVEC= 000024
RESVEC= 000010
RTNTT   014750
R6    =%000006
R7    =%000007
SINCHR  013602
SPCHAR  016730
SPCHR   015022
SPMODE  000754
SSTALL  015226
STACK = 001100
START   001062
STKLMT= 177774
STLCNT  000772
SWR     000640
SWREG   000176
SW0   = 000001
SW00  = 000001
SW01  = 000002
SW02  = 000004
SW03  = 000010
SW04  = 000020
SW05  = 000040
SW06  = 000100
SW07  = 000200
SW08  = 000400
SW09  = 001000
SW1   = 000002
SW10  = 002000
SW11  = 004000
SW12  = 010000
SW13  = 020000
SW14  = 040000
SW15  = 100000
SW2   = 000004
SW3   = 000010
SW4   = 000020
SW5   = 000040
SW6   = 000100
SW7   = 000200
SW8   = 000400
SW9   = 001000
TBITVE= 000014
TBL22   016074
TBL23   016100
TEMP    000766
TKVEC = 000060
TPB     000776
TPS     000774
TPVEC = 000064
```

DZKVBB.P11 04-NOV-76 13:51 SYMBOL TABLE

```
TRAPER  015420          T0016   010424          YDSPTR  013436          $GDADR  000620          $SETUP= 000021
TRAPVE= 000034          T0017   010514          YLINES  013120          $GDDAT  000624          $STUP = 177777
TRTVEC= 000014          T0017A  010630          YSTART  013122          $GET42  012766          $SVLAD  014466
TSTLOD  015254          T0017C  010762          $AUTOB  000634          $HD   = 000001          $SWR  = 147400
TSTLST  023050          T0020   010772          $BDADR  000622          $ICNT   000604          $SWRMK= 000000
TTOUT   014732          T0021   011160          $BDDAT  000626          $INTAG  000635          $TIMES  000722
TTYAVA  001042          T0022   011266          $BELL   000726          $ITEMB  000614          $TKB    000646
TTYOUT  014722          T0023   011362          $CMTAG  000600          $LF     000734          $TKS    000644
TUBOUT  013750          T0024   011456          $CM1  = 000010          $LPADR  000606          $TMP0   000702
TUBSWT  000756          T0025   011552          $CM2  = 000020          $LPERR  000610          $TMP1   000704
TUBTM0  001044          T0026   011646          $CM3  = 000010          $MXCNT  014530          $TMP2   000706
TUBTM1  001046          T0027   012102          $CM4  = 000010          $NULL   000654          $TMP3   000710
T0000   001514          T0030   012350          $CRLF   000733          $OVER   014514          $TMP4   000712
T0001   001770          T0031   012442          $DOAGN  013006          $PASS   000600          $TMP5   000714
T0002   002072          T0032   012554          $ENDAD  012776          $QUES   000732          $TMP6   000716
T0003   002152          T0036   013016          $ENDCT  012762          $REGAD  000660          $TMP7   000720
T0004   002226          T0037   013122          $EOP    012726          $REG0   000662          $TN   = 000001
T0005   003064          UPFAST  000762          $EOPCT  012754          $REG1   000664          $TPB    000652
T0006   007352          WCHAR   001024          $ERFLG  000603          $REG2   000666          $TPFLG  000657
T0007   007464          XCODE   013110          $ERMAX  000615          $REG3   000670          $TPS    000650
T0010   007566          XDISP   013112          $ERRPC  000616          $REG4   000672          $TSTNM  000602
T0011   007654          XDSPTB  013104          $ERRTB  001062          $REG5   000674          $XTSTR  014272
T0012   007746          XSTALL  015210          $ERTTL  000612          $REG6   000676          $$GET4= 000000
T0013   010120          XSTART  013016          $ESCAP  000724          $REG7   000700          .     = 031510
T0014   010226          YBLOCK  013116          $FILLC  000656          $RTNAD  013010
T0015   010312          YCODE   013114          $FILLS  000655          $SCOPE  014210

. ABS.  031510      000
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

DZKVBB,DZKVBB/SOL/NL:TOC_DZKVBB.P11
RUN-TIME:  44 37 1 SECONDS
RUN-TIME RATIO:  454/83=5.4
CORE USED:  18K  (36 PAGES)
```