LV11

PRINTER/PLOTTER TEST

MD-11-DZLVA-B

EP-DZLVA-B-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A.

IDENTIFICATION

*** SEQ 0001

PROGRAM-ID:

PROGRAMMER:

DATE:

MAINTAINER:

ABSTRACT

THIS IS A DYNAMIC TEST OF THE LV-11 PRINTER PLOTTER
HARDWARE AND INTERFACE. THIS TEST IS USED TO DETERMINE
THE ABILITY OF THE HARDWARE TO EXECUTE BOTH ALPHA PRINTED AND
GRAPHIC PLOTTED DATA. THE TEST IS DIVIDED INTO TWO PARTS
THE FIRST IS A MANUAL INTERVERTION TEST WHERE THE OPERATOR MUST
EXECUTE THE OPERATION TYPED ON THE CONSOLE TELETYPE.
THE SECOND TEST THE OPERATOR MUST VISUALLY INSPECT THE
PRINTED PATTERN ON THE ELECTROSTATIC PAPER FROM THE PRINTER.

THIS PROGRAM WILL RUN ON NON-SWITCH REGISTER CPU TYPES.

IF RUNNING ON A PDP15 WITH NON-STANDARD ADDRESSES:

HALT PDP15 AND PDP11, ENABLE PDP11
'LOAD' ASB11 LOADER (START ADDR. 17700) - RESET,
READIN

LOAD PDP11 ADDRESS 160000 FOR 4K, 100000 FOR 8K,
120000 FOR 12K, 140000 FOR 16K, START
DEPRESS CONTINUE ON PDP15 TO READ IN
MAINDEC
LOAD PDP11 ADDRESS 1246 DEPOSIT 164000
LOAD PDP11 ADDRESS 1242 DEPOSIT 100170
LOAD PDP11 ADDRESS 200. ENABLE AND START.
PROGRAM TEST SHOULD RUN
IF ADDRESS LOCATIONS ARE WRONG. PROGRAM
WILL HALT WITH DISPLAY 10.

2. REQUIREMENTS

2.1 EQUIPTMENT

PDP-11 COMPUTER WITH CONSOLE TERMINAL
LV-11 INTERFACE MODULE (M7258)
LV01-AX 8 INCH PRINTER/PLOTTER
OR
LV01-BX 11 INCH PRINTER/PLOTTER

2.2 STORAGE

THIS PROGRAM USES 4K OF MEMORY.

3. LOADING PROCEDURE

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

STANDARD PDP-11 FORMAT

SW15=1 HALT ON ERROR
SW14=1 LOOP ON TEST
SW13=1 INHIBIT ERROR TYPEOUTS
SW12=1 LOOP ON A LINE
SW11=1 BELL ON ERROR
SW9 =1 LOOP ON ERROR
SW8 =1 LOOP ON TEST IN SWR<7:0>

4.2 STARTING ADDRESSES

200 600 IS THE STARTING ADDRESS FOR THE LOGIC SUBTESTS.
204 604 IS THE RESTART ADDRESS FOR THE LOGIC SUBTEST.
210 610 IS THE STARTING ADDRESSFOR THE MANUAL INTERVENTION TEST.

5. OPERATING PROCEDURE

5.1 MANUAL INTERVENTION TEST

THE OPERATOR MUST EXECUTE THE INSTRUCTIONS TYPED ON THE CONSOLE TELETYPE AND DEPRESS CONT. WHEN OPERATION COMPLETED.

5.2 LOGIC AND DYNAMIC TEST

THE OPERATOR MUST VERIFY VISUALLY THE PATTERN PRINTED OR PLOTTED.

6. ERRORS

THIS PROGRAM USES THE DIAGNOSTIC "SYSMAC" PACKAGE FOR ERROR REPORTING AND TYPEOUT. THE PROGRAM WILL TYPE THE ERRORING ADDRESS, THE BUS ADDRESS, AND THE VECTOR ADDRESS.

ERRPC:	LOCATION AT WHICH AN ERROR WAS DETECTED.
BUSADR:	BUS ADDRESS
BUSVECT:	BUS VECTOR ADDRESS

7. RESTRICTIONS

ONCE THE PROGRAM HAS BEEN STARTED AT LOCATION 200, IT MUST BE RESTARTED ONLY AT 204/504.

9. MISCELANEOUS

9.1 EXECUTION TIME

MANUAL INTERVENTION TEST - N/A
DYNAMIC LOGIC AND VISUAL TEST - 5 MIN.
UPON COMPLETION OF ONE PASS, AN END-OF-PASS MESSAGE WILL BE TYPED OUT.

9.2 DEVICE ADDRESS PROGRAM LOCATIONS

LOCATION 1250 CONTAINS THE LV-11 DEVICE ADDRESS. ($BASE)
LOCATION 1244 CONTAINS THE LV-11 INTERRUPT VECTOR. ($VECT1)
LOCATION 1244 CONTAINS THE LV-11 INTERRUPT LEVEL. ($VECT1)
LOCATION 1626 CONTAINS THE CPU DELAY FACTOR <SLEW TEST ONLY>
<PRESET TO 1 FOR 11/05>

9. PROGRAM DESCRIPTION

9.1 MANUAL INTERVENTION TEST

THIS TEST THE OPERATOR MUST FOLLOW THE MESSAGE PRINTED ON
THE CONSOLE TELETYPE. THIS TEST IS USED TO VERIFY THE PROPER
OPERATION OF THE STATUS/CONTROL AND DATA REGISTER'S.

9.2 BASIC LOGIC TEST

THE PURPOSE OF THIS IS TO CHECK THE BASIC CONTROL/STATUS
AND DATA REGISTERS. ALSO TESTED ARE THE VECTOR ADDRESS
AND BR LEVEL.

9.3 DATA PATH TEST (PRINT MODE)

THE PURPOSE OF THIS TEST IS TO PREFORM BASIC CHECK OF THE
DATA PATH'S BETWEEN THE COMPUTER AND THE PRINTER.
ALTERNATING LINES OF "@?@?@?" AND "*U*U*U" ARE PRINTED ACROSS ALL 132 COLUMNS
TO TEST THE DATA LINES. THIS PATTERN IS REPEATED FOR 128 LINES.

9.4 PRINTABLE/NON-PRINTABLE TEST (PRINT MODE)

THE PURPOSE OF THIS TEST IS TO VERIFY THAT ALL LEGAL
PRINTABLE CHARACTERS MAY BE PRINTED. THIS ALSO
TESTS THAT THE NON-PRINTING CHARACTERS ARE NOT PRINTED.
A LINE OF LEGAL CHARACTERS IS PRINTED (CODES 240 THRU 377)
FOLLOWED BY A BLANK LINE. THIS BLANK LINE IS THE RESULT OF PRINTING
THE CODES 200-237 (EXECPT CODES 204(EOT),212(LF),214(FF),215 CR).
THIS PATTERN IS REPEATED 40 TIMES.

9.5 SINGLE CHARACTER PER LINE (PRINT MODE)

THIS PATTERN IS USED TO VERIFY THAT ALL CHARACTERS CAN
BE PRINTED IN ALL CHARACTER POSITIONS AND ALL LINES.
THE FULL WIDTH OF THE PAPER SHOULD BE FILLED WITH ONLY
THAT CHARACTER. BOTH UPPER AND LOWER CASE CHARACTERS AND
ALL NUMBERS AND SYMBOLS SHOULD BE PRINTED (CODES 240-377).
THIS PATTERN IS PRINTED ONLY ONCE.

9.6 ROTATING PATTERN (PRINT MODE)

THIS PATTERN IS USED TO VERIFY DIFFERENT CHARACTERS MAY BE
PRINTED ON THE SAME LINE. THE FIRST CHARACTER OF THE
FIRST LINE SHOULD BE A "SPACE" CHARACTER FOLLOWED BY
!"#$%&'() ETC. CHARACTERS ACROSS THE FULL WIDTH OF THE PAPER.
(IE. !"#$%&'()*+,-./0123456789:)
EACH LINE WILL BE ROTATED ONE CHARACTER TO THE LEFT AND THE
ROTATING PATTERN IS 96 LINES LONG.
THIS IS A BINARY COUNT PATTERN AND SHOULD INCLUDE BOTH
UPPER AND LOWER CASE CHARACTERS (CODES 240-377).
THIS PATTERN IS PRINTED ONLY ONCE.

9.7 CHARACTER WEDGE <PRINT MODE>

THIS TEST IS DESIGNED TO TEST THAT EACH CHARACTER POSITION
MAY BE SELECTED INDIVIDUALLY. THE FIRST LINE OF THE WEDGE
OF "RUBOUT'S" ARE PRINTED ACROSS THE FULL WIDTH OF THE PAPER.
THE SECOND LINE OF THE WEDGE THE "RUBOUT" CHARACTER
IS PRINTED ACROSS THE WIDTH-1 ETC.
THIS IS EXECUTED IN A DECENDING AND ASSENDING WEDGE PATTERN.
THIS PATTERN IS PRINTED ONLY ONCE.

9.8 ROTATING ONE BIT DATA PATTERN <PLOT MODE>

EACH CONFIGURATION OF THIS PATTERN IS PLOTED ACROSS
THE ENTIRE LINE LENGTH AND EACH LINE IS REPEATED 127 TIMES.
TO GIVE A VERTICAL LINE PLOT APPROXIMATELY 1 1/4 INCH LONG.
FOR EACH BIT POSITION IN THE ROTATING ONE BIT PATTERN (8 BITS)
(IE. 200-100-40-20-10-4-2-1)
THIS PATTERN IS PLOTTED ONLY ONCE.

9.9 ACCUMULATING BIT DATA PATTERN <PLOT MODE>

EACH CONFIGURATION OF THE PATTERN IS PLOTED ACROSS THE ENTIRE
LINE LENGTH AND EACH LINE IS REPEATED 127 TIMES TO GIVE A
VERTICAL LINE PLOT APPROXIMATELY 1 1/4 INCH LONG FOR EACH
ACCUMULATIVE BIT POSITION. IN THIS TEST THERE WILL BE EIGHT
VERTICAL LINE PLOTS INCREASING TO THE RIGHT ONE POSITION.
EACH ONE CORSPONDING TO THE ACCUMULATIVE BIT POSITION IN
THE CHARACTER. THIS PATTERN IS PLOTED ONLY ONCE.

9.10 ALTERNATING ONE AND ZERO DATA PATTERN <PLOT MODE>

EACH CONFIGURATION OF THE PATTERN IS PLOTTED ACROSS THE
ENTIRE LINE LENGTH AND EACH LINE IS REPEATED 127 TIMES T
GIVE A VERTICAL LINE PLOT APPROXIMATELY 1 1/4 INCH LONG.
THE RESULTING PATTERN APPEARS TO BE ATERNATING SOLID AND
BLANK BANDS. THIS PATTERN IS PLOTTED ONLY ONCE.

9.11 DIMINISHING BIT DATA PATTERN <PLOT MODE>

EACH CONFIGURATION OF THE PATTERN IS PLOTTED ACROSS THE ENTIRE
LINE LENGTH AND EACH LINE IS REPEATED 127 TIMES TO GIVE
A VERTICAL LINE PLOT APPROXIMATELY 1 1/4 INCH LONG FOR EACH
DIMINISHING BIT POSITION. IN THIS TEST THERE WILL BE EIGHT
VERTICAL LINE PLOTS DIMINISHING TO THE LEFT ONE POSITION.
EACH ONE CORSPONDING TO THE DIMINISHING BIT POSITION IN
THE CHARACTER. THIS PATTERN IS PLOTTED ONLY ONCE.

9.12 GRAPHIC WEDGE (PLOT MODE)

THIS PATTERN IS DESIGNED TO TEST THAT EACH PLOT POSITION
MAYBE SELECTED INDIVIDUALLY. THE FIRST LINE OF THE WEDGE
IS PLOTTED ACROSS THE FULL WIDTH OF THE PAPER.
THE SECOND LINE OF THE WEDGE IS PRINTED ACROSS THE WIDTH
-1 ETC. THIS PATTERN IS EXECUTED IN A DECENDING AND ASSENDING
WEDGE PATTERN. THIS PATTERN IS PLOTTED ONLY ONCE.

9.13 PAPER SLEW (PLOT MODE)

THIS TEST IS DESIGNED TO CHECK FOR PAPER MOTION WHEN NOT
PRINTING. A WIDE BAND OF PLOTTED DATA IS PRINTED AND A DELAY (LESS THAN 1 SEC.)
IS INITIATED. AFTER THE DELAY THE DATA IS PLOTTED AGAIN.
THERE SHOULD BE NO VISIABLE GAP BETWEEN THE PLOTTED DATA
(THERE MAYBE A SHADE CHANGE).
THIS PATTERN IS REPEATED ONLY ONCE.

10. SOFTWARE SWITCH REGISTER

IF YOU ARE RUNNING ON A NON-SWITCH REGISTER CPU OR SWITCH
REGISTER IS SET TO ALL ONES, THE SWITCH SETTING CAN BE
CHANGED BY TYPING CONTROL G. THE COMPUTER WILL TYPE
OUT THE VALUE OF THE SWITCH REGISTER AND WAIT FOR INPUT.
TO KEEP THE SAME SWITCH VALUE, TYPE A CARRIAGE RETURN;
OTHERWISE, TYPE NEW SWITCH SETTINGS FOLLOWED BY A CARRIAGE
RETURN.

11 BASIC DEFINITIONS
19 OPERATIONAL SWITCH SETTINGS
20 TRAP CATCHER
(1) STARTING ADDRESS(ES)
30 ACT11 HOOKS
34 APT PARAMETER BLOCK
35 COMMON TAGS
(2) APT MAILBOX-ETABLE
(3) ERROR POINTER TABLE
INITIALIZE THE COMMON TAGS
T1 TEST FOR TIME OUT ERROR
T2 TEST FOR NO ERROR FLAG, TEST BIT:5 IS RESET
T3 TEST THAT READY FLAG TEST BIT7 IS SET
T4 TEST THAT INTERRUPT ENABLE MAY BE SET AND CLEARED (BIT 6)
T5 TEST THAT MODE MAY BE SET
T6 TEST THAT MODE MAY BE CLEARED
T7 TEST THAT LOADING THE BUFFER RESETS READY
T10 TEST THAT READY WILL SET AFTER A DELAY
T11 TEST THAT THE PRINTER WILL INTERRUPT
T12 TEST THAT THE PRINTER CAN INTERRUPT ON LEVEL INDICATED -1
T13 TEST THAT THE PRINTER CAN NOT INTERRUPT ON LEVEL INDICATED
T14 TEST THAT THE PRINTER CAN NOT INTERRUPT ON LEVEL INDICATED +1
T15 CLEAN UP
T16 DATA TRANSFER PATH TEST
T17 PRINTABLE AND NON-PRINTABLE CHARACTERS
T20 SINGLE CHARACTER ACROSS ALL COLS. (96 CHARACTERS)
T21 ROTATING CHARACTERS ACROSS ALL COLS. (96 CHARACTERS)
T22 DOUBLE WEDGE PATTERN (CHARACTER)
T23 DOUBLE WEDGE PATTERN
T24 BASIC GRAPH MODE TEST
T25 GRAPH MODE TEST
T26 ALTERNATING ONE AND ZERO TEST
637 T27 DIMINISHING BIT TEST
553 T30 GRAPH-MODE DATA WEDGE (LEFT WEDGE)
T31 PLOT A LENGTH OF FULL GRAPH DATA
END OF PASS ROUTINE
MANUAL INTERVENTION TEST
PRINT/PLOT SUBROUTINE
ASCII MESSAGES
TTY INPUT ROUTINE
TYPE ROUTINE
BINARY TO OCTAL (ASCII) AND TYPE
CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
APT COMMUNICATIONS ROUTINE
ERROR HANDLER ROUTINE
ERROR MESSAGE TYPEOUT ROUTINE
SCOPE HANDLER ROUTINE
POWER DOWN AND UP ROUTINES
TRAP DECODER
TRAP TABLE

```
                                .TITLE  MAINDEC-11-DZLVA-B
                                ;*COPYRIGHT (C) 1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY RAYMOND SHOOP
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C0),MAR 21, 1976.
                                ;*
000001                          $TN=1
160000                          $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT

                                .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
                                .EQUIV  R6,SP           ;;STACK POINTER
                                .EQUIV  R7,PC           ;;PROGRAM COUNTER

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

```
(1)                             ;*"SWITCH REGISTER" SWITCH DEFINITIONS
(1)             100000          SW15=   100000
(1)             040000          SW14=   40000
(1)             020000          SW13=   20000
(1)             010000          SW12=   10000
(1)             004000          SW11=   4000
(1)             002000          SW10=   2000
(1)             001000          SW09=   1000
(1)             000400          SW08=   400
(1)             000200          SW07=   200
(1)             000100          SW06=   100
(1)             000040          SW05=   40
(1)             000020          SW04=   20
(1)             000010          SW03=   10
(1)             000004          SW02=   4
(1)             000002          SW01=   2
(1)             000001          SW00=   1
(1)                             .EQUIV  SW09,SW9
(1)                             .EQUIV  SW08,SW8
(1)                             .EQUIV  SW07,SW7
(1)                             .EQUIV  SW06,SW6
(1)                             .EQUIV  SW05,SW5
(1)                             .EQUIV  SW04,SW4
(1)                             .EQUIV  SW03,SW3
(1)                             .EQUIV  SW02,SW2
(1)                             .EQUIV  SW01,SW1
(1)                             .EQUIV  SW00,SW0
(1)
(1)                             ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)             100000          BIT15=  100000
(1)             040000          BIT14=  40000
(1)             020000          BIT13=  20000
(1)             010000          BIT12=  10000
(1)             004000          BIT11=  4000
(1)             002000          BIT10=  2000
(1)             001000          BIT09=  1000
(1)             000400          BIT08=  400
(1)             000200          BIT07=  200
(1)             000100          BIT06=  100
(1)             000040          BIT05=  40
(1)             000020          BIT04=  20
(1)             000010          BIT03=  10
(1)             000004          BIT02=  4
(1)             000002          BIT01=  2
(1)             000001          BIT00=  1
(1)                             .EQUIV  BIT09,BIT9
(1)                             .EQUIV  BIT08,BIT8
(1)                             .EQUIV  BIT07,BIT7
(1)                             .EQUIV  BIT06,BIT6
(1)                             .EQUIV  BIT05,BIT5
(1)                             .EQUIV  BIT04,BIT4
(1)                             .EQUIV  BIT03,BIT3
(1)                             .EQUIV  BIT02,BIT2
(1)                             .EQUIV  BIT01,BIT1
```

```
(1)                             .EQUIV  BIT00,BIT0
(1)
(1)
(1)                             ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1)          000004             ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
(1)          000010             RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
(1)          000014             TBITVEC=14              ;;"T" BIT
(1)          000014             TRTVEC= 14              ;;TRACE TRAP
(1)          000014             BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
(1)          000020             IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1)          000024             PWRVEC= 24              ;;POWER FAIL
(1)          000030             EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
(1)          000034             TRAPVEC=34              ;;"TRAP" TRAP
(1)          000060             TKVEC=  60              ;;TTY KEYBOARD VECTOR
(1)          000064             TPVEC=  64              ;;TTY PRINTER VECTOR
(1)          000240             PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
 12
 13          177514                     ABASE=177514
 14          100200                     AVECT1=100200
 15          000200                     APRIOR=200
 16          163400                     $SWR=163400
 17          000001                     $TN=1
 18
 19                             .SBTTL  OPERATIONAL SWITCH SETTINGS
(1)                             ;*
(1)                             ;*      SWITCH                  USE
(1)                             ;*      ------          --------------------
(1)                             ;*        15            HALT ON ERROR
(1)                             ;*        14            LOOP ON TEST
(1)                             ;*        13            INHIBIT ERROR TYPEOUTS
(1)                             ;*        12            LOOP ON A LINE
(1)                             ;*        10            BELL ON ERROR
(1)                             ;*         9            LOOP ON ERROR
(1)                             ;*         8            LOOP ON TEST IN SWR<7:0>
 20                             .SBTTL  TRAP CATCHER
(1)
(1)          000000                     .=0
(1)                             ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
(1)                             ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
(1)                             ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
(1)          000174                     .=174
(1)  000174  000000             DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
(1)  000176  000000             SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
(1)                             .SBTTL  STARTING ADDRESS(ES)
(1)  000200  000137  001662             JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
 21  000204  000137  001666             JMP     RSTRT           ;JUMP TO RESTART ADDRESS OF PROGRAM
 22  000210  000137  005262             JMP     MANTST          ;JUMP TO START OF MANUAL INTERVENTION TEST
 23
 24          000600                     .=600
 25
 26  000600  000137  001662             JMP     START           ;JUMP TO STARTING ADDRESS OF PROGRAM
 27  000604  000137  001666             JMP     RSTRT           ;JUMP TO RESTART ADDRESS OF PROGRAM
 28  000610  000137  005262             JMP     MANTST          ;JUMP TO START OF MANUAL INTERVENTION TEST
```

```
                                .SBTTL  ACT11 HOOKS

                                ;;******************************************************
                                ;HOOKS REQUIRED BY ACT11
                                        $SVPC=.                 ;SAVE PC
000614                                  .=46
000046                                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
000052                                  .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
000614                                  .=$SVPC                 ;; RESTORE PC

001000                                  .=1000

                                .SBTTL  APT PARAMETER BLOCK

                                ;;******************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;******************************************************
001000                                  .$X=.   ;;SAVE CURRENT LOCATION
000024                                  .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                          200     ;;FOR APT START UP
000044                                  .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001000                          $APTHDR ;;POINT TO APT HEADER BLOCK
001000                                  .=.$X   ;;RESET LOCATION COUNTER
                                ;;******************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

001000                          $APTHD:
001000  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001002  001172                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000310                  $TSTM:  .WORD   200.    ;;RUN TIM OF LONGEST TEST
001006  000310                  $PASTM: .WORD   200.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  000310                  $UNITM: .WORD   200.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000031                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
 35                                      .SBTTL  COMMON TAGS
(1)
(2)                                     ;;**********************************************************************
(1)                                     ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
(1)                                     ;*USED IN THE PROGRAM.
(1)
(1)             001100                          .=1100
(1)  001100                             $CMTAG:                         ;;START OF COMMON TAGS
(1)  001100  000000                             .WORD   0
(1)  001102     000             $TSTNM: .BYTE   0                       ;;CONTAINS THE TEST NUMBER
(1)  001103     000             $ERFLG: .BYTE   0                       ;;CONTAINS ERROR FLAG
(1)  001104  000000             $ICNT:  .WORD   0                       ;;CONTAINS SUBTEST ITERATION COUNT
(1)  001106  000000             $LPADR: .WORD   0                       ;;CONTAINS SCOPE LOOP ADDRESS
(1)  001110  000000             $LPERR: .WORD   0                       ;;CONTAINS SCOPE RETURN FOR ERRORS
(1)  001112  000000             $ERTTL: .WORD   0                       ;;CONTAINS TOTAL ERRORS DETECTED
(1)  001114     000             $ITEMB: .BYTE   0                       ;;CONTAINS ITEM CONTROL BYTE
(1)  001115     001             $ERMAX: .BYTE   1                       ;;CONTAINS MAX. ERRORS PER TEST
(1)  001116  000000             $ERRPC: .WORD   0                       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
(1)  001120  000000             $GDADR: .WORD   0                       ;;CONTAINS ADDRESS OF 'GOOD' DATA
(1)  001122  000000             $BDADR: .WORD   0                       ;;CONTAINS ADDRESS OF 'BAD' DATA
(1)  001124  000000             $GDDAT: .WORD   0                       ;;CONTAINS 'GOOD' DATA
(1)  001126  000000             $BDDAT: .WORD   0                       ;;CONTAINS 'BAD' DATA
(1)  001130  000000                     .WORD   0                       ;;RESERVED--NOT TO BE USED
(1)  001132  000000                     .WORD   0
(1)  001134     000             $AUTOB: .BYTE   0                       ;;AUTOMATIC MODE INDICATOR
(1)  001135     000             $INTAG: .BYTE   0                       ;;INTERRUPT MODE INDICATOR
(1)  001136  000000                     .WORD   0
(1)  001140  177570             SWR:    .WORD   DSWR                    ;;ADDRESS OF SWITCH REGISTER
(1)  001142  177570             DISPLAY: .WORD  DDISP                   ;;ADDRESS OF DISPLAY REGISTER
(1)  001144  177560             $TKS:   177560                          ;;TTY KBD STATUS
(1)  001146  177562             $TKB:   177562                          ;;TTY KBD BUFFER
(1)  001150  177564             $TPS:   177564                          ;;TTY PRINTER STATUS REG. ADDRESS
(1)  001152  177566             $TPB:   177566                          ;;TTY PRINTER BUFFER REG. ADDRESS
(1)  001154     000             $NULL:  .BYTE   0                       ;;CONTAINS NULL CHARACTER FOR FILLS
(1)  001155     002             $FILLS: .BYTE   2                       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
(1)  001156     012             $FILLC: .BYTE   12                      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
(1)  001157     000             $TPFLG: .BYTE   0                       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
(1)  001160  000000             $ESCAPE:0                               ;;ESCAPE ON ERROR ADDRESS
(1)  001162  177607  000377        $BELL:  .ASCIZ  <207><377><377>      ;;CODE FOR BELL
(1)  001166     077             $QUES:  .ASCII  /?/                     ;;QUESTION MARK
(1)  001167     015             $CRLF:  .ASCII  <15>                    ;;CARRIAGE RETURN
(1)  001170  000012             $LF:    .ASCIZ  <12>                    ;;LINE FEED
(2)                                     ;;**********************************************************************
(2)                                     .SBTTL  APT MAILBOX-ETABLE
(2)
(3)                                     ;;**********************************************************************
(2)                                     .EVEN
(2)  001172                             $MAIL:                          ;;APT MAILBOX
(2)  001172  000000             $MSGTY: .WORD   AMSGTY                  ;;MESSAGE TYPE CODE
(2)  001174  000000             $FATAL: .WORD   AFATAL                  ;;FATAL ERROR NUMBER
(2)  001176  000000             $TESTN: .WORD   ATESTN                  ;;TEST NUMBER
(2)  001200  000000             $PASS:  .WORD   APASS                   ;;PASS COUNT
(2)  001202  000000             $DEVCT: .WORD   ADEVCT                  ;;DEVICE COUNT
(2)  001204  000000             $UNIT:  .WORD   AUNIT                   ;;I/O UNIT NUMBER
```

```
                        $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
                        $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
                        $ETABLE:                ;;APT ENVIRONMENT TABLE
                        $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
                        $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
                        $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
                        $USWR:  .WORD   AUSWR   ;;USER SWITCHES
                        $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
                        ;*                      BITS 15-11=CPU TYPE
                        ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                        ;*                              11/70=06,PDQ=07,Q=10
                        ;*                      BIT 10=REAL TIME CLOCK
                        ;*                      BIT  9=FLOATING POINT PROCESSOR
                        ;*                      BIT  8=MEMORY MANAGEMENT
                        $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
                        $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                                                ;;MEM. TYPE BYTE -- (HIGH BYTE)
                        ;*                              900 NSEC CORE=001
                        ;*                              300 NSEC BIPOLAR=002
                        ;*                              500 NSEC MOS=003
                        $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                        ;*                       MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
                        $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
                        $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
                        $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
                        $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
                        $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
                        $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
                        $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
                        $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
                        $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
                        $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#
                        $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2,BUS PRIORITY#2
                        $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT TO BE TESTED
                        $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
                        $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#
                        $ETEND:
                        .MEXIT
```

```
                                .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 (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

001254                          $ERRTB:

                                ;ITEM   1
001254  010201                          EM1                     ;PRINTER ERROR BIT SET
001256  012121                          DH1
001260  012156                          DT1
001262  012166                          DF1

                                ;ITEM   2
001264  010233                          EM2                     ;PRINTER READY BIT FAILED TO SET
001266  012121                          DH1
001270  012156                          DT1
001272  012166                          DF1

                                ;ITEM   3
001274  010277                          EM3                     ;PRINTER ERROR BIT FAILED TO SET ON POWER OFF
001276  012121                          DH1
001300  012156                          DT1
001302  012166                          DF1

                                ;ITEM   4
001304  010360                          EM4                     ;PRINTER ERROR BIT FAILED TO SET ON POWER ON
001306  012121                          DH1
001310  012156                          DT1
001312  012166                          DF1
```

```
                                    ;ITEM   5
001314  010440                              EM5             ;ERROR BIT FAILED TO SET ON PRINTER DISCONNECTED
001316  012121                              DH1
001320  012156                              DT1
001322  012166                              DF1

                                    ;ITEM   6
001324  010534                              EM6             ;ERROR BIT FAILED TO RESET UPON PRINTER CONNECTED
001326  012121                              DH1
001330  012156                              DT1
001332  012166                              DF1

                                    ;ITEM   7
001334  010631                              EM7             ;ERROR BIT FAILED TO SET ON NO PAPER
001336  012121                              DH1
001340  012156                              DT1
001342  012166                              DF1

                                    ;ITEM   10
001344  010711                              EM10            ;ERROR BIT FAILED TO RESET ON PAPER AVAILABLE
001346  012121                              DH1
001350  012156                              DT1
001352  012166                              DF1

                                    ;ITEM   11
001354  011002                              EM11            ;RESET FAILED TO CLEAR LP ERROR BIT
001356  012121                              DH1
001360  012156                              DT1
001362  012156                              DF1

                                    ;ITEM   12
001364  011051                              EM12            ;RESET FAILED TO SET LP READY BIT
001366  012121                              DH1
001370  012156                              DT1
001372  012166                              DF1

                                    ;ITEM   13
001374  011116                              EM13            ;INTRPT. EN. BIT FAILED TO SET CLEAR LP STATUS
001376  012121                              DH1
001400  012156                              DT1
001402  012166                              DF1

                                    ;ITEM   14
001404  011200                              EM14            ;INTRPT. EN. FAILED TO RESET
001406  012121                              DH1
001410  012156                              DT1
001412  012166                              DF1

                                    ;ITEM   15
001414  011240                              EM15            ;MODE FAILED TO SET
001416  012121                              DH1
001420  012156                              DT1
001422  012166                              DF1
```

```
                                    ;ITEM   16
001424  011267                              EM16            ;MODE FAILED TO CLEAR
001426  012121                              DH1
001430  012156                              DT1
001432  012166                              DF1

                                    ;ITEM   17
001434  011320                              EM17            ;READY FLAG CLEARED IN ERROR
001436  012121                              DH1
001440  012156                              DT1
001442  012166                              DF1

                                    ;ITEM   20
001444  011360                              EM20            ;READY FAILED TO RESET UPON LOADING PRINTER BUFFER
001446  012121                              DH1
001450  012156                              DT1
001452  012166                              DF1

                                    ;ITEM   21
001454  011446                              EM21            ;ERROR-PRINTER ERROR BIT UNEXPECTEDLY SET
001456  012121                              DH1
001460  012156                              DT1
001462  012166                              DF1

                                    ;ITEM   22
001464  011460                              EM22            ;READY FAILED TO SET AFTER A DELAY
001466  012121                              DH1
001470  012156                              DT1
001472  012166                              DF1

                                    ;ITEM   23
001474  011526                              EM23            ;ERROR BIT SET IN PRINTER
001476  012121                              DH1
001500  012156                              DT1
001502  012166                              DF1

                                    ;ITEM   24
001504  011563                              EM24            ;READY BIT RESET IN PRINTER
001506  012121                              DH1
001510  012156                              DT1
001512  012166                              DF1

                                    ;ITEM   25
001514  011622                              EM25            ;PRINTER FAILED TO INTERRUPT
001516  012121                              DH1
001520  012156                              DT1
001522  012166                              DF1

                                    ;ITEM   26
001524  011662                              EM26            ;PRINTER FAILED TO INTRPT. ON LEVEL INDICATED
001526  012121                              DH1
001530  012156                              DT1
001532  012166                              DF1
```

F02


```
171                                     ;ITEM   27
172  001534  011745                              EM27            ;PRINTER INTERRUPTED ON LEVEL INDICATED
173  001536  012121                              DH1
174  001540  012156                              DT1
175  001542  012166                              DF1
176
177                                     ;ITEM   30
178  001544  012020                              EM30            ;ERROR FLAG SET
179  001546  012121                              DH1
180  001550  012156                              DT1
181  001552  012166                              DF1
182
183                                     ;ITEM   31
184  001554  012057                              EM31            ;TRAP TO LOC 4 WHEN ADRESSING LV11
185  001556  012121                              DH1
186  001560  012156                              DT1
187  001562  012166                              DF1
```

```
   183  001564  177514                  LVAD:   ABASE                   ;LV-11 DEVICE ADDRESS
   190  001566  100200                  LVIV:   AVEC*1                  ;LV-11 INTERRUPT VECTOR
   191  001570  000200                  LVBRL:  APRIOR                  ;LV-11 INTERRUPT LEVEL
   192  001572  000001                  ADELAY: 1                       ;CPU DELAY CONSTANT
   193  001574  004000                  DELNUM: 4000
   194  001576  000140                  K0001:  140
   195  001600  000036                  K0002:  36
   196
   197  001602  177514                  LVS:    177514
   198  001604  177516                  LVB:    177516
   199  001606  000200                  LVVCT:  200
   200  001610  000202                  LVVCT1: 202
   201  001612  000000                  DSAVE:  0
   202  001614  000000                  PSAVE:  0
   203  001616  000000                  STCHAR: 0
   204  001620  000200                  LASTCH: 200
   205  001622  000000                  FIRST:  0
   206  001624  000000                  BRLEV1: 0
   207  001626  000000                  BRLEV2: 0
   208  001630  000000                  BRLEV3: 0
   209  001632  000000                  TEMP:   0
   210  001634  000204                  WIDTH:  132.
   211  001636  000102                  WIDTHB: 66.
   212  001640  000200                  WIDTHG: 128.
   213  001642  000000                  GMDW1:  0
   214  001644  000000                  GMDW2:  0
   215  001646  000000                  GMDW3:  0
   216  001650  000000                  GMDW4:  0
   217  001652  000000                  GMDW5:  0
   218  001654  000000                  GMDW6:  0
   219  001656  000000                  SAVE1:  0
   220  001660  000000                  SAVE2:  0
```

```
                                    START:  CLR     R0
                                            BR      INIT0
                                    RSTRT:  INC     R0
                                    INIT0:
                                    .SBTTL  INITIALIZE THE COMMON TAGS
                                    ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001670  012706  001100                      MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001674  005026                      CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001676  022706  001140              CMP     #SWR,R6 ;;DONE?
001702  001374                      BNE     .-6             ;;LOOP BACK IF NO
001704  012706  001100              MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                    ;;INITIALIZE A FEW VECTORS
001710  012737  014472  000020      MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001716  012737  000340  000022      MOV     #340,@#IOTVEC+2 ;;LEVEL 7
001724  012737  014146  000030      MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
001732  012737  000340  000032      MOV     #340,@#EMTVEC+2 ;;LEVEL 7
001740  012737  015042  000034      MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001746  012737  000340  000036      MOV     #340,@#TRAPVEC+2;LEVEL 7
001754  012737  014664  000024      MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
001762  012737  000340  000026      MOV     #340,@#PWRVEC+2 ;;LEVEL 7
001770  005037  001160              CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
001774  112737  000001  001115      MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
002002  012737  002002  001106      MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002010  012737  002010  001110      MOV     #.,$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.
002016  013746  000004              MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002022  012737  002056  000004      MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002030  012737  177570  001140      MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002036  012737  177570  001142      MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002044  022777  177777  177066      CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002052  001012                      BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                            ;;AND  THE HARDWARE SWR IS NOT = -1
002054  000403                      BR      65$             ;;BRANCH IF NO TIMEOUT
002056  012716  002064      64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
002062  000002                      RTI
002064  012737  000176  001140  65$:  MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
002072  012737  000174  001142      MOV     #DISPREG,DISPLAY
002100  012637  000004          66$:  MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

002104  005037  001200              CLR     $PASS           ;;CLEAR PASS COUNT
002110  132737  000200  001213      BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
002116  001403                      BEQ     67$             ;;YES,USE NON-APT SWITCH
002120  012737  001214  001140      MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
002126                          67$:
002126  005700                      TST     R0              ;TYPE HEADER?
002130  001002                      BNE     INIT            ;NO
002132  104400  006420              TYPE    ,MSG0           ;TYPE HEADER MESSAGE
```

```
230  002136  013737  001246  001564  INIT.   MOV     $BASE,LVAD      ;LOAD DEVICE ADDRESS
231  002144  013737  001246  001602          MOV     $BASE,LVS       ;LOAD DEVICE ADDRESS
232  002152  013737  001246  001604          MOV     $BASE,LVB       ;LOAD DEVICE ADDRESS
233  002160  062737  000002  001604          ADD     #2,LVB          ;UPDATE BUFFER ADDRESS
234  002166  013737  001242  001566          MOV     $VECT1,LVIV     ;LOAD VECTOR ADDRESS
235  002174  042737  170000  001566          BIC     #170000,LVIV
236  002202  013737  001566  001606          MOV     LVIV,LVVCT      ;LOAD VECTOR ADDRESS
237  002210  013737  001566  001610          MOV     LVIV,LVVCT1     ;LOAD VECTOR ADDRESS
238  002216  062737  000002  001610          ADD     #2,LVVCT1       ;UPDATE VECTOR ADDRESS
239  002224  113737  001243  001570          MOVB    $VECT1+1,LVBRL  ;LOAD PRIORITY
240  002232  105037  001571                  CLRB    LVBRL+1
241  002236  013737  001570  001624          MOV     LVBRL,BRLEV1    ;SET UP BR LEVEL
242  002244  162737  000040  001624          SUB     #40,BRLEV1      ;BR LEVEL=-1 INDICATED
243  002252  013737  001570  001526          MOV     LVBRL,BRLEV2    ;BR LEVEL=INDICATED
244  002260  013737  001570  001630          MOV     LVBRL,BRLEV3    ;SET UP BR LEVEL
245  002266  062737  000040  001630          ADD     #40,BRLEV3      ;BR LEVEL = +1 INDICATED
```

```
                        ;;*****************************************************************
                        ;*TEST 1         TEST FOR TIME OUT ERROR
                        ;;*****************************************************************
002274  000004          TST1:   SCOPE
002276  013746  000004          MOV     @#ERRVEC,-(SP)  ;SAVE ERRVEC
002302  012737  002322  000004  MOV     #1$,@#ERRVEC    ;SET UP FOR TIME OUT ERROR
002310  005077  177266          CLR     @LVS            ;ADDRESS LV11
002314  005077  177264          CLR     @LVB            ;
002320  000402                  BR      2$              ;;SKIP NEXT INSTRUCTION
002322  022626          1$:     CMP     (SP)+,(SP)+     ;POP 2 WORDS OFF THE STACK
002324  104031                  ERROR   31              ;CANNOT ADDRESS BOARD
002326  012637  000004  2$:     MOV     (SP)+,@#ERRVEC  ;RESTORE ERRVEC

                        ;;*****************************************************************
                        ;*TEST 2         TEST FOR NO ERROR FLAG, TEST BIT15 IS RESET
                        ;;*****************************************************************
002332  000004          TST2:   SCOPE
002334  000005                  RESET                   ;CLEAR THE WORLD
002336  005777  177240          TST     @LVS            ;TEST LP STATUS ERROR
002342  100001                  BPL     TST3            ;;GO TO NEXT TEST
002344  104011                  ERROR   11              ;ERROR, RESET FAILED TO CLEAR
                                                        ; LP ERROR BIT

                        ;;*****************************************************************
                        ;*TEST 3         TEST THAT READY FLAG TEST BIT7 IS SET
                        ;;*****************************************************************
002346  000004          TST3:   SCOPE
002350  000005                  RESET                   ;CLEAR THE WORLD
002352  105777  177224          TSTB    @LVS            ;TEST BIT 7
002356  100401                  BMI     TST4            ;;GO TO NEXT TEST
002360  104012                  ERROR   12              ;ERROR, RESET FAILED TO SET
                                                        ; LP READY BIT


                        ;;*****************************************************************
                        ;*TEST 4         TEST THAT INTERRUPT ENABLE MAY BE SET AND CLEARED (BIT 6)
                        ;;*****************************************************************
002362  000004          TST4:   SCOPE
002364  052737  000340  177776  BIS     #340,PSW        ;RAISE PSW
002372  012777  000100  177202  MOV     #BIT6,@LVS      ;LOAD BIT 6 INTO LP STATUS
002400  000240                  NOP
002402  032777  000100  177172  BIT     #BIT6,@LVS      ;TEST BIT 6
002410  001001                  BNE     1$              ;;BRANCH OVER
002412  104013                  ERROR   13              ;ERROR, INTERRUPT ENABLE BIT
                                                        ; FAILED TO SET
002414  005077  177162  1$:     CLR     @LVS            ;CLEAR LP STATUS
002420  000240                  NOP
002422  032777  000100  177152  BIT     #BIT6,@LVS      ;TEST BIT 6
002430  001401                  BEQ     TST5            ;;GO TO NEXT TEST
002432  104014                  ERROR   14              ;ERROR, INTERRUPT ENABLE FAILED
                                                        ; TO RESET
```

```
                                ;;*****************************************************************
                                ;*TEST 5         TEST THAT MODE MAY BE SET
                                ;;*****************************************************************
(2) 002434  000004              TST5:   SCOPE
296 002436  012777  000001  177136          MOV     #BIT0,@LVS      ;LOAD 'MODE
297 002444  000240                          NOP
298 002446  032777  000001  177126          BIT     #BIT0,@LVS      ;TEST BIT 0
299 002454  001001                          BNE     TST6            ;;GO TO NEXT TEST
300 002456  104015                          ERROR   15              ;ERROR, MODE FAILED TO SET
301
302
                                ;;*****************************************************************
                                ;*TEST 6         TEST THAT MODE MAY BE CLEARED
                                ;;*****************************************************************
(2) 002460  000004              TST6:   SCOPE
304 002462  012777  000001  177112          MOV     #BIT0,@LVS      ;LOAD MODE
305 002470  012777  000000  177104          MOV     #0,@LVS         ;LOAD  NOT 'MODE'
306 002476  000240                          NOP
307 002500  032777  000001  177074          BIT     #BIT0,@LVS      ;TEST BIT 0
308 002506  001401                          BEQ     TST7            ;;GO TO NEXT TEST
309 002510  104016                          ERROR   16              ;ERROR, MODE FAILED TO CLEAR
310
311
                                ;;*****************************************************************
                                ;*TEST 7         TEST THAT LOADING THE BUFFER RESETS READY
                                ;;*****************************************************************
(2) 002512  000004              TST7:   SCOPE
312 002514  105777  177062                  TSTB    @LVS            ;TEST FOR READY
313 002520  100401                          BMI     1$              ;;BRANCH OVER
314 002522  104017                          ERROR   17              ;ERROR, READY FLAG CLEARED IN ERROR
315 002524  012777  000101  177052  1$:     MOV     #101,@LVB       ;ENSURE NON-ZERO BUFFER
316 002532  012777  000015  177044          MOV     #15,@LVB        ;MOVE A 'CR' TO THE PRINTER BUFFER
317 002540  105777  177036                  TSTB    @LVS            ;TEST BIT 7, IS IT SET
320 002544  100001                          BPL     2$              ;;NO, BRANCH OVER
321 002546  104020                          ERROR   20              ;ERROR, READY FAILED TO RESET
322                                                                 ; UPON LOADING PRINTER BUFFER
323 002550  005777  177026          2$:     TST     @LVS            ;TEST BIT 15 (ERROR)
324 002554  100001                          BPL     TST10           ;;IT'S SET, GO TO NEXT TEST
325 002556  104021                          ERROR   21              ;ERROR, PRINTER ERROR BIT UNEXPECTEDLY SET
326
327
                                ;;*****************************************************************
                                ;*TEST 10        TEST THAT READY WILL SET AFTER A DELAY
                                ;;*****************************************************************
(2) 002560  000004              TST10:  SCOPE
330 002562  012737  000040  001612          MOV     #40,DSAVE
331 002570  012777  000015  177006          MOV     #15,@LVB        ;PRINT A CHARACTER
332 002576  004737  006322          1$:     JSR     PC,DELAY1
333 002602  105777  176774                  TSTB    @LVS            ;TEST READY
334 002606  100404                          BMI     TST11           ;;BRANCH IF SET
335 002610  005337  001612                  DEC     DSAVE           ;NOT SET, TEST COUNTER
336 002614  001370                          BNE     1$              ;BRANCH IF NON-ZERO
337 002616  104022                          ERROR   22              ;ERROR, READY FAILED TO SET
340                                                                 ; AFTER A DELAY
```

```
                                ;;*****************************************************************
                                ;*TEST 11       TEST THAT THE PRINTER WILL INTERRUPT
                                ;;*****************************************************************
002620  000004                  TST11:  SCOPE

002622  012737  000340  177776          MOV     #340,PSW        ;LOCK OUT INTERRUPTS
002630  012777  002720  176750          MOV     #4$,@LVVCT      ;LOAD VECTOR RETURN
002636  012777  000340  176744          MOV     #340,@LVVCT1    ; ADDRESS
002644  005777  176732                  TST     @LVS            ;TEST FOR ERRORS
002650  100001                          BPL     1$              ;;BRANCH IF NO ERROR
002652  104023                          ERROR   23              ;ERROR, ERROR BIT SET IN PRINTER
002654  105777  176722          1$:     TSTB    @LVS            ;TEST FOR READY
002660  100401                          BMI     2$              ;;BRANCH IF SET
002662  104024                          ERROR   24              ;ERROR, READY BIT RESET IN PRINTER
002664  052777  000100  176710  2$:     BIS     #100,@LVS       ;LOAD INTERRUPT ENABLE
002672  012737  000000  177776          MOV     #0,PSW          ;LOWER PRIORITY
002700  012737  000100  001614          MOV     #100,PSAVE      ;SET UP A DELAY LOOP
002706  005337  001614          3$:     DEC     PSAVE           ;DELAY
002712  001375                          BNE     3$              ; LOOP
002714  104025                          ERROR   25              ;ERROR, PRINTER FAILED TO INTERRUPT
002716  000401                          BR      5$              ;;BRANCH AND CLEAR INTERUPT ENABLE
002720  022626                  4$:     CMP     (SP)+,(SP)+     ;RESET STACK POINTER
002722  005077  176654          5$:     CLR     @LVS            ;CLEAR INTERRUPT ENABLE

                                ;;*****************************************************************
                                ;*TEST 12       TEST THAT THE PRINTER CAN INTERRUPT ON LEVEL INDICATED -1
                                ;;*****************************************************************
002726  000004                  TST12:  SCOPE
002730  012737  000340  177776          MOV     #340,PSW        ;LOCK-OUT
002736  012777  000240  176640          MOV     #240,@LVB       ;PRINT
002744  105777  176632          1$:     TSTB    @LVS            ;WAIT FOR A FLAG
002750  100375                          BPL     1$
002752  012777  003030  176626          MOV     #RET2,@LVVCT    ;SET UP INTERRUPT VECTOR
002760  012777  000340  176622          MOV     #340,@LVVCT1    ;
002766  052777  000100  176606          BIS     #BIT6,@LVS      ;SET INTERRUPT ENABLE
002774  013737  001624  177776          MOV     BRLEV1,PSW      ;LOAD PSW WITH LEVEL INDICATED -1
003002  013737  001574  006354          MOV     DELNUM,DLY1     ;DELAY
003010  005337  006354          2$:     DEC     DLY1
003014  001375                          BNE     2$
003016  042777  000100  176556          BIC     #BIT6,@LVS      ;REMOVE INTERRUPT ENABLE
003024  104026                          ERROR   26              ;ERROR, PRINTER FAILED TO INTERRUPT
                                                                ;ON LEVEL INDICATED -1, CHECK LV BR
                                                                ;LEVEL LOCATION FOR PROPER BR LEVEL
003026  000401                          BR      TST13           ;;GO TO NEXT TEST
003030  022626                  RET2:   CMP     (SP)+,(SP)+
```

```
                                ;;*****************************************************************
                                ;*TEST 13       TEST THAT THE PRINTER CAN NOT INTERRUPT ON LEVEL INDICATED
                                ;;*****************************************************************
003032  000004            TST13:  SCOPE
003034  012737  000340  177776        MOV     #340,PSW        ;LOCK - OUT
003042  012777  000240  176534        MOV     #240,@LVB       ;PRINT
003050  105777  176526        1$:     TSTB    @LVS            ;WAIT FOR DONE
003054  100375                        BPL     1$
003056  012777  003132  176522        MOV     #RET3,@LVVCT    ;SET UP INTERRUPT VECTOR
003064  012777  000340  176516        MOV     #340,@LVVCT1    ;
003072  052777  000100  176502        BIS     #BIT6,@LVS      ;ENABLE INTERRUPTS
003100  013737  001626  177776        MOV     BRLEV2,PSW      ;LOAD BR LEVEL INDICATED
003106  013737  001574  006354        MOV     DELNUM,DLY1     ;DELAY, EXPECT NO INTERRUPTS
003114  005337  006354        2$:     DEC     DLY1
003120  001375                        BNE     2$
003122  042777  000100  176452        BIC     #BIT6,@LVS      ;CLEAR INTERRUPT ENABLE
003130  000405                        BR      TST14           ;;GO TO NEXT TEST
003132  042777  000100  176442  RET3: BIC     #BIT6,@LVS      ;CLEAR INTERRUPT ENABLE
003140  022626                        CMP     (SP)+,(SP)+     ;POP 2 WORDS OFF THE STACK
003142  104027                        ERROR   27              ;ERROR, PRINTER INTERRUPTED ON LEVEL
                                                              ;INDICATED, CHECK LV BR LEVEL
                                                              ;FOR PROPER BR LEVEL

                                ;;*****************************************************************
                                ;*TEST 14       TEST THAT THE PRINTER CAN NOT INTERRUPT ON LEVEL INDICATED +1
                                ;;*****************************************************************
003144  000004            TST14:  SCOPE
003146  012737  000340  177776        MOV     #340,PSW        ;LOCK - OUT
003154  012777  000240  176422        MOV     #240,@LVB       ;PRINT
003162  105777  176414        1$:     TSTB    @LVS            ;WAIT FOR DONE
003166  100375                        BPL     1$
003170  012777  003244  176410        MOV     #RET4,@LVVCT    ;SET UP INTERRUPT VECTOR
003176  012777  000340  176404        MOV     #340,@LVVCT1    ;
003204  052777  000100  176370        BIS     #BIT6,@LVS      ;ENABLE INTERRUPTS
003212  013737  001630  177776        MOV     BRLEV3,PSW      ;LOAD BR LEVEL INDICATED +1
003220  013737  001574  006354        MOV     DELNUM,DLY1     ;DELAY, EXPECT NO INTERRUPTS
003226  005337  006354        2$:     DEC     DLY1
003232  001375                        BNE     2$
003234  042777  000100  176340        BIC     #BIT6,@LVS      ;CLEAR INTERRUPT ENABLE
003242  000405                        BR      TST15           ;;GO TO NEXT TEST
003244  042777  000100  176330  RET4: BIC     #BIT6,@LVS      ;CLEAR INTERRUPT ENABLE
003252  022626                        CMP     (SP)+,(SP)+     ;POP 2 WORDS OFF THE STACK
003254  104027                        ERROR   27              ;ERROR, PRINTER INTERRUPTED ON LEVEL
                                                              ;INDICATED +1, CHECK LV BR LEVEL
                                                              ;FOR PROPER BR LEVEL

                                ;;*****************************************************************
                                ;*TEST 15       CLEAN UP
                                ;;*****************************************************************
003256  000004            TST15:  SCOPE
003260  013777  001610  176320        MOV     LVVCT1,@LVVCT
003266  005077  176316                CLR     @LVVCT1
003272  000005                        RESET
003274  004737  006232                JSR     PC,CRLFX
```

```
                                ;;*****************************************************************
                                ;*TEST 16       DATA TRANSFER PATH TEST
                                ;;*****************************************************************
(2)  003300  000004             TST16:  SCOPE

003302  000240                  DTPT:   NOP
003304  004537  006360                  JSR     R5,AMSG         ;PRINT HEADING
003310  007302                          M93
003312  012737  000100  001632          MOV     #100,TEMP       ;SET-UP A COUNTER

003320  004537  006204          DTPA:   JSR     R5,DTPSR        ;SET-UP BUFFER
003324  000077                          77                      ;OCTAL '?'
003326  000100                          100                     ;OCTAL '@'

003330  004537  005602                  JSR     R5,XPRNT        ;PRINT THIS LINE
003334  001634                          WIDTH

003336  004537  006204                  JSR     R5,DTPSR        ;SET-UP BUFFER
003342  000125                          125                     ;OCTAL 'U'
003344  000052                          52                      ;OCTAL '*'

003346  004537  005602                  JSR     R5,XPRNT        ;PRINT THIS LINE
003352  001634                          WIDTH
003354  005337  001632                  DEC     TEMP            ;COMPLETED FULL COUNT?
003360  001357                          BNE     DTPA            ;BRANCH IF NOT COMPLETED

003362  004737  006232                  JSR     PC,CRLFX        ;CRLF
```

```
                                    ;;*******************************************************
                                    ;;*TEST 17      PRINTABLE AND NON-PRINTABLE CHARACTERS
                                    ;;*******************************************************
        003366  000004              TST17:  SCOPE

        003370  000240              PNPC:   NOP
        003372  004537  006360              JSR     R5,AMSG         ;PRINT HEADING
        003376  007342                      M94
        003400  012737  000040  001632      MOV     #40,TEMP        ;SET-UP PASS COUNT

                                    ;LEGAL PRINTABLE CHARACTERS

        003406  012701  000040      LPCHA:  MOV     #40,R1          ;SET-UP STARTING CHARACTER
        003412  004537  005544              JSR     R5,[illegible]  ;LOAD BUFFER WITH LEGAL CHARACTERS
        [illegible]                         K0001

        003420  004537  005602              JSR     R5,XPRNT        ;PRINT THIS LINE
        003424  001576                      K0001
        003426  012737  000001  006300      MOV     #1,CRCNT        ;LOAD COUNT
        003434  004737  006240              JSR     PC,CRLFA

                                    ;NON-PRINTING CHARACTERS

        003440  012700  015106              MOV     #BUFFER,R0      ;LOAD BUFFER POINTER
        003444  012701  000000              MOV     #0,R1           ;LOAD STARTING CHARACTER
        003450  110120              NPCLB:  MOVB    R1,(R0)+        ;LOAD A CHARACTER INTO THE BUFFER
        003452  005201              NPCLC:  INC     R1              ;INCREMENT CHARACTER
        003454  022701  000004              CMP     #4,R1           ;TEST FOR A 'EOT' CHARACTER
        003460  001774                      BEQ     NPCLC           ;BRANCH IF SAME
        003462  022701  000012              CMP     #12,R1          ;TEST FOR A 'LF' CHARACTER
        003466  001771                      BEQ     NPCLC           ;BRANCH IF SAME
        003470  022701  000014              CMP     #14,R1          ;TEST FOR A 'FF' CHARACTER
        003474  001766                      BEQ     NPCLC           ;BRANCH IF SAME
        003476  022701  000015              CMP     #15,R1          ;TEST FOR A 'CR' CHARACTER
        003502  001763                      BEQ     NPCLC           ;BRANCH IF SAME
        003504  022701  000040              CMP     #40,R1          ;TEST FOR A LEGAL CHARACTER
        003510  001357                      BNE     NPCLB           ;BRANCH IF NOT EQUAL
        003512  112720  000015              MOVB    #15,(R0)+       ;LOAD A 'CR' CHARACTER
        003516  112720  000012              MOVB    #12,(R0)+       ;LOAD A 'LF' CHARACTER

        003522  004537  005602              JSR     R5,XPRNT        ;PRINT THIS LINE
        003526  001600                      K0002

        003530  005337  001632              DEC     TEMP            ;FINISHED A FULL PASS?
        003534  001324                      BNE     LPCHA           ;BRANCH IF NOT COMPLETE

        003536  004737  006232              JSR     PC,CRLFY                ;CR-LF
```

```
                                        ;;*****************************************************************
                                        ;*TEST 20       SINGLE CHARACTER ACROSS ALL COLS. (96 CHARACTERS)
                                        ;;*****************************************************************
003542  000004                  TST20:  SCOPE

003544  000240                  SCPL:   NOP
003546  004537  006360                  JSR     R5,AMSG         ;PRINT HEADING
003552  007422                          M95
003554  012737  000040  001616          MOV     #40,STCHAR      ;SET-UP STARTING CHARACTER

003562  013701  001616          SCAC:   MOV     STCHAR,R1       ;LOAD R1= TO CHARACTER
003566  004537  005522                  JSR     R5,FILBUF       ;LOAD A BUFFER WITH
003572  001634                          WIDTH                   ; THAT CHARACTER AND WIDTH (BYTE)

003574  004537  005602                  JSR     R5,XPRNT        ;PRINT A FULL LINE FROM THE BUFFER
003600  001634                          WIDTH

003602  062737  000001  001616          ADD     #1,STCHAR       ;UPDATE THE CHARACTER
003610  022737  000200  001616          CMP     #200,STCHAR     ;TEST FOR FINAL CHARACTER
003616  001361                          BNE     SCAC            ;BRANCH IF NOT COMPLETED

003620  004737  006232                  JSR     PC,CRLFX                ;"CR-LF"

                                        ;;*****************************************************************
                                        ;*TEST 21       ROTATING CHARACTERS ACROSS ALL COLS. (96 CHARACTERS)
                                        ;;*****************************************************************
003624  000004                  TST21:  SCOPE

003626  000240                  RCPL:   NOP
003630  004537  006360                  JSR     R5,AMSG         ;PRINT HEADING
003634  007476                          M96
003636  012737  000040  001616          MOV     #40,STCHAR      ;SET-UP STARTING CHARACTER

003644  013701  001616          RCAC:   MOV     STCHAR,R1       ;LOAD R1=TO CHARACTER
003650  004537  005544                  JSR     R5,LIC          ;LOAD A BUFFER STARTING WITH
003654  001634                          WIDTH                   ; THAT CHARACTER AND WIDTH (BYTE)

003656  004537  005602                  JSR     R5,XPRNT        ;PRINT A FULL LINE FROM THE BUFFER
003662  001634                          WIDTH

003664  062737  000001  001616          ADD     #1,STCHAR       ;UPDATE THE STARTING CHARACTER
003672  023737  001620  001616          CMP     LASTCH,STCHAR   ;TEST FOR FINAL CHARACTER
003700  001361                          BNE     RCAC            ;BRANCH IF NOT COMPLETED

003702  004737  006232                  JSR     PC,CRLFX
```

```
                                ;;*****************************************************************
                                ;*TEST 22       DOUBLE WEDGE PATTERN (CHARACTER)
                                ;;*****************************************************************
003706  000004                  TST22:  SCOPE
                                ;;<LEFT WEDGE>>

003710  000240                  DBW:    NOP
003712  004537  006360                  JSR     R5,AMSG         ;PRINT HEADING
003716  007541                          M97
003720  013737  001634  003742          MOV     WIDTH,DWPAA     ;SET-UP INITIAL 1 ST CHARACTER WIDTH
003726  012737  000000  003746          MOV     #0,DWPBB        ;SET-UP INITIAL 2ND CHARACTER WIDTH

003734  004537  005672          DWPA:   JSR     R5,WDGSUB       ;GO TO DOUBLE WEDGE SUBROUTINE
003740  000177                          177                     ;OCTAL VALUE OF THE 1ST CHARACTER
003742  000120                  DWPAA:  80.                     ;1ST CHATACTER WIDTH
003744  000040                          40                      ;OCTAL VALUE OF THE 2ND CHARACER
003746  000000                  DWPBB:  0                       ;2ND CHARACTER WIDTH

003750  004537  005602                  JSR     R5,XFRNT        ;PRINT A FULL LINE
003754  001634                          WIDTH

003756  005337  003742                  DEC     DWPAA           ;DECREMENT 1ST CHARACTER WIDTH
003762  001404                          BEQ     DWRW            ;BRANCH IF COMPLETED
003764  005237  003746                  INC     DWPBB           ;INCREMENT 2ND CHARACTER WIDTH
003770  000761                          BR      DWPA            ;BRANCH BACK AND TEST NEXT LINE

                                ;;*****************************************************************
                                ;*TEST 23       DOUBLE WEDGE PATTERN
                                ;;*****************************************************************
003772  000004                  TST23:  SCOPE
                                ;;<<RIGHT WEDGE>>

003774  012737  000000  004016  DWRW:   MOV     #0,DWPCC        ;SET-UP INITIAL 1ST CHARACTER WIDTH
004002  013737  001634  004022          MOV     WIDTH,DWPDD     ;SET-UP INITIAL 2ND CHARACTER WIDTH

004010  004537  005672          DWRWA:  JSR     R5,WDGSUB       ;GO TO DOUBLE WEDGE SUBROUTINE
004014  000040                          40                      ;
004016  000000                  DWPCC:  0                       ;FIRST CHARACTER WIDTH
004020  000177                          177                     ;SECOND CHARACTER OCTAL VALUE
004022  000120                  DWPDD:  80.                     ;SECOND CHARACTER WIDTH

004024  004537  005602                  JSR     R5,XPRNT        ;PRINT A FULL LINE
004030  001634                          WIDTH

004032  005237  004016                  INC     DWPCC           ;INCREMENT FIRST CHARACTER WIDTH
004036  005337  004022                  DEC     DWPDD           ;DECREMENT SECOND CHARACTER WIDTH
004042  001362                          BNE     DWRWA           ;BRANCH IF NOT COMPLETED

004044  004737  006232                  JSR     PC,CRLFX                ;"CR-LF"
```

```
                                ;;**********************************************************************
                                ;*TEST 24        BASIC GRAPH MODE TEST
                                ;;**********************************************************************
004050  000004                  TST24:  SCOPE

004052  000240                  GMTO:   NOP
004054  004537  006360                  JSR     R5,AMSG         ;PRINT HEADING
004060  007603                          M98
004062  012737  000200  001616          MOV     #200,STCHAR     ;
004070  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 7
004074  012737  000100  001616          MOV     #100,STCHAR     ;
004102  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 6
004106  012737  000040  001616          MOV     #40,STCHAR
004114  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 5
004120  012737  000020  001616          MOV     #20,STCHAR      ;
004126  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 4
004132  012737  000010  001616          MOV     #10,STCHAR      ;
004140  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 3
004144  012737  000004  001616          MOV     #4,STCHAR       ;
004152  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 2
004156  012737  000002  001616          MOV     #2,STCHAR       ;
004164  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 1
004170  012737  000001  001616          MOV     #1,STCHAR       ;
004176  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 0

004232  004737  006232                  JSR     PC,CRLFX                ;"CR-LF"
```

```
                                ;;*****************************************************************
                                ;*TEST 25       GRAPH MODE TEST
                                ;;*****************************************************************
(2)  004206  000004             TST25:  SCOPE
                                :ACCUMULATING BIT TEST

600  004210  000240             GMT1:   NOP
601  004212  004537  006360             JSR     R5,AMSG         ;PRINT HEADING
602  004216  007662                     M99
603  004220  012737  000200  001616     MOV     #200,STCHAR     ;
604  004226  004737  006024             JSR     PC,BGMTA        ;PLOT BIT 7
605  004232  012737  000300  001616     MOV     #300,STCHAR     ;
606  004240  004737  006024             JSR     PC,BGMTA        ;PLOT BIT 7, 6
607  004244  012737  000340  001616     MOV     #340,STCHAR     ;
608  004252  004737  006024             JSR     PC,BGMTA        ;PLOT BIT 7, 6, 5
609  004256  012737  000360  001616     MOV     #360,STCHAR     ;
610  004264  004737  006024             JSR     PC,BGMTA        ;PLOT BIT 7, 6, 5, 4
611  004270  012737  000370  001616     MOV     #370,STCHAR     ;
612  004276  004737  006024             JSR     PC,BGMTA        ;PLOT BIT 7, 6, 5, 4, 3
613  004302  012737  000374  001616     MOV     #374,STCHAR     ;
614  004310  004737  006024             JSR     PC,BGMTA        ;PLOT BIT 7, 6, 5, 4, 3, 2
615  004314  012737  000376  001616     MOV     #376,STCHAR     ;
616  004322  004737  006024             JSR     PC,BGMTA        ;PLOT BIT 7, 6, 5, 4, 3, 2, 1
617  004326  012737  000377  001616     MOV     #377,STCHAR     ;
618  004334  004737  006024             JSR     PC,BGMTA        ;PLOT BIT 7, 6, 5, 4, 3, 2, 1, 0
619  004340  012737  000000  001616     MOV     #0,STCHAR       ;
620  004346  004737  006024             JSR     PC,BGMTA        ;PLOT 0
621
622                                 ;;*****************************************************************
                                ;*TEST 26       ALTERNATING ONE AND ZERO TEST
                                ;;*****************************************************************
(2)  004352  000004             TST26:  SCOPE
623
624  004354  000240             GMT2:   NOP
625  004356  004537  006360             JSR     R5,AMSG         ;PRINT HEADING
626  004362  007741                     M910
627  004364  012737  000007  001632     MOV     #7,TEMP
628  004372  012737  000377  001616 GMT21:  MOV     #377,STCHAR     ;
629  004400  004737  006024             JSR     PC,BGMTA        ;PLOT 377
630  004404  012737  000000  001616     MOV     #0,STCHAR       ;
631  004412  004737  006024             JSR     PC,BGMTA        ;PLOT 0
632  004416  005337  001632             DEC     TEMP            ;DONE ?
633  004422  001363                     BNE     GMT21
634
```

```
                                ;;*****************************************************************
                                ;*TEST 27       DIMINISHING BIT TEST
                                ;;*****************************************************************
004424  000004                  TST27:  SCOPE

004426  000240                  GMT4:   NOP
004430  004537  006360                  JSR     R5,AMSG
004434  010030                          M911
004436  012737  000377  001616          MOV     #377,STCHAR     ;
004444  004737  006024                  JSR     PC,BGMTA        ;PLOT 377
004450  012737  000376  001616          MOV     #376,STCHAR     ;
004456  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 7, 6, 5, 4, 3, 2, 1
004462  012737  000374  001616          MOV     #374,STCHAR     ;
004470  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 7, 6, 5, 4, 3, 2
004474  012737  000370  001616          MOV     #370,STCHAR     ;
004502  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 7, 6, 5, 4, 3
004506  012737  000360  001616          MOV     #360,STCHAR     ;
004514  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 7, 6, 5, 4
004520  012737  000340  001616          MOV     #340,STCHAR     ;
004526  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 7, 6, 5
004532  012737  000300  001616          MOV     #300,STCHAR     ;
004540  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 7, 6
004544  012737  000200  001616          MOV     #200,STCHAR     ;
004552  004737  006024                  JSR     PC,BGMTA        ;PLOT BIT 7
004556  012737  000000  001616          MOV     #0,STCHAR       ;
004564  004737  006024                  JSR     PC,BGMTA        ;PLOT 000

004570  004737  006232                  JSR     PC,CRLFX                ;"CR-LF"
```

```
                                ;;*****************************************************************
                                ;*TEST 30       GRAPH-MODE DATA WEDGE  (LEFT WEDGE)
                                ;;*****************************************************************
004574  000004                  TST30:  SCOPE
004576  004537  006360                  JSR     R5,AMSG         ;PRINT HEADING
004602  010106                          M912
004604  013737  001640  001642          MOV     WIDTHG,GMDW1    ;SET-UP HORIZ. COUNT
004612  005337  001642          GMDWA:  DEC     GMDW1           ;DECREMENT WIDTH COUNT
004616  012737  000377  001644          MOV     #377,GMDW2      ;LOAD A FULL GRAPH DATA
004624  012737  000001  001646          MOV     #1,GMDW3        ;LOAD A 1 BIT MASK
004632  013737  001642  001650  GMDWB:  MOV     GMDW1,GMDW4     ;LOAD A HORIZ. SUB-COUNT
004640  012700  015106                  MOV     #BUFFER,R0      ;LOAD THE BUFFER POINTER
004644  005737  001642          GMDWC:  TST     GMDW1           ;TEST FOR LAST HORIZ.
004650  001405                          BEQ     GMDWD           ;BRANCH IF LAST FRAME
004652  112720  000377                  MOVB    #377,(0)+       ;LOAD A FULL GRAPH DATA
004656  005337  001650                  DEC     GMDW4           ;DECREMENT SUB-COUNT
004662  001370                          BNE     GMDWC           ;BRANCH IF NOT COMPLETE
004664  113720  001644          GMDWD:  MOVB    GMDW2,(0)+      ;NOW LOAD THE MASKED DATA
004670  112720  000123                  MOVB    #123,(0)+       ;LOAD THE TERMINATOR CHERACTER
004674  004737  006100                  JSR     PC,GRDWS        ;PLOT THIS LINE
004700  043737  001646  001644          BIC     GMDW3,GMDW2     ;MASK THE NEW DATA
004706  000261                          SEC                     ;SET THE 'C' BIT
004710  006137  001646                  ROL     GMDW3           ;ROL TO THE NEW MASK DATA
004714  022737  000777  001646          CMP     #777,GMDW3      ;TEST FOR FINAL MASK DATA
004722  001343                          BNE     GMDWB           ;BRANCH IF MORE DATA MASKS
004724  005737  001642                  TST     GMDW1           ;TEST FOR MORE HORIZ BYTES
004730  001330                          BNE     GMDWA           ;BRANCH IF NOT COMPLETE
                                ;;<<RIGHT GRAPH WEDGE>
004732  013737  001640  001642          MOV     WIDTHG,GMDW1    ;SET-UP HORIZ COUNT
004740  005337  001642                  DEC     GMDW1           ;
004744  012737  000000  001650          MOV     #0,GMDW4        ;SET UP
004752  012737  000377  001644  GMDWH:  MOV     #377,GMDW2      ;SET-UP GRAPH DATA
004760  012737  000200  001646          MOV     #200,GMDW3
004766  012700  015105          GMDWJ:  MOV     #BUFFER,R0      ;LOAD THE BUFFER POINTER
004772  013737  001650  001652          MOV     GMDW4,GMDW5
005000  001405                          BEQ     GMDWL           ;BRANCH IF THE FIRST
005002  112720  000000          GMDWK:  MOVB    #0,(0)+         ;LOAD THE DATA BUFFER
005006  005337  001652                  DEC     GMDW5           ;DECREMENT COUNT
005012  001373                          BNE     GMDWK           ;BRANCH IF NOT FINISHED
005014  113720  001644          GMDWL:  MOVB    GMDW2,(0)+      ;LOAD THE MASKED DATA
005020  013737  001642  001654          MOV     GMDW1,GMDW6     ;LOAD REMANDER HORIZ. COUNT
005026  001405                          BEQ     GMDWN
005030  112720  000377          GMDWM:  MOVB    #377,(0)+       ;LOAD UNMASKED GRAPH DATA
005034  005337  001654                  DEC     GMDW6           ;DECREMENT COUNT
005040  001373                          BNE     GMDWM           ;BRANCH IF NOT COMPLETE LINE
005042  112720  000123          GMDWN:  MOVB    #123,(0)+       ;LOAD TERMINATOR CHAR.
005046  004737  006100                  JSR     PC,GRDWS        ;PLOT THIS LINE
005052  043737  001646  001644          BIC     GMDW3,GMDW2     ;MASK THE GRAPH DATA
005060  006237  001646                  ASR     GMDW3           ;SHIFT LEFT
005064  001340                          BNE     GMDWJ           ;MORE GRAPH DATA BYTE
005066  005237  001650                  INC     GMDW4           ;INCREMENT POSITION COUNT
005072  005337  001642                  DEC     GMDW1           ;DECREMENT TOTAL COUNT
005076  100325                          BPL     GMDWH           ;BRANCH UNTIL A BLANK LINE
005100  004737  006232                  JSR     PC,CRLFX                ;"CR-LF"
```

```
                                ;;*****************************************************************
                                ;*TEST 31       PLOT A LENGTH OF FULL GRAPH DATA
                                ;;*****************************************************************
005104  000004                  T$T31:  SCOPE
                                ;THEN DELAY APPROX. 1 SEC.
                                ;THEN REPEAT DATA 10 TIMES

005106  000240                  JBT:    NOP
005110  004537  006360                  JSR     R5,AMSG         ;PRINT HEADING
005114  010145                          M913
005116  012704  000010                  MOV     #10,R4          ;SET-UP DELAY
005122  012737  000377  001616  JBTA:   MOV     #377,STCHAR     ;SET-UP CHARACTER
005130  004737  006024                  JSR     PC,BGMTA        ;PLOT DATA
005134  004737  006322                  JSR     PC,DELAY1       ;DELAY
005140  005304                          DEC     R4              ;DECREMENT
005142  001367                          BNE     JBTA            ;BRANCH IF NOT COMPLETE
                                .SBTTL  END OF PASS ROUTINE

                                ;;*****************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO INIT

005144                          $EOP:
005144  000004                          SCOPE
005146  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
005152  005237  001200                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
005156  042737  100000  001200          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
005164  005327                          DEC     (PC)+           ;;LOOP?
005166  000001                  $EOPCT: .WORD   1
005170  003022                          BGT     $DOAGN          ;;YES
005172  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
005174  000001                  $ENDCT: .WORD   1
005176  005166                          $EOPCT
005200  104400  005245                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
005204  013746  001200                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
005210  104404                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
005212  104400  005242                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
005216  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
005222  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
005224  000005                          RESET                   ;;CLEAR THE WORLD
005226  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
005230  000240                          NOP                     ;;SAVE ROOM
005232  000240                          NOP                     ;;FOR
005234  000240                          NOP                     ;;ACT11
005236                          $DOAGN:
005236  000137                          JMP     @(PC)+          ;;RETURN
005240  002136                  $RTNAD: .WORD   INIT
005242     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
005245     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
005252  050040  051501  020123
005260  000043
```

```
                                                MANTST: RESET
                                                        RESET
005262  000005
005264  000005
005266  012706  001100                                  MOV     #STACK,SP
                                                .SBTTL  MANUAL INTERVENTION TEST
005272  013737  001246  001602                          MOV     $BASE,LVS       ;LOAD DEVICE ADDRESS
005300  013737  001246  001604                          MOV     $BASE,LVB       ;LOAD DEVICE ADDRESS
005306  062737  000002  001604                          ADD     #2,LVB          ;UPDATE BUFFER ADDRESS
005314  104400  007237                                  TYPE    ,MSG1           ;PRINT HEADER
005320  104400  006522                                  TYPE    ,MSG2           ;PRINT CONT.
005324  000000                                          HALT                    ;WAIT FOR OPERATOR
005326  005777  174250                                  TST     @LVS            ;TEST STATUS
005332  100001                                          BPL     1$              ;;BRANCH OVER
005334  104001                                          ERROR   1               ;ERROR, PRINTER ERROR BIT SET
005336  105777  174240                        1$:       TSTB    @LVS            ;TEST READY
005342  100401                                          BMI     2$              ;;BRANCH ON READY SET
005344  104002                                          ERROR   2               ;ERROR, PRINTER READY BIT
                                                                                ;FAILED TO SET
005346  104400  006724                        2$:       TYPE    .MSG4           ;PRINT 'TURN OFF POWER'
005352  104400  006651                                  TYPE    .MSG3           ;PRINT 'DEPRESS CONT'
005356  000000                                          HALT                    ;WAIT FOR OPERATOR
005360  005777  174216                                  TST     @LVS            ;TEST STATUS
005364  100401                                          BMI     3$              ;;BRANCH OVER
005366  104003                                          ERROR   3               ;ERROR, PRINTER ERROR BIT FAILED
                                                                                ; TO SET ON POWER OFF
005370  104400  006522                        3$:       TYPE    .MSG2           ;PRINT 'RESTORE'
005374  000000                                          HALT                    ;WAIT FOR OPERATOR
005376  005777  174200                                  TST     @LVS            ;TEST STATUS
005402  100001                                          BPL     4$              ;;BRANCH OVER
005404  104004                                          ERROR   4               ;ERROR, PRINTER ERROR BIT FAILED TO
                                                                                ;RESET ON POWER ON
005406  104400  006766                        4$:       TYPE    .MSG5           ;PRINT 'DISCONNECT'
005412  104400  006651                                  TYPE    ,MSG3           ;PRINT CONT
005416  000000                                          HALT                    ;WAIT FOR OPERATOR
005420  005777  174156                                  TST     @LVS            ;TEST STATUS
005424  100401                                          BMI     5$              ;;BRANCH OVER
005426  104005                                          ERROR   5               ;ERROR, PRINTER ERROR FAILED TO SET
                                                                                ; ON PRINTER DISCONNECTED
005430  104400  006522                        5$:       TYPE    .,MSG2          ;PRINT RESTORE
005434  000000                                          HALT                    ;WAIT FOR OPERATOR
005436  005777  174140                                  TST     @LVS            ;TEST STATUS
005442  100001                                          BPL     6$              ;;BRANCH OVER
005444  104006                                          ERROR   6               ;ERROR, PRINTER ERROR BIT FAILED
                                                                                ; TO RESET UPON PRINTER CONNECTED
```

```


005446 104400 007034        6$:     TYPE    ,MSG6           ;TYPE 'REMOVE PAPER'
005452 104400 006651                TYPE    ,MSG3           ;TYPE 'CONT'
005456 000000                       HALT                    ;WAIT FOR OPERATOR

005460 005777 174116                TST     @LVS            ;TEST STATUS
005464 100401                       BMI     7$              ;;BRANCH OVER
005466 104007                       ERROR   7               ;ERROR, PRINTER ERROR BIT FAILED
                                                            ; TO SET ON NO PAPER

005470 104400 006522        7$:     TYPE    ,MSG2           ;PRINT RESTORE
005474 000000                       HALT                    ;WAIT FOR OPERATOR

005476 005777 174100                TST     @LVS            ;TEST STATUS
005502 100001                       BPL     10$             ;;BRANCH OVER
005504 104010                       ERROR   10              ;ERROR, PRINTER ERROR FAILED TO
                                                            ;RESET ON PAPER AVAILABLE

005506 104400 007145        10$:    TYPE    ,MSG7           ;PRINT 'END OF THIS TEST'
005512 000005                       RESET
005514 000000                       HALT                    ;MANUAL INTERVENTION TEST COMPLETED
005516 000137 005262                JMP     MANTST          ;REPEAT TEST IF HIT CONTINUE
```

```
                                        ;LOAD A SINGLE CHARACTER ACROSS THE PAPER WIDTH
                                        ;
005522  012700  015106          FILBUF: MOV     #BUFFER,R0      ;SET-UP BUFFER POINTER
005526  013502                          MOV     @(5)+,R2        ;SET-UP PAPER WIDTH
005530  110120                  FILBFA: MOVB    R1,(0)+         ;SAVE THE CHARACTER IN THE BUFFER
005532  005302                          DEC     R2              ;FINISHED?
005534  001375                          BNE     FILBFA          ;BRANCH IF NOT COMPLETED
005536  000240                          NOP
005540  000240                          NOP
005542  000205                          RTS     R5              ;EXIT

                                        ;LOAD A INCREMENTING CHARACTER ACROSS THE PAPER WIDTH
                                        ;ONLY 40 THRU 177 ARE LEGAL CHARACTERS

005544  012700  015106          LIC:    MOV     #BUFFER,R0      ;SET-UP BUFFER POINTER
005550  013502                          MOV     @(5)+,R2        ;SET-UP WIDTH
005552  110120                  LICA:   MOVB    R1,(0)+         ;SAVE A CHARACTER IN THE BUFFER
005554  005201                          INC     R1              ;UPDATE THE CHARACTER
005556  023701  001620                  CMP     LASTCH,R1       ;TEST FOR
005562  001002                          BNE     LICB            ;BRANCH IF NOT
005564  012701  000040                  MOV     #40,R1          ;MAKE A LEGAL CHARACTER
005570  005302                  LICB:   DEC     R2              ;DECREMENT COUNT
005572  001367                          BNE     LICA            ;BRANCH IF NOT COMPLETED
005574  000240                          NOP
005576  000240                          NOP
005600  000205                          RTS     R5              ;EXIT

                                .SBTTL  PRINT/PLOT SUBROUTINE

005602  012737  015106  001656  XPRNT:  MOV     #BUFFER,SAVE1   ;SETUP BUFFER POINTER
005610  013537  001660                  MOV     @(5)+,SAVE2     ;SETUP WIDTH
005614  013700  001656          XPRNTC: MOV     SAVE1,R0
005620  013701  001660                  MOV     SAVE2,R1
005624  105777  173752          XPRNTA: TSTB    @LVS            ;TEST READY
005630  100404                          BMI     XPRNTB          ;BRANCH IF SET
005632  005777  173744                  TST     @LVS            ;TEST ERROR
005636  100372                          BPL     XPRNTA          ;BRANCH IF RESET
005640  104030                          ERROR   30              ;ERROR, ERROR FLAG SET
005642  112077  173736          XPRNTB: MOVB    (0)+,@LVB
005646  005301                          DEC     R1
005650  001365                          BNE     XPRNTA
005652  000240                          NOP
005654  000240                          NOP
005656  104406                          CKSWR                   ;TEST FOR CONTROL G
005660  032777  010000  173252          BIT     #BIT12,@SWR     ;TEST LOOP THIS LINE SWITCH
005666  001352                          BNE     XPRNTC          ;YES
005670  000205                          RTS     R5
```

```
                                        ;WEDGE SUBROUTINE <CHARACTER>
                                        
005672  012500                  WDGSUB: MOV     (5)+,R0         ;SAVE FIRST CHAR.
005674  012501                          MOV     (5)+,R1         ;SAVE FIRST COUNT
005676  012502                          MOV     (5)+,R2         ;SAVE SECOND CHAR
005700  012503                          MOV     (5)+,R3         ;SAVE SECOND COUNT
005702  012704  015106                  MOV     #BUFFER,R4      ;SETUP BUFFER POINTER
005706  005701                          TST     R1              ;TEST R1
005710  001405                          BEQ     WDGB            ;BRANCH IF NO FIRST CHAR
005712  110024                  WDGA:   MOVB    R0,(4)+         ;LOAD FIRST CHAR INTO BUFFER
005714  005301                          DEC     R1              ;DECREMENT UNTIL COMPLETED
005716  001375                          BNE     WDGA            ;BRANCH
005720  005703                          TST     R3              ;TEST R1
005722  001403                          BEQ     WDGC            ;BRANCH IF NO SECOND CHAR
005724  110224                  WDGB:   MOVB    R2,(4)+         ;LOAD SECOND CHAR INTO BUFFER
005726  005303                          DEC     R3              ;DECREMENT UNTIL COMPLETED
005730  001375                          BNE     WDGB            ;BRANCH
                                        
005732  000240                  WDGC:   NOP
005734  000240                          NOP
005736  000205                          RTS     R5              ;EXIT
                                        
                                        ;GRAPHIC SUBROUTINE
                                        
005740  012737  000100  006076  GMTA:   MOV     #64.,BGSAVE     ;SET-UP EXECUTION COUNT
005746  012700  015106                  MOV     #BUFFER,R0      ;SET-UP BUFFER POINTER
005752  052777  000001  173622          BIS     #BIT0,@LVS      ;PLOT MODE
005760  012501                          MOV     (R5)+,R1        ;GET FIRST GRAPHIC DATA
005762  012502                          MOV     (R5)+,R2        ;GET SECOND GRAPHIC DATA
005764  013703  001640                  MOV     WIDTHG,R3       ;SET-UP WIDTH COUNT
                                        
005770  110120                  GMTAA:  MOVB    R1,(0)+         ;LOAD FIRST GRAPHIC DATA
005772  110220                          MOVB    R2,(0)+         ;LOAD SECOND GRAPHIC DATA
005774  005303                          DEC     R3              ;FINISHED THE BUFFER?
005776  001374                          BNE     GMTAA           ;BRANCH IF NOT COMPLETE
                                        
006000  004537  005602          GMTAB:  JSR     R5,XPRNT        ;PRINT THIS LINE
006004  001640                          WIDTHG                  ;
                                        
006006  005337  006076                  DEC     BGSAVE          ;FINISHED THE EXECUTION COUNT
006012  001372                          BNE     GMTAB           ;BRANCH IF NOT COMPLETE
006014  042777  000001  173560          BIC     #BIT0,@LVS      ;CLEAR PLOT MODE
006022  000205                          RTS     R5              ;EXIT
```

```
                                        ;GRAPH MODE SUBROUTINE
                                        ;
006024  012737  000100  006076  BGMTA:  MOV     #64.,BGSAVE     ;SET-UP LENGTH PER GRAPH DATA
006032  052777  000001  173542          BIS     #BIT0,@LVS      ;PLOT MODE
006040  013701  001616          BGMTB:  MOV     STCHAR,R1       ;SET-UP GRAPH DATA
006044  004537  005522                  JSR     R5,FILBUF       ;FILL THE BUFFER WITH GRAPH DATA
006050  001640                          WIDTHG                  ;
006052  004537  005602                  JSR     R5,XPRNT        ;PRINT THIS LINE
006056  001640                          WIDTHG
006060  005337  006076                  DEC     BGSAVE          ;DECREMENT GRAPH LENGTH
006064  001365                          BNE     BGMTB           ;BRANCH IF NOT COMPLETED
006066  042777  000001  173506          BIC     #BIT0,@LVS      ;CLEAR PLOT MODE
006074  000207                          RTS     PC              ;EXIT
006076  000040          BGSAVE: 40                              ;TEMP LOCATION.
                                        ;WEDGE SUBROUTINE <GRAPH>
006100  012737  015106  001656  GRDWS:  MOV     #BUFFER,SAVE1
006106  052777  000001  173466          BIS     #BIT0,@LVS      ;INSURE GRAPHIC MODE
006114  013700  001656  GRDWSE: MOV     SAVE1,R0
006120  112001          GRDWSA: MOVB    (0)+,R1         ;GET A DATA WORD
006122  022701  000123                  CMP     #123,R1         ;TEST FOR TERMINATOR
006126  001412                          BEQ     GRDWSD          ;BRANCH IF COMPLETED
006130  105777  173446  GRDWSB: TSTB    @LVS            ;TEST READY
006134  100404                          BMI     GRDWSC          ;BRANCH IF READY
006136  005777  173440                  TST     @LVS            ;TEST ERROR
006142  100372                          BPL     GRDWSB
006144  104030                          ERROR   30                      ;ERROR, ERROR FLAG SET
006146  110177  173432  GRDWSC: MOVB    R1,@LVB         ;PLOT THE GRAPH DATA
006152  000762                          BR      GRDWSA
006154  052777  000002  173420  GRDWSD: BIS     #BIT1,@LVS      ;TERMINATE THE LINE
006162  104406                          CKSWR                   ;TEST FOR CONTROL G
006164  032777  010000  172746          BIT     #BIT12,@SWR     ;TEST LOOP THIS LINE SWITCH
006172  001350                          BNE     GRDWSE  ;BRANCH IF YES
006174  042777  000001  173400          BIC     #BIT0,@LVS      ;CLEAR PLOT MODE
006202  000207                          RTS     PC
                                        ;LOAD BUFFER SUBROUTINE
006204  012700  015106          DTPSR:  MOV     #BUFFER,R0
006210  012501                          MOV     (5)+,R1         ;GET FIRST CHARACTER
006212  012502                          MOV     (5)+,R2         ;GET SECOND CHARACTER
006214  013703  001636                  MOV     WIDTHB,R3       ;SET THE WIDTH
006220  110120          DTPSRA: MOVB    R1,(0)+         ;
006222  110220                  MOVB    R2,(0)+
006224  005303                  DEC     R3
006226  001374                  BNE     DTPSRA
006230  000205                  RTS     R5
```

```
006232  012737  000003  006300  CRLFX:  MOV     #3,CRCNT
006240  004737  006302          CRLFA:  JSR     PC,CRLFB           ;WAIT
006244  012737  000215  173332          MOV     #215,@LVB          ;PRINT "CR"
006252  004737  006302                  JSR     PC,CRLFB           ;WAIT
006256  012737  000212  173320          MOV     #212,@LVB          ;PRINT "LF"
006264  004737  006302                  JSR     PC,CRLFB           ;WAIT
006270  005337  006300                  DEC     CRCNT
006274  001361                          BNE     CRLFA              ;BRANCK UNTIL DONE
006276  000207                          RTS     PC                 ;EXIT

006300  000000                  CRCNT:  0

006302  105777  173274          CRLFB:  TSTB    @LVS               ;TEST FOR READY
006306  100404                          BMI     CRLFC              ;BRANCH IF SET
006310  005777  173266                  TST     @LVS               ;TEST FOR ERROR
006314  100372                          BPL     CRLFB              ;BRANCH IF NOT
006316  104030                          ERROR   30                         ;ERROR, ERROR BIT SET
006320  000207                  CRLFC:  RTS     PC                 ;EXIT

006322  013737  001572  006354  DELAY1: MOV     ADELAY,DLY1
006330  012737  000000  006356          MOV     #0,DLY2
006336  005237  006356          DLA1:   INC     DLY2
006342  001375                          BNE     DLA1
006344  005337  006354                  DEC     DLY1
006350  001372                          BNE     DLA1
006352  000207                          RTS     PC

006354  000000                  DLY1:   0
006356  000000                  DLY2:   0

                                ;HEADER SUBROUTINE FOR LV-11

006360  012500                  AMSG:   MOV     (R5)+,R0           ;GET POINTER
006362  105777  173214          AMSGA:  TSTB    @LVS               ;TEST READY
006366  100404                          BMI     AMSGB              ;BR IF YES
006370  005777  173206                  TST     @LVS               ;TEST ERROR
006374  100372                          BPL     AMSGA
006376  104030                          ERROR   30                         ;ERROR BIT SET
006400  112001                  AMSGB:  MOVB    (R0)+,R1           ;GET A BYTE
006402  001403                          BEQ     AMSGC              ;TEST FOR TERM.
006404  110177  173174                  MOVB    R1,@LVB            ;LOAD THE BYTE
006410  000764                          BR      AMSGA
006412  004737  006232          AMSGC:  JSR     PC,CRLFX
006416  000205                          RTS     R5                 ;EXIT
```

```
                                        .SBTTL  ASCII MESSAGES
                                ;
                                ;
006420 006415 046412 044501     MSG0:   .ASCII  <15><15><12>/MAINDEC-11-DZLVA-B PRINTER PLOTTER TEST/<15><12>
006426 042116 041505 030455
006434 026461 055104 053114
006442 026501 020102 051120
006450 047111 042524 020122
006456 046120 052117 042524
006464 020122 042524 052123
006472 005015
006474 042522 052123 051101             .ASCIZ  /RESTART ADDRESS 204/ <15><12>
006502 020124 042101 051104
006510 051505 020123 030062
006516 006464 000012
006522 006415 052012 051125     MSG2:   .ASCII  <15><15><12>/TURN PRINTER POWER ON/ <15><12>
006530 020116 051120 047111
006536 042524 020122 047520
006544 042527 020122 047117
006552 005015
006554 047503 047116 041505             .ASCII   CONNECT CABLE TO THE PRINTER/ <15><12>
006562 020124 040503 046102
006570 020105 047524 052040
006576 042510 050040 044522
006604 052116 051105 005015
006612 047111 042523 052122             .ASCII   INSERT PAPER INTO THE PRINTER/ <15><12>
006620 050040 050101 051105
006626 044440 052116 020117
006634 044124 020105 051120
006642 047111 042524 006522
006650      012
006651      015 042015 050105     MSG3:   .ASCIZ  <15><15> DEPRESS CONT. WHEN OPERATION COMPLETED/ <15><12>
006656 042522 051523 041440
006664 047117 027124 053440
006672 042510 020116 050117
007000 051105 052101 047511
007006 020116 047503 050115
007014 042514 042524 006504
007022 000012
007024 006415 052524 047122     MSG4:   .ASCIZ  <15><15>/TURN OFF POWER TO THE PRINTER  <15><12>
007032 047440 043106 050040
007040 053517 051105 052040
007046 020117 044124 020105
007054 051120 047111 042524
007062 006522 000012
007066 006415 044504 041523     MSG5:   .ASCIZ  <15><15>/DISCONNECT CABLE FROM THE PRINTER  <15><12>
007074 047117 042516 052103
007102 041440 041101 042514
007110 043040 047522 020115
007116 044124 020105 051120
007124 047111 042524 006522
007132 000012
```

```
 1011  007034  006415  042522  047515  MSG6:   .ASCII  <15><15>/REMOVE PAPER FROM THE PRINTER/ <15><12>
       007042  042526  050040  050101
       007050  051105  043040  047522
       007056  020115  044124  020105
       007064  051120  047111  042524
       007072  006522     012
 1012  007075     104  050105  042522          .ASCIZ  /DEPRESS ADVANCE UNTIL 'PAPER' LIGHTS/ <15><12><12>
       007102  051523  040440  053104
       007110  047101  042503  052440
       007116  052116  046111  023440
       007124  040520  042520  023522
       007132  046040  043511  052110
       007140  006523  005012     000
 1013
 1014  007145     015  046415  047101  MSG7:   .ASCII  <15><15>/MANUAL INTERVENTION SUB-TEST COMPLETED/ <15><12>
       007152  040525  020114  047111
       007160  042524  053122  047105
       007166  044524  047117  051440
       007174  041125  052055  051505
       007202  020124  047503  050115
       007210  042514  042524  006504
       007216     012
 1015  007217     104  050105  042522          .ASCIZ  /DEPRESS CONT./ <15><12>
       007224  051523  041440  047117
       007232  027124  005015     000
 1016  007237     015  006415  046412  MSG1:   .ASCIZ  <15><15><15><12>/MANUAL INTERVENTION SUB-TEST/ <15><12>
       007244  047101  040525  020114
       007252  047111  042524  053122
       007260  047105  044524  047117
       007266  051440  041125  052055
       007274  051505  006524  000012
 1017  007302  027071  020063  042040  M93:    .ASCIZ  /9.3  DATA PATH TEST <PRINT MODE>/
       007310  052101  020101  040520
       007316  044124  052040  051505
       007324  020124  050074  044522
       007332  052116  046440  042117
       007340  037105     000
 1018  007343     071  032056  020040  M94:    .ASCIZ  /9.4  PRINTABLE NON-PRINTABLE TEST <PRINT MODE>/
       007350  051120  047111  040524
       007356  046102  020105  047516
       007364  026516  051120  047111
       007372  040524  046102  020105
       007400  042524  052123  036040
       007406  051120  047111  020124
       007414  047515  042504  000076
```

```
 1020  007422  027071  020065  051440  M95:    .ASCIZ  /9.5  SINGLE CHARACTER PER LINE <PRINT MODE>/
       007430  047111  046107  020105
       007436  044103  051101  041501
       007444  042524  020123  042520
       007452  020122  044514  042516
       007460  036040  051120  047111
       007466  020124  047515  042504
       007474  000076
 1021  007476  027071  020066  051040  M96:    .ASCIZ  /9.6  ROTATING PATTERN <PRINT MODE>/
       007504  052117  052101  047111
       007512  020107  040520  052124
       007520  051105  020116  050074
       007526  044522  052116  046440
       007534  042117  037105     000
 1022  007541     071  033456  020040  M97:    .ASCIZ  /9.7  CHARACTER WEDGE <PRINT MODE>/
       007546  044103  051101  041501
       007554  042524  020122  042527
       007562  043504  020105  050074
       007570  044522  052116  046440
       007576  042117  037105     000
 1023  007603     071  034056  020040  M98:    .ASCIZ  /9.8  ROTATING ONE BIT DATA PATTERN <PLOT MODE>/
       007610  047522  040524  044524
       007616  043516  047440  042516
       007624  041040  052111  042040
       007632  052101  020101  040520
       007640  052124  051105  020116
       007646  050074  047514  020124
       007654  047515  042504  000076
 1024  007662  027071  020071  040440  M99:    .ASCIZ  /9.9  ACCUMULATING BIT DATA PATTERN <PLOT MODE>/
       007670  041503  046525  046125
       007676  052101  047111  020107
       007704  044502  020124  040504
       007712  040524  050040  052101
       007720  042524  047122  036040
       007726  046120  052117  046440
       007734  042117  037105     000
 1025  007741     071  030456  020060  M910:   .ASCIZ  /9.10 ALTERNATING ONE AND ZERO DATA PATTERN <PLOT MODE>/
       007746  046101  042524  047122
       007754  052101  047111  020107
       007762  047117  020105  047101
       007770  020104  042532  047522
       007776  042040  052101  020101
       010004  040520  052124  051105
       010012  020116  050074  047514
       010020  020124  047515  042504
       010026  000076
 1026  010030  027071  030461  042040  M911:   .ASCIZ  /9.11 DIMINISHING BIT DATA PATTERN <PLOT MODE>/
       010036  046511  047111  051511
       010044  044510  043516  041040
       010052  052111  042040  052101
       010060  020101  040520  052124
       010066  051105  020116  050074
       010074  047514  020124  047515
       010102  042504  000076
```

```
 1028  010106  027071  031061  043440  M912:   .ASCIZ  /9.12 GRAPHIC WEDGE <PLOT MODE>/
       010114  040522  044120  041511
       010122  053440  042105  042507
       010130  036040  046120  052117
       010136  046440  042117  037105
       010144     000
 1029  010145     071  030456  020063  M913:   .ASCIZ  /9.13 PAPER SLEW <PLOT MODE>/
       010152  040520  042520  020122
       010160  046123  053505  036040
       010166  046120  052117  046440
       010174  042117  037105     000
 1030  010201     015  050012  044522  EM1:    .ASCIZ  <15><12>/PRINTER ERROR BIT SET/<15><12>
       010206  052116  051105  042440
       010214  051122  051117  041040
       010222  052111  051440  052105
       010230  005015     000
 1031  010233     015  050012  044522  EM2:    .ASCIZ  <15><12>/PRINTER READY BIT FAILED TO SET/<15><12>
       010240  052116  051105  051040
       010246  040505  054504  041040
       010254  052111  043040  044501
       010262  042514  020104  047524
       010270  051440  052105  005015
       010276     000
 1032  010277     015  050012  044522  EM3:    .ASCIZ  <15><12>/PRINTER ERROR BIT FAILED TO SET ON POWER OFF/<15><12>
       010304  052116  051105  042440
       010312  051122  051117  041040
       010320  052111  043040  044501
       010326  042514  020104  047524
       010334  051440  052105  047440
       010342  020116  047520  042527
       010350  020122  043117  006506
       010356  000012
 1033  010360  005015  051120  047111  EM4:    .ASCIZ  <15><12>/PRINTER ERROR BIT FAILED TO SET ON POWER ON/<15><12>
       010366  042524  020122  051105
       010374  047522  020122  044502
       010402  020124  040506  046111
       010410  042105  052040  020117
       010416  042523  020124  047117
       010424  050040  053517  051105
       010432  047440  006516  000012
 1034  010440  005015  051120  047111  EM5:    .ASCIZ  <15><12>/PRINTER ERROR BIT FAILED TO SET ON PRINTER DISCONNECTED/<15><12
       010446  042524  020122  051105
       010454  047522  020122  044502
       010462  020124  040506  046111
       010470  042105  052040  020117
       010476  042523  020124  047117
       010504  050040  044522  052116
       010512  051105  042040  051511
       010520  047503  047116  041505
       010526  042524  006504  000012
```

```
 1036  010534  005015  051120  047111  EM6:    .ASCIZ  <15><12>/PRINTER ERROR BIT FAILED TO RESET UPON PRINTER CONNECTED/<15><1
       010542  042524  020122  051105
       010550  047522  020122  044502
       010556  020124  040506  046111
       010564  042105  052040  020117
       010572  042522  042523  020124
       010600  050125  047117  050040
       010606  044522  052116  051105
       010614  041440  047117  042516
       010622  052103  042105  005015
       010630     000
 1037  010631     015  050012  044522  EM7:    .ASCIZ  <15><12>/PRINTER ERROR BIT FAILED TO SET ON NO PAPER/<15><12>
       010636  052116  051105  042440
       010644  051122  051117  041040
       010652  052111  043040  044501
       010660  042514  020104  047524
       010666  051440  052105  047440
       010674  020116  047516  050040
       010702  050101  051105  005015
       010710     000
 1038  010711     015  050012  044522  EM10:   .ASCIZ  <15><12>/PRINTER ERROR BIT FAILED TO RESET ON PAPER AVAILABLE/<15><12>
       010716  052116  051105  042440
       010724  051122  051117  041040
       010732  052111  043040  044501
       010740  042514  020104  047524
       010746  051040  051505  052105
       010754  047440  020116  040520
       010762  042520  020122  053101
       010770  044501  040514  046102
       010776  006500  000012
 1039  011002  005[illegible]3  042522  042523  EM11:   .ASCIZ  <15><12>/RESET FAILED TO CLEAR LP ERROR BIT/<15><12>
       011010  020124  040506  046111
       011016  042105  052040  020117
       011024  046103  040505  020122
       011032  050114  042440  051122
       011040  051117  041040  052111
       011046  005015     000
 1040  011051     015  051012  051505  EM12:   .ASCIZ  <15><12>/RESET FAILED TO SET LP READY BIT/<15><12>
       011056  052105  043040  044501
       011064  042514  020104  047524
       011072  051440  052105  046040
       011100  020120  042522  042101
       011106  020131  044502  006524
       011114  000012
 1041  011116  005015  047111  051124  EM13:   .ASCIZ  <15><12>/INTRPT. EN. BIT FAILED TO SET CLEAR LP STATUS/<15><12>
       011124  052120  020056  047105
       011132  020056  044502  020124
       011140  040506  046111  042105
       011146  052040  020117  042523
       011154  020124  046103  040505
       011162  020122  050114  051440
       011170  042524  052524  006523
       011176  000012
```

```
 1043  011200  005015  047111  051124  EM14:   .ASCIZ  <15><12>/INTRPT. EN. FAILED TO RESET/<15><12>
       011206  052120  020056  047105
       011214  020056  040506  046111
       011222  042105  052040  020117
       011230  042522  042523  006524
       011236  000012
 1044  011240  005015  047515  042504  EM15:   .ASCIZ  <15><12>/MODE FAILED TO SET/<15><12>
       011246  043040  044501  042514
       011254  020104  047524  051440
       011262  052105  005015      000
 1045  011267      015  046412  042117  EM16:   .ASCIZ  <15><12>/MODE FAILED TO CLEAR/<15><12>
       011274  020105  040506  046111
       011302  042105  052040  020117
       011310  046103  040505  006522
       011316  000012
 1046  011320  005015  042522  042101  EM17:   .ASCIZ  <15><12>/READY FLAG CLEARED IN ERROR/<15><12>
       011326  020131  046106  043501
       011334  041440  042514  051101
       011342  042105  044440  020116
       011350  051105  047522  006522
       011356  000012
 1047  011360  005015  042522  042101  EM20:   .ASCIZ  <15><12>/READY FAILED TO RESET UPON LOADING PRINTER BUFFER/<15><12>
       011366  020131  040506  046111
       011374  042105  052040  020117
       011402  042522  042523  020124
       011410  050125  047117  046040
       011416  040517  044504  043516
       011424  050040  044522  052116
       011432  051105  041040  043125
       011440  042506  006522  000012
 1048  011446  005015  051105  047522  EM21:   .ASCIZ  <15><12>/ERROR/<15><12>
       011454  006522  000012
 1049  011460  005015  042522  042101  EM22:   .ASCIZ  <15><12>/READY FAILED TO SET AFTER A DELAY/<15><12>
       011466  020131  040506  046111
       011474  042105  052040  020117
       011502  042523  020124  043101
       011510  042524  020122  020101
       011516  042504  040514  006531
       011524  000012
 1050  011526  005015  051105  047522  EM23:   .ASCIZ  <15><12>/ERROR BIT SET IN PRINTER/<15><12>
       011534  020122  044502  020124
       011542  042523  020124  047111
       011550  050040  044522  052116
       011556  051105  005015      000
 1051  011563      015  051012  040505  EM24:   .ASCIZ  <15><12>/READY BIT RESET IN PRINTER/<15><12>
       011570  054504  041040  052111
       011576  051040  051505  052105
       011604  044440  020116  051120
       011612  047111  042524  006522
       011620  000012
```

```
 1053  011622  005015  051120  047111  EM25:   .ASCIZ  <15><12>/PRINTER FAILED TO INTERRUPT/<15><12>
       011630  042524  020122  040506
       011636  046111  042105  052040
       011644  020117  047111  042524
       011652  051122  050125  006524
       011660  000012
 1054  011662  005015  051120  047111  EM26:   .ASCIZ  <15><12>/PRINTER FAILED TO INTERRUPT ON LEVEL INDICATED/<15><12>
       011670  042524  020122  040506
       011676  046111  042105  052040
       011704  020117  047111  042524
       011712  051122  050125  020124
       011720  047117  046040  053105
       011726  046105  044440  042116
       011734  041511  052101  042105
       011742  005015      000
 1055  011745      015  050012  044522  EM27:   .ASCIZ  <15><12>/PRINTER INTERRUPTED ON LEVEL INDICATED/<15><12>
       011752  052115  051105  044440
       011760  052116  051105  052522
       011766  052120  042105  047440
       011774  020116  042514  042526
       012002  020114  047111  044504
       012010  040503  042524  006504
       012016  000012
 1056  012020  005015  051105  047522  EM30:   .ASCIZ  <15><12>/ERROR BIT UNEXPECTEDLY SET/<15><12>
       012026  020122  044502  020124
       012034  047125  054105  042520
       012042  052103  042105  054514
       012050  051440  052105  005015
       012056      000
 1057  012057      015  052012  040522  EM31:   .ASCIZ  <15><12>/TRAP TO 4 WHEN ADRESSING LV11/<15><12>
       012064  020120  047524  032040
       012072  053440  042510  020116
       012100  042101  042522  051523
       012106  047111  020107  053114
       012114  030461  005015      000
 1058  012121      015  042412  051122  DH1:    .ASCIZ  <15><12>/ERRPC   BUSADR  BUSVECT/<15><12>
       012126  041520  020040  041040
       012134  051525  042101  020122
       012142  041040  051525  042526
       012150  052103  005015      000
 1059          012156                  .EVEN
 1060  012156  001116  001246  001606  DT1:    $ERRPC,$BASE,LVVCT,0
       012164  000000
 1061  012166  000000                  DF1:    0
```

```
1063                                    .SBTTL  TTY INPUT ROUTINE
 (1)
 (2)                                    ;;*****************************************************************
 (1)                                    .ENABL  LSB
 (1)
 (2)                                    ;;*****************************************************************
 (1)                                    ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
 (1)                                    ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
 (1)                                    ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
 (1)                                    ;*WHEN OPERATI. ` IN TTY FLAG MODE.
 (1)  012170  022737  000176  001140  $CKSWR: CMP     @SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
 (1)  012176  001074                         BNE     15$             ;;BRANCH IF NO
 (1)  012200  105777  166740                 TSTB    @$TKS           ;;CHAR THERE?
 (1)  012204  100071                         BPL     15$             ;;IF NO, DON'T WAIT AROUND
 (1)  012206  117746  166734                 MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
 (1)  012212  042716  177600                 BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
 (1)  012216  022726  000007                 CMP     #7,(SP)+        ;;IS IT A CONTROL G?
 (1)  012222  001062                         BNE     15$             ;;NO, RETURN TO USER
 (1)  012224  123727  001134  000001         CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
 (1)  012232  001456                         BEQ     15$             ;;BRANCH IF YES
 (1)
 (1)  012234  104400  012715                 TYPE    .$CNTLG         ;;ECHO THE CONTROL-G (^G)
 (1)  012240  104400  012722          $GTSWR: TYPE    .$MSWR          ;;TYPE CURRENT CONTENTS
 (2)  012244  013746  000176                 MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
 (2)  012250  104401                         TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 (1)  012252  104400  012733                 TYPE    .$MNEW          ;;PROMPT FOR NEW SWR
 (1)  012256  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
 (1)  012260  005046                          CLR     -(SP)           ;;THE NEW SWR
 (1)  012262  105777  166656          7$:     TSTB    @$TKS           ;;CHAR THERE?
 (1)  012266  100375                         BPL     7$              ;;IF NOT TRY AGAIN
 (1)
 (1)  012270  117746  166652                 MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
 (1)  012274  042716  177600                 BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 (1)
 (1)
 (1)
 (1)  012300  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
 (1)  012304  001005                         BNE     10$             ;;BRANCH IF NOT
 (1)  012306  104400  012710                 TYPE    .$CNTLU         ;;YES, ECHO CONTROL-U (^U)
 (1)  012312  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
 (1)  012316  000757                         BR      19$             ;;LET'S TRY IT AGAIN
 (1)
 (1)
 (1)  012320  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
 (1)  012324  001022                         BNE     16$             ;;BRANCH IF NO
 (1)  012326  005766  000004                 TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
 (1)  012332  001403                         BEQ     11$             ;;BRANCH IF YES
 (1)  012334  016677  000002  166576         MOV     2(SP),@SWR      ;;SAVE NEW SWR
 (1)  012342  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
 (1)  012346  104400  001167          14$:    TYPE    .$CRLF          ;;ECHO <CR> AND <LF>
 (1)  012352  123727  001135  000001         CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
 (1)  012360  001003                         BNE     15$             ;;BRANCH IF NOT
 (1)  012362  012777  000100  166554         MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
 (1)  012370  000002                  15$:    RTI                     ;;RETURN
```

```
(1)  012372  004737  013156          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
(1)  012376  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
(1)  012402  002420                          BLT     18$             ;;BRANCH IF YES
(1)  012404  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
(1)  012410  003015                          BGT     18$             ;;BRANCH IF YES
(1)  012412  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
(1)  012416  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
(1)  012422  001403                          BEQ     17$             ;;BRANCH IF YES
(1)  012424  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
(1)  012426  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
(1)  012430  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
(1)  012432  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
(1)  012436  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
(1)  012442  000707                          BR      7$              ;;GET THE NEXT ONE
(1)  012444  104400  001166          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
(1)  012450  000720                          BR      20$             ;;SIMULATE CONTROL-U
(1)                                  .DSABL  LSB
(1)
(1)
(2)                                  ;;**********************************************************************
(1)                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                  ;*CALL:
(1)                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
(1)                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
(1)                                  ;
(1)
(1)  012452  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
(1)  012454  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
(1)  012462  105777  166456          1$:     TSTB    @$TKS           ;;WAIT FOR
(1)  012466  100375                          BPL     1$              ;;A CHARACTER
(1)  012470  117766  166452  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
(1)  012476  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
(1)  012504  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
(1)  012512  001013                          BNE     3$              ;;BRANCH IF NO
(1)  012514  105777  166424          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
(1)  012520  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
(1)  012522  117746  166420                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
(1)  012526  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
(1)  012532  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
(1)  012536  001366                          BNE     2$              ;;IF NOT DISCARD IT
(1)  012540  000750                          BR      1$              ;;YES, RESUME
(1)  012542  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
(1)  012550  002407                          BLT     4$              ;;BRANCH IF YES
(1)  012552  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
(1)  012560  003003                          BGT     4$              ;;BRANCH IF YES
(1)  012562  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
(1)  012570  000002                  4$:     RTI                     ;;GO BACK TO USER
(2)                                  ;;**********************************************************************
(1)                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1)                                  ;*CALL:
(1)                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
(1)                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1)                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
```

```
                    
(1)                 
(1)  012572 010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
(1)  012574 012703  012700          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
(1)  012600 022703  012710          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
(1)  012604 101405                          BLOS    4$              ;;BR IF YES
(1)  012606 104407                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
(1)  012610 112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
(1)  012612 122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
(1)  012616 001003                          BNE     3$              ;;SKIP IF NOT
(1)  012620 104400  001166          4$:     TYPE    ,$QUES          ;;TYPE A '?'
(1)  012624 000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
(1)  012626 111337  012676          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
(1)  012632 104400  012676                  TYPE    ,9$
(1)  012636 122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
(1)  012642 001356                          BNE     2$              ;;LOOP IF NOT RETURN
(1)  012644 105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
(1)  012650 104400  001170                  TYPE    ,$LF            ;;TYPE A LINE FEED
(1)  012654 012603                          MOV     (SP)+,R3        ;;RESTORE R3
(1)  012656 011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
(1)  012660 016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
(1)  012666 012766  012700  000004          MOV     #$TTYIN,4(SP)
(1)  012674 000002                          RTI                     ;;RETURN
(1)  012676    000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
(1)  012677    000                          .BYTE   0               ;;TERMINATOR
(1)  012700 000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
(1)  012710 052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
(1)  012715    136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
(1)  012722 005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
(1)  012730 020075     000
(1)  012733    040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
(1)  012740 036440  000040
1064                                .SBTTL  TYPE ROUTINE
(1)
(2)                                 ;;**************************************************************
(1)                                 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1)                                 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1)                                 ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1)                                 ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1)                                 ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1)                                 ;*
(1)                                 ;*CALL:
(1)                                 ;*1) USING A TRAP INSTRUCTION
(1)                                 ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(1)                                 ;*OR
(1)                                 ;*      TYPE
(1)                                 ;*      MESADR
(1)                                 ;*
(1)
(1)  012744  105737  001157         $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
(1)  012750  100002                         BPL     1$              ;;BR IF YES
(1)  012752  000000                         HALT                    ;;HALT HERE IF NO TERMINAL
(1)  012754  000430                         BR      3$              ;;LEAVE
(1)  012756  010046                 1$:     MOV     R0,-(SP)        ;;SAVE R0
(1)  012760  017600  000002                 MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
```

```
(1)  012764  122737  000001  001212          CMPB    #APTENV,$ENV      ;;RUNNING IN APT MODE
(1)  012772  001011                           BNE     62$               ;;NO,GO CHECK FOR APT CONSOLE
(1)  012774  132737  000100  001213          BITB    #APTSPOOL,$ENVM   ;;SPOOL MESSAGE TO APT
(1)  013002  001405                           BEQ     62$               ;;NO,GO CHECK FOR CONSOLE
(1)  013004  010037  013014                   MOV     R0,61$            ;;SETUP MESSAGE ADDRESS FOR APT
(1)  013010  004737  013706                   JSR     PC,$ATY3          ;;SPOOL MESSAGE TO APT
(1)  013014  000000                  61$:     .WORD   0                 ;;MESSAGE ADDRESS
(1)  013016  132737  000040  001213  62$:     BITB    #APTCSUP,$ENVM    ;;APT CONSOLE SUPPRESSED
(1)  013024  001003                           BNE     60$               ;;YES,SKIP TYPE OUT
(1)  013026  112046                  2$:      MOVB    (R0)+,-(SP)       ;;PUSH CHARACTER TO BE TYPED ONTO STACK
(1)  013030  001005                           BNE     4$                ;;BR IF IT ISN'T THE TERMINATOR
(1)  013032  005726                           TST     (SP)+             ;;IF TERMINATOR POP IT OFF THE STACK
(1)  013034  012600                  60$:     MOV     (SP)+,R0          ;;RESTORE R0
(1)  013036  062716  000002                   ADD     #2,(SP)           ;;ADJUST RETURN PC
(1)  013042  000002                           RTI                       ;;RETURN
(1)  013044  122716  000011          4$:      CMPB    #HT,(SP)          ;;BRANCH IF <HT>
(1)  013050  001430                           BEQ     8$
(1)  013052  122716  000200                   CMPB    #CRLF,(SP)        ;;BRANCH IF NOT <CR_F>
(1)  013056  001006                           BNE     5$
(1)  013060  005726                           TST     (SP)+             ;;POP  <CR><LF> EQUIV
(1)  013062  104400                           TYPE                      ;;TYPE A CR AND LF
(1)  013064  001167                           $CRLF
(1)  013066  105037  013222                   CLRB    $CHARCNT          ;;CLEAR CHARACTER COUNT
(1)  013072  000755                           BR      2$                ;;GET NEXT CHARACTER
(1)  013074  004737  013156          5$:      JSR     PC,$TYPEC         ;;GO TYPE THIS CHARACTER
(1)  013100  123726  001156          6$:      CMPB    $FILLC,(SP)+      ;;IS IT TIME FOR FILLER CHARS.?
(1)  013104  001350                           BNE     2$                ;;IF NO GO GET NEXT CHAR.
(1)  013106  013746  001154                   MOV     $NULL,-(SP)       ;;GET # OF FILLER CHARS. NEEDED
(1)                                                                     ;;AND THE NULL CHAR.
(1)  013112  105366  000001          7$:      DECB    1(SP)             ;;DOES A NULL NEED TO BE TYPED?
(1)  013116  002770                           BLT     6$                ;;BR IF NO--GO POP THE NULL OFF OF STACK
(1)  013120  004737  013156                   JSR     PC,$TYPEC         ;;GO TYPE A NULL
(1)  013124  105337  013222                   DECB    $CHARCNT          ;;DO NOT COUNT AS A COUNT
(1)  013130  000770                           BR      7$                ;;LOOP
(1)
(1)                                  ;HORIZONTAL TAB PROCESSOR
(1)
(1)  013132  112716  000040          8$:      MOVB    #' ,(SP)          ;;REPLACE TAB WITH SPACE
(1)  013136  004737  013156          9$:      JSR     PC,$TYPEC         ;;TYPE A SPACE
(1)  013142  132737  000007  013222           BITB    #7,$CHARCNT       ;;BRANCH IF NOT AT
(1)  013150  001372                           BNE     9$                ;;TAB STOP
(1)  013152  005726                           TST     (SP)+             ;;POP SPACE OFF STACK
(1)  013154  000724                           BR      2$                ;;GET NEXT CHARACTER
(1)  013156  105777  165766          $TYPEC:  TSTB    @$TPS             ;;WAIT UNTIL PRINTER IS READY
(1)  013162  100375                           BPL     $TYPEC
(1)  013164  116677  000002  165760           MOVB    2(SP),@$TPB       ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1)  013172  122766  000015  000002           CMPB    #CR,2(SP)         ;;IS CHARACTER A CARRIAGE RETURN?
(1)  013200  001003                           BNE     1$                ;;BRANCH IF NO
(1)  013202  105037  013222                   CLRB    $CHARCNT          ;;YES--CLEAR CHARACTER COUNT
(1)  013206  000406                           BR      $TYPEX            ;;EXIT
(1)  013210  122766  000012  000002  1$:      CMPB    #LF,2(SP)         ;;IS CHARACTER A LINE FEED?
(1)  013216  001402                           BEQ     $TYPEX            ;;BRANCH IF YES
(1)  013220  105227                           INCB    (PC)+             ;;COUNT THE CHARACTER
(1)  013222  000000          $CHARCNT:.WORD   0                         ;;CHARACTER COUNT STORAGE
```

```
                                $TYPEX: RTS     PC

                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

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

C13226  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
013232  116637  000001  013451         MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
013240  112637  013453                 MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
013244  062716  000002                 ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
013250  000406                         BR      $TYPON
013252  112737  000001  013451 $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
013260  112737  000006  013453         MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
013266  112737  000005  013450 $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
013274  010346                         MOV     R3,-(SP)        ;;SAVE R3
013276  010446                         MOV     R4,-(SP)        ;;SAVE R4
013300  010546                         MOV     R5,-(SP)        ;;SAVE R5
013302  113704  013453                 MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
013306  005404                         NEG     R4
013310  062704  000006                 ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
013314  110437  013452                 MOVB    R4,$OMODE       ;;SAVE IT FOR USE
013320  113704  013451                 MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
013324  016605  000012                 MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
013330  005003                         CLR     R3              ;;CLEAR THE OUTPUT WORD
013332  006105                 1$:     ROL     R5              ;;ROTATE MSB INTO "C"
013334  000404                         BR      3$              ;;GO DO MSB
013336  006105                 2$:     ROL     R5              ;;FORM THIS DIGIT
013340  006105                         ROL     R5
013342  006105                         ROL     R5
013344  010503                         MOV     R5,R3
013346  006103                 3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
013350  105337  013452                 DECB    $OMODE          ;;TYPE THIS DIGIT?
013354  100016                         BPL     7$              ;;BR IF NO
```

BOS


```
(1)   013356  042703  177770            BIC     #177770,R3      ;;GET RID OF JUNK
(1)   013362  001002                    BNE     4$              ;;TEST FOR 0
(1)   013364  005704                    TST     R4              ;;SUPPRESS THIS 0?
(1)   013366  001403                    BEQ     5$              ;;BR IF YES
(1)   013370  005204            4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
(1)   013372  052703  000060            BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
(1)   013376  052703  000040    5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
(1)   013402  110337  013446            MOVB    R3,8$           ;;SAVE FOR TYPING
(1)   013406  104400  013446            TYPE    ,8$             ;;GO TYPE THIS DIGIT
(1)   013412  105337  013450    7$:     DECB    $OCNT           ;;COUNT BY 1
(1)   013416  003347                    BGT     2$              ;;BR IF MORE TO DO
(1)   013420  002402                    BLT     6$              ;;BR IF DONE
(1)   013422  005204                    INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
(1)   013424  000743                    BR      2$              ;;GO DO THE LAST DIGIT
(1)   013426  012605            6$:     MOV     (SP)+,R5        ;;RESTORE R5
(1)   013430  012604                    MOV     (SP)+,R4        ;;RESTORE R4
(1)   013432  012603                    MOV     (SP)+,R3        ;;RESTORE R3
(1)   013434  016666  000002  000004    MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
(1)   013442  012616                    MOV     (SP)+,(SP)
(1)   013444  000002                    RTI                     ;;RETURN
(1)   013446     000            8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
(1)   013447     000                    .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
(1)   013450     000            $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
(1)   013451     000            $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
(1)   013452  000000            $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

(2)                             ;;***********************************************************
(2)                             ;;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
(2)                             ;;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
(2)                             ;;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
(2)                             ;;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
(2)                             ;;*REPLACED WITH SPACES.
(2)                             ;;*CALL:
(2)                             ;;*     MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
(2)                             ;;*     TYPDS                   ;;GO TO THE ROUTINE

(2)   013454                    $TYPDS:
(2)   013454  010046                    MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(2)   013456  010146                    MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(2)   013460  010246                    MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(2)   013462  010346                    MOV     R3,-(SP)        ;;PUSH R3 ON STACK
(2)   013464  010546                    MOV     R5,-(SP)        ;;PUSH R5 ON STACK
(2)   013466  012746  020200            MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
(2)   013472  016605  000020            MOV     20(SP),R5       ;;GET THE INPUT NUMBER
(2)   013476  100004                    BPL     1$              ;;BR IF INPUT IS POS.
(2)   013500  005405                    NEG     R5              ;;MAKE THE BINARY NUMBER POS.
(2)   013502  112766  000055  000001    MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
(2)   013510  005000            1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
(2)   013512  012703  013670            MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
(2)   013516  112723  000040            MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
(2)   013522  005002            2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
(2)   013524  016001  013660            MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
(2)   013530  160105            3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
```

```
                013532  002402                          BLT     4$              ;;BR IF DONE
                013534  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
                013536  000774                          BR      3$
                013540  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
                013542  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
                013544  001002                          BNE     5$              ;;FALL THROUGH IF 0
                013546  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
                013550  100403                          BMI     7$              ;;BR IF YES
                013552  106316                  5$:     ASLB    (SP)            ;;MSD?
                013554  103003                          BCC     6$              ;;BR IF NO
                013556  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
                013564  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
                013570  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
                013574  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
                013576  005720                          TST     (R0)+           ;;JUST INCREMENTING
                013600  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
                013604  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
                013606  000302                          BGT     8$              ;;GO TO EXIT
                013610  010502                          MOV     R5,R2           ;;GET THE LSD
                013612  000764                          BR      6$              ;;GO CHANGE TO ASCII
                013614  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
                013616  100003                          BPL     9$              ;;BR IF NO
                013620  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
                013626  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
                013630  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
                013632  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
                013634  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
                013636  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
                013640  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
                013642  104400  013670                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
                013646  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
                013654  012616                          MOV     (SP)+,(SP)
                013656  000002                          RTI                     ;;RETURN TO USER
                013660  023420                  $CTBL:  10000.
                013662  001750                          1000.
                013664  000144                          100.
                013666  000012                          10.
                013670  000004                  $DBLK:  .BLKW   4
1067                                            .SBTTL  APT COMMUNICATIONS ROUTINE
(1)
(2)                                             ;;*****************************************************************
(1)             013700  112737  000001  014144  $ATY1:: MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
(1)             013706  112737  000001  014142  $ATY3:: MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
(1)             013714  000403                          BR      $ATYC
(1)             013716  112737  000000  014...  $ATY4:: MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
(1)             013724                          $ATYC::
(1)             013724  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(1)             013726  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(1)             013730  105737  014142                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
(1)             013734  001453                          BEQ     5$              ;;IF NOT:  BR
(1)             013736  122737  000001  001212          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
(1)             013744  001031                          BNE     3$              ;;IF NOT:  BR
(1)             013746  132737  000100  001213          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
(1)             013754  001425                          BEQ     3$              ;;IF NOT:  BR
```

```
(1)  013756  017600  000004                    MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
(1)  013762  062766  000002  000004            ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
(1)  013770  005737  001172            1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
(1)  013774  001375                            BNE     1$              ;;IF NOT:  WAIT
(1)  013776  010037  001206                    MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX .
(1)  014002  105720                    2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
(1)  014004  001376                            BNE     2$
(1)  014006  163700  001206                    SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
(1)  014012  006200                            ASR     R0              ;;GET MESSAGE LNGTH IN WCRDS
(1)  014014  010037  001210                    MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
(1)  014020  012737  000004  001172            MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
(1)  014026  000413                            BR      5$
(1)  014030  017637  000004  014054    3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
(1)  014036  062766  000002  000004            ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
(3)  014044  013746  177776                    MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
(1)  014050  004737  012744                    JSR     PC,$TYPE        ;;CALL TYPE MACRO
(1)  014054  000000                    4$:     .WORD   0
(1)  014056                            5$:
(1)  014056  105737  014144            10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
(1)  014062  001416                            BEQ     12$             ;;IF NOT:  BR
(1)  014064  005737  001212                    TST     $ENV            ;;RUNNING UNDER APT?
(1)  014070  001413                            BEQ     12$             ;;IF NOT:  BR
(1)  014072  005737  001172            11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
(1)  014076  001375                            BNE     11$             ;;IF NOT:  WAIT
(1)  014100  017637  000004  001174            MOV     @4(SP),$FATAL   ;;GET ERROR #
(1)  014106  062766  000002  000004            ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
(1)  014114  005237  001172                    INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
(1)  014120  105037  014144            12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
(1)  014124  105037  014143                    CLRB    $LFLG           ;;CLEAR LOG FLAG
(1)  014130  105037  014142                    CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
(3)  014134  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  014136  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  014140  000207                            RTS     PC              ;;RETURN
(1)  014142  000                       $MFLG:  .BYTE   0               ;;MESSG. FLAG
(1)  014143  000                       $LFLG:  .BYTE   0               ;;LOG FLAG
(1)  014144  000                       $FFLG:  .BYTE   0               ;;FATAL FLAG
(1)                                            .EVEN
                 014146
                 000200                APTSIZE=200
                 000001                APTENV=001
                 000100                APTSPOOL=100
                 000040                APTCSUP=040
1068                                   .SBTTL  ERROR HANDLER ROUTINE
(1)
(2)
(1)                                    ;;**************************************** ******************
(1)                                    ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
(1)                                    ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
(1)                                    ;*AND GO TO $ERRTYP ON ERROR
(1)                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1)                                    ;*SW15=1        HALT ON ERROR
(1)                                    ;*SW13=1        INHIBIT ERROR TYPEOUTS
(1)                                    ;*SW10=1        BELL ON ERROR
(1)                                    ;*SW09=1        LOOP ON ERROR
(1)                                    ;*CALL
(1)                                    ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
```

```
(1)
(1)  014146                          $ERROR:
(1)  014146  104406                          CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
(2)  014150  104406                          CKSWR
(1)  014152  105237  001103          7$:     INCB    $ERFLG                  ;;SET THE ERROR FLAG
(1)  014156  001775                          BEQ     7$                      ;;DON'T LET THE FLAG GO TO ZERO
(1)  014160  013777  001102  164754          MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER AND ERROR FLAG
(1)  014166  032777  002000  164744          BIT     #BIT10,@SWR             ;;BELL ON ERROR?
(1)  014174  001402                          BEQ     1$                      ;;NO - SKIP
(1)  014176  104401  001162                  TYPE    ,$BELL                  ;;RING BELL
(1)  014202  005237  001112          1$:     INC     $ERTTL                  ;;COUNT THE NUMBER OF ERRORS
(1)  014206  011637  001116                  MOV     (SP),$ERRPC             ;;GET ADDRESS OF ERROR INSTRUCTION
(1)  014212  162737  000002  001116          SUB     #2,$ERRPC
(1)  014220  117737  164672  001114          MOVB    @$ERRPC,$ITEMB          ;;STRIP AND SAVE THE ERROR ITEM CODE
(1)  014226  032777  020000  164704          BIT     #BIT13,@SWR             ;;SKIP TYPEOUT IF SET
(1)  014234  001004                          BNE     20$                     ;;SKIP TYPEOUTS
(1)  014236  004737  014336                  JSR     PC,$ERRTYP              ;;GO TO USER ERROR ROUTINE
(1)  014242  104400  001167                  TYPE    ,$CRLF
(1)  014246                          20$:
(1)  014246  122737  000001  001212          CMPB    #APTENV,$ENV            ;;RUNNING IN APT MODE
(1)  014254  001007                          BNE     2$                      ;;NO,SKIP APT ERROR REPORT
(1)  014256  113737  001114  014270          MOVB    $ITEMB,21$              ;;SET ITEM NUMBER AS ERROR NUMBER
(1)  014264  004737  013716                  JSR     PC,$ATY4                ;;REPORT FATAL ERROR TO APT
(1)  014270     000                  21$:    .BYTE   0
(1)  014271     000                          .BYTE   0
(1)  014272  000777                  22$:    BR      22$                     ;;APT ERROR LOOP
(1)  014274  005777  164640          2$:     TST     @SWR                    ;;HALT ON ERROR
(1)  014300  100002                          BPL     3$                      ;;SKIP IF CONTINUE
(1)  014302  000000                          HALT                            ;;HALT ON ERROR!
(1)  014304  104406                          CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
(1)  014306  032777  001000  164624  3$:     BIT     #BIT09,@SWR             ;;LOOP ON ERROR SWITCH SET?
(1)  014314  001402                          BEQ     4$                      ;;BR IF NO
(1)  014316  013716  001110                  MOV     $LPERR,(SP)             ;;FUDGE RETURN FOR LOOPING
(1)  014322  005737  001160          4$:     TST     $ESCAPE                 ;;CHECK FOR AN ESCAPE ADDRESS
(1)  014326  001402                          BEQ     5$                      ;;BR IF NONE
(1)  014330  013716  001160                  MOV     $ESCAPE,(SP)            ;;FUDGE RETURN ADDRESS FOR ESCAPE
(1)  014334                          5$:
(1)  014334  000002                          RTI                             ;;RETURN
1069                                 .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
(1)
(2)                                  ;;*************************************************************
(1)                                  ;;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
(1)                                  ;;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
(1)                                  ;;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
(1)
(1)  014336                          $ERRTYP:
(1)  014336  104400  001167                  TYPE    ,$CRLF                  ;;"CARRIAGE RETURN" & "LINE FEED"
(1)  014342  010046                          MOV     R0,-(SP)                ;;SAVE R0
(1)  014344  005000                          CLR     R0                      ;;PICKUP THE ITEM INDEX
(1)  014346  153700  001114                  BISB    @#$ITEMB,R0
(1)  014352  001004                          BNE     1$                      ;;IF ITEM NUMBER IS ZERO, JUST
(1)                                                                          ;;TYPE THE PC OF THE ERROR
(2)  014354  013746  001116                  MOV     $ERRPC,-(SP)            ;;SAVE $ERRPC FOR TYPEOUT
(2)                                                                          ;;ERROR ADDRESS
```

```
(2)  014360  104401                        TYPOC                ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
(1)  014362  000426                        BR      5$           ;;GET OUT
(1)  014364  005300                  1$:   DEC     R0           ;;ADJUST THE INDEX SO THAT IT WILL
(1)  014366  006300                        ASL     R0           ;;      WORK FOR THE ERROR TABLE
(1)  014370  006300                        ASL     R0
(1)  014372  006300                        ASL     R0
(1)  014374  062700  001254                ADD     #$ERRTB,R0   ;;FORM TABLE POINTER
(1)  014400  012037  014410                MOV     (R0)+,2$     ;;PICKUP "ERROR MESSAGE" POINTER
(1)  014404  001404                        BEQ     3$           ;;SKIP TYPEOUT IF NO POINTER
(1)  014406  104400                        TYPE                 ;;TYPE THE "ERROR MESSAGE"
(1)  014410  000000                  2$:   .WORD   0            ;;"ERROR MESSAGE" POINTER GOES HERE
(1)  014412  104400  001167                TYPE    ,$CRLF       ;;"CARRIAGE RETURN" & "LINE FEED"
(1)  014416  012037  014426          3$:   MOV     (R0)+,4$     ;;PICKUP "DATA HEADER" POINTER
(1)  014422  001404                        BEQ     5$           ;;SKIP TYPEOUT IF 0
(1)  014424  104400                        TYPE                 ;;TYPE THE "DATA HEADER"
(1)  014426  000000                  4$:   .WORD   0            ;;"DATA HEADER" POINTER GOES HERE
(1)  014430  104400  001167                TYPE    ,$CRLF       ;;"CARRIAGE RETURN" & "LINE FEED"
(1)  014434  011000                  5$:   MOV     (R0),R0      ;;PICKUP "DATA TABLE" POINTER
(1)  014436  001004                        BNE     7$           ;;GO TYPE THE DATA
(1)  014440  012600                  6$:   MOV     (SP)+,R0     ;;RESTORE R0
(1)  014442  104400  001167                TYPE    ,$CRLF       ;;"CARRIAGE RETURN" & "LINE FEED"
(1)  014446  000207                        RTS     PC           ;;RETURN
(2)  014450                          7$:
(2)  014450  013046                        MOV     @(R0)+,-(SP) ;;SAVE @(R0)+ FOR TYPEOUT
(2)  014452  104401                        TYPOC                ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
(1)  014454  005710                        TST     (R0)         ;;IS THERE ANOTHER NUMBER?
(1)  014456  001770                        BEQ     6$           ;;BR IF NO
(1)  014460  104400  014466                TYPE    ,8$          ;;TYPE TWO(2) SPACES
(1)  014464  000771                        BR      7$           ;;LOOP
(1)  014466  020040      000         8$:   .ASCIZ  /  /         ;;TWO(2) SPACES
(1)          014472                        .EVEN
.070                               .SBTTL  SCOPE HANDLER ROUTINE
(1)
(2)
(1)                                ;;**********************************************************************
(1)                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
(1)                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY:7:0)
(1)                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY(15:08)
(1)                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1)                                ;*SW14=1          LOOP ON TEST
(1)                                ;*SW09=1          LOOP ON ERROR
(1)                                ;*SW08=1          LOOP ON TEST IN SWR(7:0)
(1)                                ;*CALL
(1)                                ;*      SCOPE               ;;SCOPE=IOT
(1)
(1)  014472                        $SCOPE:
(1)  014472  104406                        CKSWR                ;;TEST  FOR CHANGE IN SOFT-SWR
(1)  014474  032777  040000  164436  1$:   BIT     #BIT14,@SWR  ;;LOOP ON PRESENT TEST?
(1)  014502  001062                        BNE     $OVER        ;;YES IF SW14=1
(1)                                ;*****START OF CODE FOR THE XOR TESTER*****
(1)  014504  000416                $XTSTR: BR      6$           ;;IF RUNNING ON THE "XOR" TESTER CHANGE
(1)                                                             ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
(1)  014506  013746  000004                MOV     @#ERRVEC,-(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
(1)  014512  012737  014532  000004        MOV     #5$,@#ERR.EC ;;SET FOR TIMEOUT
(1)  014520  005737  177060                TST     @#177060     ;;TIME OUT ON XOR?
```

```
(1)  014524  012637  000004            MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1)  014530  000431                     BR      $SVLAD          ;;GO TO THE NEXT TEST
(1)  014532  022626                5$:  CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
(1)  014534  012637  000004            MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1)  014540  000417                     BR      7$              ;;LOOP ON THE PRESENT TEST
(1)  014542                        6$:;*****END OF CODE FOR THE XOR TESTER*****
(1)  014542  032777  000400  164370     BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
(1)  014550  001404                     BEQ     2$              ;;BR IF NO
(1)  014552  127737  164362  001102     CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
(1)  014560  001433                     BEQ     $OVER           ;;BR IF YES
(1)  014562  105737  001103        2$:  TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
(1)  014566  001412                     BEQ     $SVLAD          ;;BR IF NO
(1)  014570  032777  001000  164342     BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
(1)  014576  001404                     BEQ     4$              ;;BR IF NO
(1)  014600  013737  001110  001106  7$: MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
(1)  014606  000420                     BR      $LOOP
(1)  014610  105037  001103        4$:  CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
(1)  014614  105237  001102  $SVLAD:  INCB    $TSTNM          ;;COUNT TEST NUMBERS
(1)  014620  113737  001102  001176     MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
(1)  014626  011637  001106             MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
(1)  014632  011637  001110             MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
(1)  014636  005037  001160             CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
(1)  014642  112737  000001  001115     MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1)  014650  013777  001102  164264  $OVER: MOV   $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
(1)  014656  013716  001106             MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
(1)  014662  000002                     RTI                     ;;FIXES PS
1071                               .SBTTL  POWER DOWN AND UP ROUTINES
(1)
(2)                                ;;**************************************************************
(1)                                ;POWER DOWN ROUTINE
(1)  014664  012737  015024  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
(1)  014672  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
(3)  014700  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  014702  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(3)  014704  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(3)  014706  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
(3)  014710  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
(3)  014712  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
(3)  014714  017746  164220                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
(1)  014720  010637  015030                MOV     SP,$SAVR5       ;;SAVE SP
(1)  014724  012737  014736  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
(1)  014732  000000                        HALT
(1)  014734  000776                        BR      .-2             ;;HANG UP
(1)
(2)                                ;;**************************************************************
(1)                                ;POWER UP ROUTINE
(1)  014736  012737  015024  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
(1)  014744  013706  015030                MOV     $SAVR6,SP       ;;GET SP
(1)  014750  005037  015030                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
(1)  014754  005237  015030           1$:  INC     $SAVR6          ;;WAIT FOR THE INC
(1)  014760  001375                        BNE     1$              ;;OF  WORD
(3)  014762  012677  164152                MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
(3)  014766  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
(3)  014770  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
```

```
(3)  014772  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
(3)  014774  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  014776  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  015000  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  015002  012737  014664  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
(1)  015010  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
(1)  015016  104400                        TYPE                    ;REPORT THE POWER FAILURE
(1)  015020  015032                $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
(1)  015022  000002                        RTI
(1)  015024  000000                $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
(1)  015026  000776                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
(1)  015030  000000                $SAVR6: 0                       ;;PUT THE SP HERE
(1)  015032  005015  047520  042527 $POWER: .ASCIZ  <15><12>"POWER"
(1)  015040  000122
(1)                                        .EVEN
1072                               .SBTTL  TRAP DECODER
(1)
(2)                                ;;**************************************************************
(1)                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
(1)                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
(1)                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
(1)                                ;*GO TO THAT ROUTINE.
(1)
(1)  015042  010046                $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
(1)  015044  016600  000002                MOV     2(SP),R0        ;;GET TRAP ADDRESS
(1)  015050  005740                        TST     -(R0)           ;;BACKUP BY 2
(1)  015052  111000                        MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
(1)  015054  006300                        ASL     R0              ;;POSITION FOR INDEXING
(1)  015056  016000  015064                MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
(1)  015062  000200                        RTS     R0              ;;GO TO ROUTINE
(1)
(3)                                .SBTTL  TRAP TABLE
(3)
(3)                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
(3)                                ;*BY THE "TRAP" INSTRUCTION.
(3)
(3)                                ;       ROUTINE
(3)                                ;       -------
(3)  015064                        $TRPAD:
(3)  015064  012744                        $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
(3)  015066  013252                        $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
(3)  015070  013226                        $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
(3)  015072  013266                        $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
(3)  015074  013454                        $TYPDS  ;;CALL=TYPDS    TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)
(1)
(3)  015076  012240                        $GTSWR  ;;CALL=GTSWR    TRAP+5(104405)  GET SOFT-SWR SETTING
(1)
(3)  015100  012170                        $CKSWR  ;;CALL=CKSWR    TRAP+6(104406)  TEST FOR CHANGE IN SOFT-SWR
(3)  015102  012452                        $RDCHR  ;;CALL=RDCHR    TRAP+7(104407)  TTY TYPEIN CHARACTER ROUTINE
(3)  015104  012572                        $RDLIN  ;;CALL=RDLIN    TRAP+10(104410) TTY TYPEIN STRING ROUTINE
1073
1074  015106  000000               BUFFER: 0
1075           000001                       .END
```

```
ABASE = 177514            13#     35     189
ADDW1 = 000000            35
ADDW2 = 000000            35
ADPCSP= 000000            35
ADDW0 = 000000            35
ADDW1 = 000000            35
ADDW10= 000000            35
ADDW11= 000000            35
ADDW12= 000000            35
ADDW13= 000000            35
ADDW14= 000000            35
ADDW15= 000000            35
ADDW2 = 000000            35
ADDW3 = 000000            35
ADDW4 = 000000            35
ADDW5 = 000000            35
ADDW6 = 000000            35
ADDW7 = 000000            35
ADDW8 = 000000            35
ADDW9 = 000000            35
ADELAY  001572           192#    974
ADEVCT= 000005            35
ADEVM = 000000            35
AENV  = 000000            35
AENVM = 000000            35
AFATAL= 000000            35
AMADR1= 000000            35
AMADR2= 000000            35
AMADR3= 000000            35
AMADR4= 000000            35
AMAMS1= 000000            35
AMAMS2= 000000            35
AMAMS3= 000000            35
AMAMS4= 000000            35
AMSG    006360            417     443     489     509     531     575     601     625     640     664     720     997#
AMSGA   006362            968#    991     996
AMSGADR= 000000           968
AMSGB   006400            989     993#
AMSGC   006412            994     997#
AMSGLG= 000000            35
AMSGTY= 000000            35
AMTYP1= 000000            35
AMTYP2= 000000            35
AMTYP3= 000000            35
AMTYP4= 000000            35
APASS = 000000            35
APRIOR= 000200             15#     35     191
APTCSU= 000040           1064    1067#
APTENV= 000001           1064    1067#   1068
APTSIZ= 000200            225    1067#
APTSPC= 000100           1064    1067#
AS_REG= 000200             35
ATTESTN= 000000            35
AUNIT = 000000             35
```

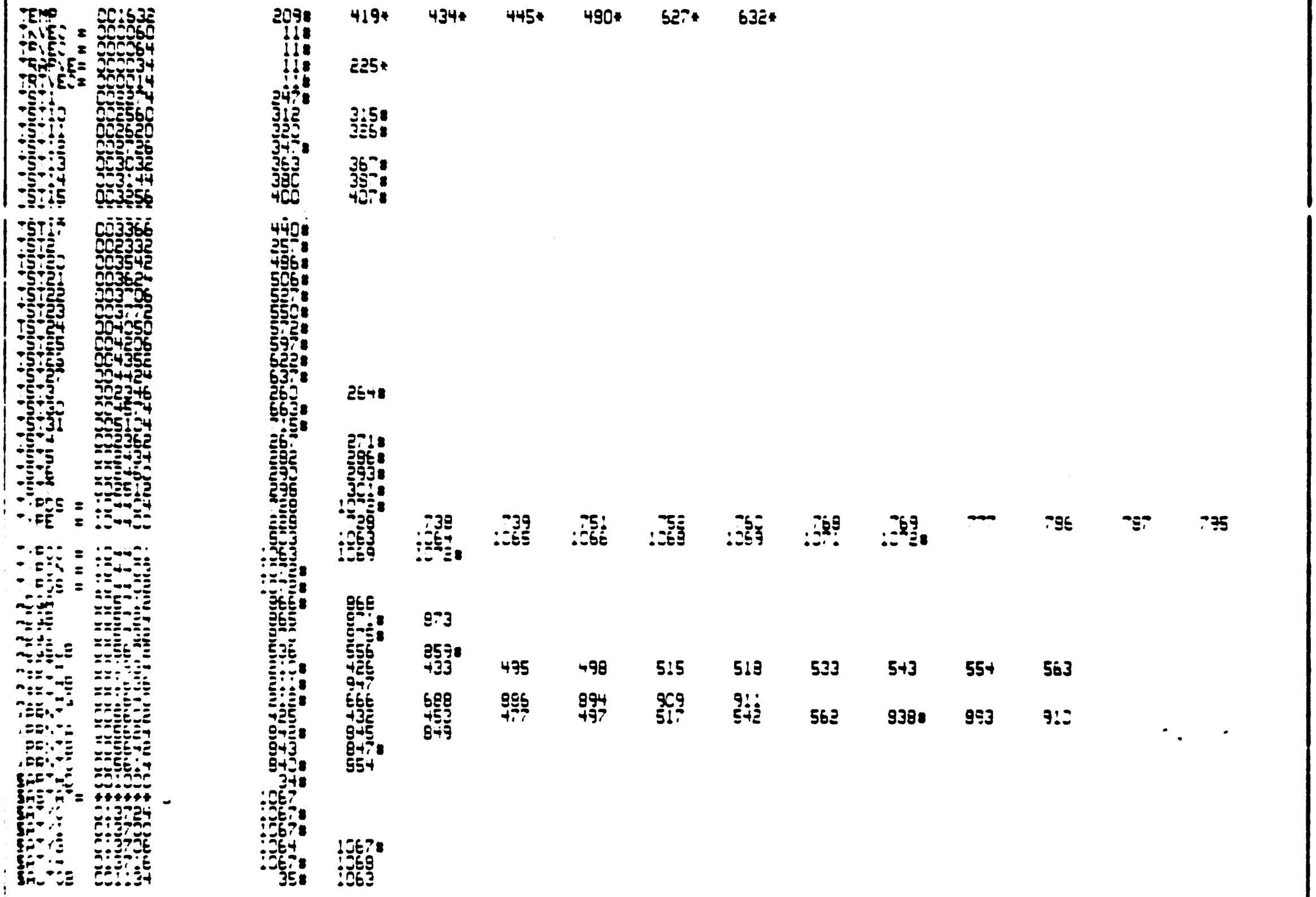
```
AUSWR = 000000            35
AVECT1= 100200            14*     35     190
AVECT2= 000000            35
BGMTA   006024           578     580     582     584     586     588     590     592     604     606     608     610     612
                         614     616     618     620     629     631     643     645     647     649     651     653     655
                         657     659     724     905*
BGMTB   006040           907*    913
BGSAVE  006076           881*    896*    905*    912*    917*
BIT0  = 000001            11*    287     289     294     297     883     898     906     914     922     940
BIT00 = 000001            11*
BIT01 = 000002            11*
BIT02 = 000004            11*
BIT03 = 000010            11*
BIT04 = 000020            11*
BIT05 = 000040            11*
BIT06 = 000100            11*
BIT07 = 000200            11*
BIT08 = 000400            11*    1070
BIT09 = 001000            11*    1068    1070
BIT1  = 000002            11*     936
BIT10 = 002000            11*    1068
BIT11 = 004000            11*
BIT12 = 010000            11*     853     938
BIT13 = 020000            11*    1068
BIT14 = 040000            11*    1070
BIT15 = 100000            11*
BIT2  = 000004            11*
BIT3  = 000010            11*
BIT4  = 000020            11*
BIT5  = 000040            11*
BIT6  = 000100            11*     273     275     281     354     359     374     379     381     394     399     401
BIT7  = 000200            11*
BIT8  = 000400            11*
BIT9  = 001000            11*
BPTVEC= 000014            11*
BRLEV1  001624           206*    241*    242*    355
BRLEV2  001626           207*    243*    375
BRLEV3  001630           208*    244*    245*    395
BUFFER  015106           460     671     693     811     823     838     863     882     921     944    1074*
CDBW    003710           530*
CKSWR = 104406           852     937    1068    1070    1072*
CR    = 000015            11*    1064
CRCNT   006300           455*    955*    961*    965*
CRLF  = 000200            11*    1064
CRLFA   006240           456     956*    962
CRLFB   006302           956     958     960     967*    970
CRLFC   006320           968     972*
CRLFX   006232           411     437     483     504     524     569     594     661     713     955*    997
CDISP = 177570            11*      35     225
DELAY1  006322           318     725     974*
DELNUM  001574           193*    356     376     396
DF1     012166            43      49      55      61      67      73      79      85      91      97     103     109     115
                         121     127     133     139     145     151     157     163     169     175     181     187    1061*
DH1     012121            41      47      53      59      65      71      77      83      89      95     101     107     113
```

```
                        119     125     131     137     143     149     155     161     167     173     179     185     1058#
DISPLA  001142           35#    225*   1068*   1070*
DISPRE  000174           20#    225
DLA1    006336          976#    977     979
DLY1    006354          356*    357*    376*    377*    396*    397*    974*    978*    982#
DLY2    006356          975*    976*    983#
DSAVE   001612          201#    316*    321*
DSWR  = 177570           11#     35     225
DTPA    003320          421#    435
DTPSR   006204          421     428     944#
DTPSRA  006220          948#    951
DTPT    003302          416#
DT1     012156           42      43      54      60      66      72      78      84      90      96     102     108     114
                        120     126     132     138     144     150     156     162     168     174     180     186     1060#
DWPA    003734          536#    548
DWPAA   003742          533*    539#    545*
DWPBB   003746          534*    540#    547*
DWPCC   004015          553*    558#    565*
DWPDD   004022          554*    560#    566*
DWRW    003774          546     553#
DWRWA   004010          556#    567
EMTVEC= 000030           11#    225*
EM1     010201           40   1030#
EM10    010711           82   1038#
EM11    011002           88   1039#
EM12    011051           94   1040#
EM13    011116          100   1041#
EM14    011200          106   1043#
EM15    011240          112   1044#
EM16    011267          118   1045#
EM17    011320          124   1046#
EM2     010233           46   1031#
EM20    011360          130   1047#
EM21    011446          136   1048#
EM22    011460          142   1049#
EM23    011526          148   1050#
EM24    011563          154   1051#
EM25    011622          160   1053#
EM26    011662          166   1054#
EM27    011745          172   1055#
EM3     010277           52   1032#
EM30    012020          178   1056#
EM31    012057          184   1057#
EM4     010360           58   1033#
EM5     010440           64   1034#
EM6     010534           70   1036#
EM7     010631           76   1037#
ERRVEC= 000004           11#    225*    248     249*    255*   1070*
FILBFA  005530          813#    815
FILBUF  005522          494     811#    908
FIRST   001622          205#
GMDWA   004612          667#    686
GMDWB   004632          670#    684
GMDWC   004644          672#    676
```

```
GMDWD   004664          673     677*
GMDWH   004752          691*    712
GMDWJ   004766          693*    709
GMDWK   005002          696*    698
GMDWL   005014          695     699*
GMDWM   005030          702*    704
GMDWN   005042          701     705*
GMDW1   001642          213*    666*    667*    670     672     685     688*    689*    700     711*
GMDW2   001644          214*    668*    677     680*    691*    699     707*
GMDW3   001646          215*    669*    680     682*    683     692*    707     708*
GMDW4   001650          216*    670*    675*    690*    694     710*
GMDW5   001652          217*    694*    697*
GMDW6   001654          218*    700*    703*
GMTA    005740          981*
GMTAA   005770          988*    891
GMTAB   006000          893*    897
GMTO    004052          574*
GMT1    004210          600*
GMT2    004354          624*
GMT21   004372          628*    633
GMT4    004426          639*
GNS   = ****** U          20    1072
GRDWS   006100          679     706     921*
GRDWSA  006120          924*    934
GRDWSB  006130          927*    930
GRDWSC  006146          928     933*
GRDWSD  006154          926     936*
GRDWSE  006114          923*    939
GTSWR = 104405          1072*
HT    = 000011            11*  1064
INIT    002136          227     230*    728
INITO   001670          223     225*
IOTVEC= 000020            11*   225*
JBT     005106          719*
JBTA    005122          723*    727
K0001   001576          194*    451     454
K0002   001600          195*    478
LASTCH  001620          204*    521     827
LF    = 000012            11*  1064
LIC     005544          450     514     823*
LICA    005552          825*    831
LICB    005570          828     830*
LPCHA   003406          449*    481
LVAD    001564          189*    230*
LVB     001604          198*    232*    233*    251*    305*    306*    317*    349*    369*    389*    736*    737*    847*
                        933*    957*    959*    995*
LVBRL   001570          191*    239*    240*    241     243     244
LVIV    001566          190*    234*    235*    236     237
LVS     001602          197*    231*    250*    259     266     273*    275     279*    281     287*    289     294*    295*
                        297     302     307     311     319     331     334     337*    345*    350     354*    359*    370
                        374*    379*    381*    390     394*    399*    401*    735*    742     746     755     763     772
                        760     790     798     842     844     883*    898*    906*    914*    922*    927     929     936*
                        940*    967     969     988     990
LVVCT   001606          199*    236*    329*    352*    372*    392*    408*    1060
```

```
LVVCT1  001610  200*  237*  238*  330*  353*  373*  393*  409  409*
MANTST  005262   22    28   730*  806
MSG0    006420  228  1002*
MSG1    007237  738  1016*
MSG2    006522  739   760   777   795  1004*
MSG3    006651  752   769   787  1007*
MSG4    006724  751  1008*
MSG5    006766  768  1009*
MSG6    007034  786  1011*
MSG7    007145  803  1014*
M910    007741  626  1025*
M911    010030  641  1026*
M912    010106  665  1028*
M913    010145  721  1029*
M93     007302  418  1017*
M94     007343  444  1018*
M95     007422  490  1020*
M96     007476  510  1021*
M97     007541  532  1022*
M98     007603  576  1023*
M99     007662  602  1024*
NPCLB   003450  462*  473
NPCLC   003452  463*  465   467   469   471
PC    =%000007   11*  318*  411*  437*  456*  483*  504*  524*  569*  579*  580*  582*  584*
                 586*  588*  590*  592*  594*  604*  606*  608*  610*  612*  614*  616*  618*
                 620*  629*  631*  643*  645*  647*  649*  651*  653*  655*  657*  659*  661*
                 679*  706*  713*  724*  725*  728*  915*  941*  956*  958*  960*  963*  972*
                 980*  997*  1063*  1064*  1067*  1068*  1069*
PIRQ  = 177772   11*
PIRQVE= 000240   11*
PNPC    003370  442*
PR0   = 000000   11*
PR1   = 000040   11*
PR2   = 000100   11*
PR3   = 000140   11*
PR4   = 000200   11*
PR5   = 000240   11*
PR6   = 000300   11*
PR7   = 000340   11*
PS    = 177776   11*
PSAVE   001614  202*  339*  340*
PSW   = 177776   11*  272*  328*  338*  348*  355*  368*  375*  388*  395*
PWRVEC= 000024   11*  225*  1071*
RCAC    003644  513*  522
RCPL    003626  508*
RDCHR = 104407  1063  1072*
RDLIN = 104410  1072*
RESVEC= 000010   11*
RET2    003030  352  364*
RET3    003132  372  381*
RET4    003244  392  401*
RSTRT   001666   21   27   224*
RO    =%000000   11*  222*  224*  226  460*  462*  474*  475*  671*  693*  728*  911*  923*
                 940*  859*  866  882*  923*  944*  987*  993  1064*  1066*  1067*  1069*  1071*
```

```
                    1072*
R1      =%000001     11*   449*  461*  462   463*  464   466   468   470   472   493*  513*  813
                    825   826*  827   829*  941*  848*  860*  864   867*  884*  888   907*  924*
                    925   933   945*  948   993*  995   1066* 1067* 1071*
R2      =%000002     11*   812*  814*  824*  830*  861*  871   885*  889   946*  949   1066* 1071*
R3      =%000003     11*   862*  869   872*  886*  890*  947*  950*  1063* 1065* 1066* 1071*
R4      =%000004     11*   722*  726*  863*  1065* 1071*
R5      =%000005     11*   417*  421*  425*  428*  432*  443*  450*  453*  477*  489*  494*  497*
                    509*  514*  517*  531*  536*  542*  556*  562*  575*  601*  625*  640*  664*
                    720*  818*  834*  855*  877*  884   885   893*  899*  908*  910*  952*  987
                    998*  1065* 1066* 1071*
R6      =%000006     11*   225*
R7      =%000007     11*
SAVE1    001656     219*   838*  840   921*  923
SAVE2    001660     220*   839*  841
SCAC     003562     493*   502
SCPL     003544     488*
SP      =%000006     11*   225*  248*  253   255   344   364   382   402   728*  732*  1063* 1064*
                    1065*  1066* 1067* 1068* 1069* 1070* 1071* 1072*
STACK = 001100       11*   225   732
START    001662      20     26   222*
STCHAR   001616     203*   491*  493   500*  501   511*  513   520*  521   577*  579*  581*  583*
                    585*   587*  589*  591*  603*  605*  607*  609*  611*  613*  615*  617*  619*
                    628*   630*  642*  644*  646*  648*  650*  652*  654*  656*  658*  723*  907
STKLMT= 177774       11*
SWR      001140      35*   225*  853   938   1063*  1068  1070  1071*
SWREG    000176      20*   225   1063
SW0     = 000001     11*
SW00    = 000001     11*
SW01    = 000002     11*
SW02    = 000004     11*
SW03    = 000010     11*
SW04    = 000020     11*
SW05    = 000040     11*
SW06    = 000100     11*
SW07    = 000200     11*
SW08    = 000400     11*
SW09    = 001000     11*
SW1     = 000002     11*
SW10    = 002000     11*
SW11    = 004000     11*
SW12    = 010000     11*
SW13    = 020000     11*
SW14    = 040000     11*
SW15    = 100000     11*
SW2     = 000004     11*
SW3     = 000010     11*
SW4     = 000020     11*
SW5     = 000040     11*
SW6     = 000100     11*
SW7     = 000200     11*
SW8     = 000400     11*
SW9     = 001000     11*
TBITVE= 000014       11*
```



```
$BASE    001246         35#    230     231     232     735     736     1060
$BDADR   001122         35#
$BDDAT   001126         35#
$BELL    001162         35#    1069
$CDW1    001252         35#
$CHARC   013222       1064#*
$CKSWR   012170       1063#    1072
$CMTAG   001100         35#     225
$CM3   = 000000         35#
$CNTLG   012715       1063#
$CNTLU   012710       1063#
$CPUOP   001220         35#
$CRLF    001167         35#    1063    1064    1068    1069
$DBLK    013670       1066#
$DEVCT   001202         35#
$DEVM    001250         35#
$DOAGN   005236        728#
$DTBL    013660       1066#
$ENDAD   005226         30      729#
$ENDCT   005174        728#
$ENDMG   005245        728#
$ENULL   005242        729#
$ENV     001212         35#    1064    1067    1069
$ENVM    001213         35#     225    1064    1067
$EOP     005144        729#
$EOPCT   005166        728#
$ERFLG   001103         35#    1068*   1070*
$ERMAX   001115         35#     225*   1070*
$ERROR   014146        225     1068#
$ERRPC   001116         35#    1060    1068*   1069
$ERRTB   001254         35#    1069
$ERRTY   014336       1068     1069#
$ERTTL   001112         35#    1069*
$ESCAP   001160         35#     225*   1068    1070*
$ETABL   001212         35#
$ETEND   001254         34       35#
$FATAL   001174         35#    1067*
$FFLG    014144       1067#*
$FILLC   001156         35#    1064
$FILLS   001155         35#    1064
$GDADR   001120         35#
$GDDAT   001124         35#
$GET42   005216        728#
$GTSWR   012240       1063#    1072
$HD    = 000003          9
$HIBTS   001000         34#
$ICNT    001104         35#
$ILLUP   015024       1071#
$INTAG   001135         35#    1063
$ITEMB   001114         35#    1068*   1069
$LF      001170         35#    1063    1064    1068
$LFLG    014143       1067#*
$LPADR   001106         35#     225*   1070*
$LPERR   001110         35#     225*   1068    1070*
```

```
$MADR1   001224          35#
$MADR2   001230          35#
$MADR3   001234          35#
$MADR4   001240          35#
$MAIL    001172          34      35#     225     1064    1068    1070
$MAMS1   001222          35#
$MAMS2   001226          35#
$MAMS3   001232          35#
$MAMS4   001236          35#
$MBADR   001002          34#
$MFLG    014142        1067#*
$MNEW    012733        1063#
$MSGAD   001206          35#   1067*
$MSGLG   001210          35#   1067*
$MSGTY   001172          35#   1067*
$MSWR    012722        1063#
$MTYP1   001223          35#
$MTYP2   001227          35#
$MTYP3   001233          35#
$MTYP4   001237          35#
$NULL    001154          35#   1064
$NWTST=  000001         247#    257#    264#    271#    296#    293#    301#    315#    326#    347#    367#    387#    407#
                        414#    440#    486#    506#    527#    550#    572#    597#    622#    637#    563#    715#
$OCNT    013450        1065#*
$OMODE   013452        1065#*
$OVER    014650        1070#
$PASS    001200          35#    225*    728*
$PASTM   001006          34#
$POWER   015032        1071#
$PWRDN   014664         225    1071#
$PWRMG   015020        1071#
$PWRUP   014736        1071#
$QUES    001166          35#   1063    1064    1068
$RDCHR   012452        1063#   1072
$RDDEC=  ****** U      1072
$RDLIN   012572        1063#   1072
$RDOCT=  ****** U      1072
$RDSZ =  000010        1063#
$RTNAD   005240         728#
$R2A  =  ****** U      1072
$SAVRE=  ****** U      1072
$SAVR6   015030        1071#*
$SCOPE   014472         225    1070#
$SETUP=  000117          37#    225     728     1063    1068    1070
$STUP =  177777          37#
$SVLAD   014614        1070#
$SVPC =  000614          30#
$SWR  =  163400           9#     16#     19      35     225     247     257     264     271     296     293     301     315
                        326     347     367     387     407     414     440     496     506     527     550     572     597
                        622     637     663     715     728    1068    1070    1071
$SWREG   001214          35#    225
$SWRMK=  000000          19    1070
$TESTN   001176          35#   1070*
$TKB     001146          35#   1063
```

```
$TKS    001144          35#   1063*
$TN   = 000032           9#     17#    247#    257#    260     264#    267     271#    282     286#    290     293#    298
                       301#    312     315#    320     326#    347#    363     367#    380     387#    400     407#    414#
                       440#    496#    506#    527#    550#    572#    597#    622#    637#    663#    715#
$TPB    001152          35#   1064*
$TPFLG  001156          35#   1064
$TPS    001150          35#   1064
$TRAP   015042         225    1072#
$TRP  = 000011        1072#
$TRPAD  015064        1072#
$TSTM   001004          34#
$TSTNM  001102          35#    729*    1068    1070*
$TTYIN  012700        1063#
$TYPBN= ******  U     1072
$TYPDS  013454        1066#   1072
$TYPE   012744        1064#   1067    1072
$TYPEC  013156        1063    1064#
$TYPEX  013224        1064#
$TYPOC  013252        1065#   1072
$TYPON  013266        1065#   1072
$TYPOS  013226        1065#   1072
$UNIT   001204          35#
$UNITM  001010          34#
$USWR   001216          35#
$VECT1  001242          35#    234     239
$VECT2  001244          36#
$XTSTR  014504        1070#
$$GET4= 000000         728#
$CFILL  013451        1065#*
$40CAT= ******  U     1066    1070
.       015110          20#     24#     30#     32#     34#     35#     225     729     1059#   1063#   1064    1066#   1067#
                      1068    1069#   1070    1071
.$ASTA= ******  U     1067
.$X   = 001000          34#
```

```
COMMEN     11#
ENDCOM     11#
ERROR      11#    254    261    268    277    283    291    299    304    309    313    323    333    335    342
          360     383    403    744    748    757    765    774    782    792    800    846    931    971    992
ESCAPE     11#
GETPRI     11#
GETSWR     11#
MULT       11#
NEWTST     11#    247    257    264    271    286    293    301    315    326    347    367    387    407    414
          440     496    506    527    550    572    597    622    637    663    715
POP        11#   1066   1067   1071
PUSH       11#   1066   1067   1071
REPORT     11#
SCOPE      11#    247    257    264    271    286    293    301    315    326    347    367    387    407    414
          440     486    506    527    550    572    597    622    637    663    715    728
SETPRI     11#
SETTRA   1072#
SETUP      11#    225
SKIP       11#    252    260    267    276    282    290    298    303    308    312    320    332    335    343
          363     380    400    743    747    756    764    773    791    791    799
SLASH      11#
SPACE      11#
STARS      11#     30     34     35    247    257    264    271    286    293    301    315    326    347    367
          387     407    414    440    496    506    527    550    572    597    622    637    663    715    728
         1063    1064   1065   1066   1067   1068   1069   1070   1071   1072
SWRSU      11#    225#
TRMTRP   1072#
TYPBIN     11#
TYPDEC     11#    728
TYPNAM     11#
TYPNUM     11#
TYPOCS     11#
TYPOCT     11#   1063   1069
TYPTXT     11#
$$CMRE     35#
$$CMTM     35#
$$ESCA     11#
$$NEWT     11#    247    257    264    271    286    293    301    315    326    347    367    387    407    414
          440     486    506    527    550    572    597    622    637    663    715
$$SET    1072#
$$SETM    225#
$$SKIP     11#    260    267    282    290    298    312    320    363    380    400
.EQUAT      6#     11
.HEADE      5#      9
.SETUP      5#     37
.SWRHI      6#     19
.SWRLO     19#
.SACTI      7#     30
.SAPTB      7#     35#
.SAPTH      7#     34
.SAPTY      7#   1067
.SCATC      5#     20
.SCMTA      5#     35
.SEOP       5#    728
```

```
.$ERRO       6#   1068
.$ERRT       5#   1069
.$POWE       6#   1071
.$READ       5#   1063
.$SCOP       6#   1070
.$TRAP       6#   1072
.$TYPD       5#   1066
.$TYPE       5#   1064
.$TYPO       5#   1065
```

ADD 233 238 245 500 520 737 1063 1064 1065 1066 1067 1069
ASL 1063 1069 1072
ASLB 1066
ASR 708 1067
BCC 1066
BEQ 225 282 298 465 467 469 471 546 673 695 701 728 865 870 925
994 1063 1064 1065 1067 1068 1069 1070
BGT 728 1063 1065 1066
BIC 235 359 379 381 399 401 680 707 728 898 914 940 1063 1065
BIS 272 337 354 374 394 883 906 922 935 1063 1065 1066
BISB 1069
BIT 275 281 289 297 953 939 1068 1070
BITB 225 1064 1067
BLOS 1063
BLT 1063 1064 1065 1066
BMI 267 303 320 335 747 756 773 791 843 928 968 989 1066
BNE 225 227 276 290 322 341 358 378 398 435 473 481 502 522 567
633 676 684 686 698 704 709 727 815 928 831 849 854 868 873
891 897 913 939 951 962 977 979 1063 1064 1065 1066 1067 1068 1069
1070 1071
BPL 260 305 312 332 351 371 391 712 743 764 781 799 845 930 970
991 1062 1064 1065 1066 1068
BR 223 225 252 343 363 380 400 548 934 996 1063 1064 1065 1066 1067
1068 1069 1070 1071
CLR 222 225 250 251 279 345 409 728 1063 1065 1066 1069 1070 1071
CLRB 240 1063 1064 1066 1067 1070
CMP 225 253 344 364 382 402 464 466 468 470 472 501 521 683 827
825 1063 1066 1070
CMPB 1063 1064 1067 1068 1070
DEC 321 340 357 377 397 434 480 545 566 632 667 675 699 697 703
711 726 728 814 830 948 957 972 990 996 912 950 961 978 1069
DECB 1064 1065
EMT 11
HALT 20 740 753 761 770 778 788 796 805 1064 1069 1071
INC 224 463 547 565 710 728 826 976 1063 1065 1066 1067 1068 1071
INCB 1064 1068 1070
IOT 11
JMP 20 21 22 26 27 28 728 806
JSR 318 411 417 421 425 428 432 437 443 450 453 456 477 483 499
494 497 504 509 514 517 524 531 536 542 556 562 569 575 578
580 582 584 586 588 590 592 594 601 604 606 608 614 614
616 618 620 625 629 631 640 643 645 647 649 651 655 657
653 661 664 679 706 713 721 724 725 728 993 908 956 959
960 997 1063 1064 1067 1068
MOV 225 230 231 232 234 236 237 241 243 244 248 249 255 273 297
294 295 305 306 316 317 328 329 330 338 339 349 349 353 353
355 356 368 369 372 373 375 376 388 389 392 393 395 398 398
419 445 449 455 460 461 491 493 511 513 533 534 553 554 557
579 581 583 585 587 589 591 603 605 607 609 611 613 615 617
619 627 628 630 642 644 646 648 650 652 654 656 659 666 669
669 670 671 688 690 691 692 693 694 700 722 723 728 733 735
736 811 812 823 824 829 838 839 840 841 859 860 861 863 863
881 882 884 885 886 905 907 921 923 944 945 946 947 955 955
959 974 975 987 1063 1064 1065 1066 1067 1068 1069 1070 1071 1073

```
MOVB    225  239  462  474  475  674  677  678  696  699  702  705  813  825  847
        865  871  888  889  924  933  948  949  993  995 1063 1064 1065 1066 1067
       1068 1070 1072
NEG    1065 1066
NOP     274  280  288  296  416  442  488  508  530  574  600  624  639  719  728
        816  817  832  833  850  851  875  876
RESET   258  265  410  728  730  731  804
ROL     682 1065
RTI     225 1063 1064 1065 1066 1068 1070 1071
RTS     918  934  855  877  899  915  941  952  963  972  980  998 1064 1067 1039
       1072
SEC     681
SEC     848 1066 1067 1068
TRAP   1072
TST     226  259  311  331  672  685  742  755  763  772  780  790  798  844  864
        869  929  969  990 1063 1064 1065 1066 1067 1068 1069 1070 1072
TSTB    266  302  307  319  334  350  370  390  746  842  927  967  988 1063 1064
       1066 1067 1070
.ASCII    35 1002 1004 1005 1006 1011 1014
.ASCIZ    35  728 1003 1007 1008 1009 1012 1015 1016 1017 1018 1020 1021 1022 1023
       1024 1025 1026 1028 1029 1030 1031 1032 1033 1034 1036 1037 1038 1039 1040
       1041 1043 1044 1045 1046 1047 1048 1049 1050 1051 1053 1054 1055 1056 1057
       1058 1063 1069 1071
.BLKB  1063
.BLKW  1066
.BYTE    35  728 1063 1065 1067 1068
.DSABL 1063
.ENABL    4 1063
.END   1075
.ENDC     9   11   19   20   30   34   35   37  225  247  252  257  260  264  267
        271  276  282  286  290  293  298  301  303  308  312  315  320  326  332
        335  343  347  363  367  380  387  400  407  414  440  486  506  527  550
        572  597  622  637  663  715  728  743  747  756  764  773  781  791  799
       1063 1064 1065 1066 1067 1068 1069 1070 1071 1072
.EQUIV   11
.EVEN    35 1059 1067 1069 1071
.IF       9   11   19   20   30   34   35   37  225  247  252  257  260  264  267
        271  276  282  286  290  293  298  301  303  308  312  315  320  326  332
        335  343  347  363  367  380  387  400  407  414  440  486  506  527  550
        572  597  622  637  663  715  728  743  747  756  764  773  781  791  799
       1063 1064 1065 1066 1067 1068 1069 1070 1071 1072
.IFF      11   19   30   34   35  225  247  252  257  260  264  267  271  276  292
        286  290  293  298  301  303  308  312  315  320  326  332  335  343  347
        363  367  380  387  400  407  414  440  486  506  527  550  572  597  622
        637  663  715  728  743  747  756  764  773  781  791  799 1063 1064 1065
       1066 1067 1068 1069 1070 1071 1072
.IFT   1063 1068 1070
.IFTF  1063 1068 1070
.IIF      9   19   20   35  225  728 1063 1064 1068 1069 1070 1072
.IRP     37  247  257  264  271  286  293  301  315  326  347  367  387  407  414
        440  486  506  527  550  572  597  622  637  663  715 1066 1067 1068 1071
.LIST     2    8   11   19   20   35   37  225  247  257  264  271  286  293  301
        315  326  347  367  387  407  414  440  486  506  527  550  572  597  622
        637  663  715  728 1063 1068 1070 1072
```

JO6


```
.MACRO      19    35   225  1072
.MCALL       5     6     7    11    35   225
.MEXIT      35
.NLIST       1     3    11    19    20    35    37   225   247   257   264   271   286   293   301
           315   326   347   367   387   407   414   440   486   506   527   550   572   597   622
           637   663   715   726  1063  1068  1070  1072
.PAGE       35
.REPT       20
.SBTTL      11    19    20    30    34    35   225   247   257   264   271   286   293   301   315
           326   347   367   387   407   414   440   486   506   527   550   572   597   622   637
           663   715   728   734   936  1000  1063  1064  1065  1066  1067  1068  1069  1070  1071
          1072
.TITLE       9
.WORD       20    30    34    35   728  1064  1065  1067  1069  1071
```

ERRORS DETECTED: 0

```
*DZLVAB/I,DZLVAB/CRF<DZLVAB.P11
RUN-TIME:  19  9  1 SECONDS
CORE USED:  25K
```

KO6

Spooler runtime 10 Seconds, 298 KCS, 279 disk reads, 3 disk writes, 74 pages

[redacted] Date 14-Oct-76 15:40:42 Mueller IPC-0 5078 [1,3] ***EOF***