IDENTIFICATION

Product Code:	AC-8898E-MC
Product Name:	CZLACEO LA36 TERM (DL11 & KL11)
Date Created:	August 1978
Maintainer:	DIAGNOSTIC GROUP
Authors:	Robert W. Baker R. Quenneville Ralph A. Schauber John V. Chatalian

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

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

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

HISTORY

1.0 DECO CZLAC-E-0

1.1.0 Closed Problem Report AA3318

1.1.1 It was reported that a DL11-A operated at 110 baud caused failure in the AREAD routine because the 200 msec. delay is not of sufficient duration to allow setting of the Receiver Register Status "DONE" bit through the Maintenance bit facility. The time delay was increased from 200 to 600 msec.

1.2.0 Closed Problem Report AA3643

1.2.1 Tests 56,57,60,61,62,63,64,65,and 66 do not run properly when run on an LSI-11. This problem was resolved by changing the branch after the CHAIN command to go back to test for the LSI-11 switch in order to effect the appropriate action during each test.

1.2.2 Second time-out in Test 64 allows excessive wait for operator response. The time delay was reduced from "177777" to "600".

1.2.3 Common routine TYPE does not save the contents of R0 resulting in the loss of this information and consequent failure.

Instructions were included to save the contents of R0 on entry into the routine and to restore them upon exit.

1.2.4 Loss of stack contents for non-LSI-11 computers due to incorrect sequence of instructions in Test 65 was also reported in Problem Report AA3803. Refer to 1.3.1.

1.3.0 Closed Problem Report AA3803

```
1.3.1  Testing of non-LSI-11 computer
s results in the program  hanging
       up  because  the stack gets popped awa
y in Test 65.  The branch
       after the test for the LSI-11 switch in Test 6
5  should  go  to
       the  CHAIN command for proper exit from the test for n
on-LSI-11
       machines.  This change supersedes the change released  in  DEP
O
       MO-11-DZLAC-D-1.
```

TABLE OF CONTENTS

1.0 ABSTRACT

This diagnostic is divided into three basic sections:

1. A check of the console terminal interface logic.
2. A check of the printing characteristics and control logic.
3. An echo portion designed to check the keyboard and to aid in the diagnosis of terminal problems.

Patterns used by the printing tests were chosen for ease of visual verification. The echo tests were designed for maximum flexibility, with Test 24 allowing any desired pattern to be used.

2.0 REQUIREMENTS

2.1 EQUIPMENT AND ASSIGNMENTS

The diagnostic is written to run on all models of the PDP-11 computer with either a KL11 or DL11 console terminal interface. The diagnostic is preset to test up to 16 additional terminals (on DL11's) assigned between addresses 776500 and 776676. This preset quantity (16) and preset address (776500) can be changed by depositing the quantity in DLNR and the starting address in DLADR. For example, to allow for up to 31 additional terminals, the address 775610 could be placed into DLADR and the octal equivalent of 31, i.e., (37) would be placed into DLNR. The number of additional DL11's actually tested will be adjusted automatically downward based upon the first DL11 address (within the implied range) found to be unresponsive. Thus if there is no DL11 present to match the address in DLADR only the console terminal will be tested. Therefore, all DL11's in excess of the console terminal must have contiguous address assignments with the lowest address corres ponding to the value in DLADR.

The console terminal (assigned standard) can be reassigned by placing the address of its receiver status register into CONADD and its receiver interrupt vector into CONVEC. This reassignment can be made to a terminal within the set of terminals implied by DLNR and DLADR without adverse effect. Note that a terminal with a slower speed (if any) will determine the speed at which all of the terminals are tested. Such a terminal should generally be excluded from the test, or tested separately. (Refer to the symbol definitions in the listing for the above mentioned locations.)

2.2 STORAGE

The diagnostic program uses all of 4K of memory with exception of the area used by the absolute loader.

2.3 PRELIMINARY PROGRAMS

Any applicable PDP-11 diagnostics should be run on the processor. If any errors are encountered during the interface check, refer to the appropriate interface diagnostic for further help in locating the problem if needed.

2.4 ADDITIONAL PROGRAMS

This diagnostic is for verification of basic terminal functions only. If the terminals under test have hardware options installed run diagnostic MAINDEC-11-DZLAF-A , the LA36 TERMINAL OPTIONS TEST.

3.0 LOADING PROCEDURE AND INITIALIZATION

Load the LA36 diagnostic program tape following normal procedures. Before starting the program, refer to the description of the routine "DLY". Time delays used by the program are a function of the CPU model and memory type and should be set-up before running the diagnostic. The routine is preset for a PDP-11/05 with core memory. Refer to Section 2.1 for non-standard terminal addresses and for testing multiple DL11 interfaces.

If a hardware switch register does not exist, the program will use the contents of location 176 as the value of the switches. Therefore, be sure to load location 176 with the switch value before starting the program when not using hardware switches.

If the CPU is an LSI-11, 11/03 be sure to set switch register bit 9 to a 1. Special tests are run on the DLV11 interface.

4.0 STARTING PROCEDURE

4.1 STARTING ADDRESSES

200(8) = Run with Switch Register Control
- perform Console Terminal I/O tests.

204(8) = Run with Switch Register Control
- skip Console Terminal I/O tests.

210(8) = Run with Keyboard Control
- perform Console Terminal I/O tests.

214(8) = Run with Keyboard Control
- skip Console Terminal I/O tests.

4.2 Switch Register Control With I/O Tests

A. Set the switch register to 200(8) and press the load address switch.

B. Set switch register bit 9 to a 1 if the processor is an LSI-11, 11/03. Refer to Section 5.1.5.

C. Set the switch register bits 7-0 equal to the paper width in terms of the number of columns (octal). Refer to Section 5.1.8.

D. Set the switch register bit 8 equal to 1 or 0 and press the start switch. A message will be printed indicating the number of DL11's being tested. Refer to Section 5.1.6.

E. If bit 8 were zero when starting, the Printer tests are executed sequentially, after the entire series of I/O tests are executed.

F. If bit 8 was set when the start switch was pressed, the entire series of I/O tests will be executed and the CPU will halt at location SELHLT. The program will then be waiting for control via the switch register.

4.3 Switch Register Control - Without I/O Tests

Same as Section 4.2 except in step A, set the switch register to 204(8).

4.4 Keyboard Control - With I/O Tests

A. Set the switch register to 210(8) and press the load address switch.

B. Set the switch register bits 7-0 equal to the paper width in terms of the number of columns (octal). Refer to Section 5.1.8.

C. Set switch register bit 9 to a 1 if the processor is an LSI-11, 11/03. Refer to Section 5.1.5.

D. Set switch 8 and press the start switch. A message will be printed indicating the number of DL11's being tested. Refer to Section 5.1.6.

E. If bit 8 was zero when starting, the printer tests are executed sequentially after the entire series of I/O tests are executed.

F. If bit 8 were set when the start switch was pressed, the entire series of I/O tests will be executed followed by the select test message. The program will then be waiting for a test selection via any terminal keyboard. Refer to Section 5.2.

4.5 Keyboard Control - Without I/O Tests

Same as Section 4.4 except in step A, set the switch register to 214 (8).

5.0 OPERATING PROCEDURE

The program can be controlled in either of two methods: by the console switch register or from the keyboard of the terminal(s) under test.

5.1 SWITCH REGISTER CONTROL

The variou
s switches and their functions are listed below. Switches
may be changed
and set as desired except as noted in the specific
switch descriptions. Refe
r to the detailed switch descriptions for
further, more complete information
.

```
SWITCH NUMBER              DESCRIPTION

           15          1(up)    = HAL
T AT END OF TEST
                       0(down) = CONTINUE TEST SEQUENCE

 14             1(up)    = CONTINUE ON ERROR
                       0(down) = HALT ON
 ERROR
           13          1(up)    = DRIVE ONLY CONSOLE TERMINAL

             0(down) = DRIVE ALL TERMINALS

           11          1(up)    = LOOP ON
INDIVIDUAL TEST
                       0(down) = NORMAL TEST SEQUENCE

           9
               1(up)    = CPU TYPE IS AN LSI-11, 11/03
                       0(down) =
 ALL OTHER PDP-11'S
           8           1(up)    = RUN TEST ONCE AND HALT

                    0(down) = LOOP ON TEST SEQUENCE

           5-0
  TEST NUMBER SELECTION

           7-0                        NUMBER OF COLUMNS AT ST
ART-UP
```

5.1.1 Switch 15

With switch 15 in the up position, the program
 will halt at the end of
the current test. Replacing switch 15 to the down pos
ition and press-

ing CONTINUE will continue the normal test operation. During the halt, any of the control switches may be changed or set as desired.

5.1.2 Switch 14

Placing switch 14 in the up position will cause the program to continue on errors during any of the I/O tests only. With switch 14 down, the program will halt (at ERRHLT) on any errors during the I-O tests with the location of the error in R0. Pressing CONTINUE will cause the program to continue if switch 14 is down. With switch 14 up, pressing continue will cause the program to loop on the error.

NOTE

Error halts can occur only during the I/O tests. The terminal is connected to a serial line and there is no error information returned to the program from the terminal. Therefore the program cannot report errors occurring in the terminal. Errors detected during the interface tests will result in halts as described above.

5.1.3 Switch 13

Placing switch 13 in the down position will cause the driving of all multiple terminals during the Printer tests only. If switch 13 is up, only the console terminal is driven.

** Note: Switch 13 should only be changed when the program is waiting for a test selection.

5.1.4 Switch 11

Placing switch 11 up at any time will cause the program to loop on the current test as long as switch 11 remains up. Replacing switch 11 down will cause the program to resume normal operation at the completion of the test.

5.1.5 Switch 9

Placing switch 9 up at the start of the test will cause an automatic change in the DELAY timing, and the execution of special DLV11 I/O tests. The DLV11 has no maintenance mode and will cause th

e program
to hang if tested as a DL11.

5.1.6 Switch 8

With switch 8
in the down position the program will continue to loop
through the present t
est sequence. Placing switch 8 up will cause the
program to halt (at SFLHLT) a
t the completion of the current test.
After the halt, set the control switc
hes as desired and set switches 5
to 0 to the next desired test number, and the
n press CONTINUE to start
the test.

When starting the diagnostic the operator can select a specific test rather than automatically starting the printing test sequence by setting switch 8 up before starting the diagnostic. Upon completion of the I/O test sequence (if being run) the program will either halt at SELHLT waiting for a test selection via the switch register or print the select test message and wait for a test selection from any keyboard. Refer to Section 4 for further information.

5.1.7 Switches 5 to 0

Switches 5 to 0 are used to select specific tests when under switch register control. Test numbers are always in octal.

5.1.8 Switches 7 to 0 (at start-up only)

At start-up only, switches 7 to 0 are used to set the desired maximum number of columns the diagnostic is to test. If the number set is greater than 132(10) or less than 30(10), the program will default to 132(10). The value set must be in octal form. Thus, for normal operation leave switches 7 to 0 down to test the full 132(10) columns.

5.2 KEYBOARD CONTROL

The program will be under keyboard control whenever the diagnostic is started at location 210 or 214. Switches on the console switch register will have no effect when under terminal control except for switch 15. The I/O tests cannot be selected when under keyboard control.

To stop a test at any time, type the "RUBOUT" or "DELETE" key on any keyboard. Any terminal may stop the test and select the next test if switch 13 is down. When a test is stopped by typing a "RUBOUT" or "DELETE", the test will terminate and the following message will be typed:

SELECT TEST NUMBER

At this time, type the desired test number followed by any one of the following control characters:

. (period) = Run the selected test once and return for another test selection.

L = Loop on the selected test until a "RUBOUT" is typed.

S = Start the test sequence with the selected test. Continue to loop on the printing test sequence until a "RUBOUT" is typed.

The "L" or "S" may be either upper or lower case, but the test number must always be a 2 digit octal number. The test number and terminator are echoed by the program, thus each character will be printed twice if the terminal is in half duplex. For all echo tests, the "L" and "S" will only run the test once (the same as if typing a period). For

all option tests, the "S" will o
nly run the test once (the same as if
typing a period), however, typing an "L"
will cause the program to
loop on the selected test. If an error is detect
ed in the test selec-
tion (illegal test number or control character), a quest
ion mark is
printed and the message will be repeated.

6.0 TEST DESCRIPTIONS

6.1 PRINTING TESTS

These tests are designed as a test of the printing mechanism and the associated control logic. At the beginning of each test, the test number will be printed indicating which test is being executed and, if the test is a function of the number of columns, the number of columns being tested will be indicated. A detailed description and sample patterns for each printing test follows:

6.1.1 Test 0 - Data Path Test

This test is used to test the data lines to and through the interface and to the terminal. An alternating bit pattern is sent which will print alternating *'s and U's in a checkerboard pattern to the maximum column width. The starting character for each line is alternated and a total of four lines are printed.

With the Auto Line Feed Option set to produce an automatic line feed after every received carriage return, there will be a blank line between each printed line.

EXAMPLE:

```
           *U*U*U*U*U*U*U*U*U*U*U*U*U*U*U

U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*
           *U*U*U*U*U*U*U*U*U*U*U*U*U*U*U

    U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*
```

6.1.2 Test 1 - Printable Character Test

This test produces a check of all 94(10) printable characters. The characters are printed in groups of three with three groups per line, separated by three spaces between groups. The first column will contain all ASCII codes from 040 to 077. Column two will contain all ASCII codes from 100 to 137 - primarily the capital letter set. The last column will contain all ASCII codes from 140 to 176 - primarily the small letter set.

With the Auto Line Feed Option set to produce an automatic line feed after every received carriage return, there will be a blank line between each printed line.

```
EXAM
PLE:

                @@@   ```
          !!!   AAA   aaa
          """   BB
B   bbb
          ###   CCC   ccc
          $$$   DDD   ddd
          %%%   E
EE   eee
          &&&   FFF   fff
          '''   GGG   ggg
          (((
HHH   hhh
          )))   III   iii
          ***   JJJ   jjj
          +++
 KKK   kkk
          ,,,   LLL   lll
          ---   MMM   mmm
          ...
  NNN   nnn
          ///   OOO   ooo
          000   PPP   ppp
          111
   QQQ   qqq
          222   RRR   rrr
          333   SSS   sss
          44
4   TTT   ttt
          555   UUU   uuu
          666   VVV   vvv
          7
77   WWW    www
          888   XXX   xxx
          999   YYY   yyy

:::   ZZZ     zzz
          ;;;   [[[
          <<<   \\\
          ===   ]]
]

          >>>
          ???
```

6.1.3 Test 2 - Non-printable Character Test

This test checks all non-printable characters that have no control function in the LA36 terminal or the LA36 options (such as CR, LF, BS,

& BEL).
First the ASCII code will be printed followed by the mnemonic
after a few se
parating spaces. Following the mnemonic, the actual
control character will be
sent three times and nothing should happen
at the printer. This pattern
is repeated, three times on a line,
until all of the non-printing characters
have been tested.

With the Auto Line Feed Option set to produce an automatic
line feed
after every received carriage return, there will be a blank l
ine
between each printed line.

EXAMPLE:

```
             001    SOH  002  STX
 006  ACK
             020    DLE  021  DC1   022  DC2
             023    DC3  024  DC
4  025  NAK
             026    SYN  027  ETB   030  CAN
             031    EM   032
SUB  034   FS
             035    GS   036  RS    037  US
             177    DEL
```

6.1.4 Test 3 - Carriage Return Test

This test checks the carriage return from all even numbered columns and the spacing of the solenoid head from the left margin. It is also a good check for proper operation of the position decoder.

The test prints a full line of alternating 0's and spaces, starting with a 0. At the end of the line the print head is returned to the left margin with a carriage return. The spaces are then filled in by spacing the print head out from the left margin to the first space, printing an "X", and executing a carriage return. This pattern is repeated until the line is completed. Check to see that all X's are in the middle of the space between the two zeroes on either side of it.

EXAMPLE:

```
          0X0X0X0X0X0X0X0X0X0X0X0X0
X0X0X
```

With the Auto Line Feed Option set to produce an automatic line feed after every received carriage return, this test will print a line of 0's and spaces, then print a diagonal line of X's. To correctly check the encoder, the Auto Line Feed Option should be disabled.

```
EXAMPLE:

          O O O
O O O O O O
           X
             X
               X
                 X

                   X
                     X
                       X

                  X
```

6.1.5 Test 4 - Multiple Line Feed Test

This test checks the line feed capability of the printer by sending various groups of line feeds interspaced with reference lines. The number printed as the reference line indicates the number of line feeds that follow. The first and last lines also contain a string of dashes as reference points for measuring the total distance between the two dashed lines, i.e., 63(10) lines.

With the Auto Line Feed Option set to produce an automatic line feed after every carriage return, the number printed will indicate one less than the number of line feeds (the number of blank lines) that follow. The total distance between the two dashed lines will then be 69 lines.

EXAMPLE:

```
          01-------------
-------------------------------------
          02

          04



     08







          16
            \
             15 Blank Line
s
            /
          32
            \
             31 Blank Lines
            /
          00----------------------------------------------------
```

6.1.6 Test 5 - Single Line Feed Test

This test is designed to check the timing of single line feeds and the capability of doing line feeds in all columns. Two reference lines are used by this test (and Test 6) which also can be used to easily check the number of columns the printer is printing.

The first reference line contains 130(10) zeroes followed by two 2's if testing 132(10) columns. If less than 132 columns, the line will contain 0's for two less than the maximum number of columns followed by the two 2's. This reference line is a quick check for 132(10) columns if testing the full 132(10) columns. The second reference line prints a string of numbers (1 to 9 & 0) repeated to the maximum column. This line, again, can be used as a quick check of the number of columns.

The line feed test is accomplished by: printing the first reference line of 0's and two 2's; then either sending 60(10) 3's, if testing 132(10) columns, or waiting 1.8 seconds for an LCV, if testing less than 132(10) columns. If testing 132(10) columns, nothing should happen, except for an LCV, at the end of the line. The 3's should be lost and never printed. After the LCV, with the print head at the extreme right, a carriage return - line feed will be sent followed by repeated backslashes "\" and linefeeds to print a diagonal line down the paper. When a backslash is printed in the maximum column, a carriage return will be sent immediately after the line feed and the second reference line of sequential numbers will be printed. After completing the line, a carriage return - line feed will be sent and the program will wait one second for the carriage return function to complete. After the delay, the reference line will be repeated, the last line being guaranteed to be correct. Any timing problems during the line feeds will show as misprints or missing characters during the first 16(10) characters of the middle reference line. Also, any paper feed problems will cause misalignment of the slashes forming the diagonal line.

```
EXAMP
LE:
          000000000000000000000000000022
          \
           \

        \
             \
              \
               \
                \

                 \
                  \
                   \

    \
                     \
                      \
                       \

                        \
                         \

   \
                           \
                            \

                \
                              \

   \
                                \
                                 \

                               \
                                   \

                            \
                                     \

                             \
                                       \

    123456789012345678901234567890
          123456789012345678901234567890
```

With the Auto Line Feed Option set to produce an automatic line feed after every received carriage return, there will be a blank line every

place a
carriage return is executed.

EXAMPLE:

```
          0000000022
          \
           \
            \
             \
              \
               \
                \
                 \
                  \
 \
          1234567890
          1234567890
```

6.1.7 Test 6 - Backspace Test

This test is designed to test the print timing as in Test 5 as well as the backward and forward movement of the print solenoid head.

The test consists of the same first reference line as in Test 5 then a carriage return-line feed. A full line is then printed using the following pattern:

```
Forward Slash   "/"
          Backspace
          Back Slash      "\"
```

This pattern produces a line of all X″s. The two slashes should cross exactly at the middle, producing the X character. When the line is completed a carriage return-line feed is sent and the last two reference lines are printed as in Test 5. Anv timing problems will show in the first 16(10) characters of the middle reference line; again as in Test 5.

With the Auto Line Feed Option set to produce an automatic line feed after every received carriage return, there will be a blank line between each printed line.

EXAMPLE:

```
  000000000000000000000000000022
          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

      123456789012345678901234567890
          123456789012345678901234567890
```

6.1.8 Test 7 - Overprint Test

This test is designed to check the spacing
and repeatable printing
characteristics of the printer. Three rows of ch
aracters are each
overprinted two times. The rows consist of the following c
haracters
alternated across the line:

```
          Row 1            M-SP

    Row 2            SP-@
          Row 3            &-SP
```

The resulting patter
n will be a checkerboard pattern and the over-
printed characters should
be aligned properly with the initial char-
acters.

EXAMPLE:

```
          M
 M M M M M M M M M M M M M M
                @ @ @ @ @ @ @ @ @ @ @ @ @ @

& & & & & & & & & & & & & & &
```

With the Auto Line Feed Option set to produce
an automatic line feed
after every received carriage return, the lines will
not be overprint-
ed. There will be three lines of each character with a bl
ank line
between each group of characters. The characters in each group shoul
d
be in the same columns.

EXAMPLE:

```
          M M M M M M M M M M

     M M M M M M M M M M
          M M M M M M M M M M

           @ @ @ @ @
@ @ @ @
           @ @ @ @ @ @ @ @ @
           @ @ @ @ @ @ @ @ @
```

```
&   &   &   &   &   &   &   &   &   &
                     &   &   &   &   &   &   &   &   &   &
                     &   &   &   &   &   &   &   &   &
&
```

6.1.9 Test 10 - Printing Frequency Sweep Test

This test prints the character "H" repeatedly, 30(10) characters per line for four lines. During the first two lines, the time interval between characters is increased from 30(10) milliseconds to 1.8 seconds using the following formula to create a logarithmic increase:

New Delay = Old Delay + Old Delay/16 + Old Delay/128

The last two lines do just the reverse. The time interval between characters is decreased from 1.8 seconds to 30(10) milliseconds using the following formula to again create a logarithmic decrease:

New Delay = Old Delay - Old Delav/16 - Old Delay/128

Look for possible misalignment of the characters or spaces between characters as an indication of timing problems.

With the Auto Line Feed Option set to produce an automatic line feed after every received carriage return, there will be a blank line between each printed line.

EXAMPLE:

```
          HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH

         HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
          HHHHHHHHHHHHHHHHHHHHHHHHHHHHH
H
          HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
```

6.1.10 Test 11 - Ribbon Feed Test

This test checks the ribbon feed mechanism by printing a single column of 24 lines of X's down the left hand margin of the page. Visually check for proper operation of the ribbon feed mechanism during this test.

```
With the
 Auto Line Feed Option set to produce an automatic  line  feed
after  every  re
ceived  carriage  return,  there  will be a blank line
between each printed lin
e.

EXAMPLE:

X
X
X
X
X
X
X
```

6.1.11 Test 12 - Printer Bell Test

T
his test checks the printer bell buffer to insure that eight bells
are dist
inctly heard, even when sent at the maximum transfer rate.
The program sends
 8 bell codes at the maximum rate to the printer
then waits 2.5 seconds to
 allow the operator to hear the bells.

6.1.12 Test 17 - Life Test

Th
is test runs continuously and is run as an individual, special test.
It is not
part of the standard printing test sequence.

This test prints 2 lines of each
 printable character and then repeats
continuously. The second line of e
ach character is overprinted 4
times to conserve paper. At the end of each com
plete pass through the
character set, a message is printed indicating the nu
mber of passes
executed. If any character (except "Rubout") is typed on the ke
yboard
during this test, the pattern will change and restart with the typed
character. This will only happen if keyboard control is in use.

EXAMPLE:

```
              AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
              AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AA
              BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
              BBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBB
```

If the Auto Line Feed Option is set to produce an automatic line fe
ed
after every received carriage return, the test will print six lines of
each
 character with a blank line between the first and second lines as
well as betw
een each group of characters.

EXAMPLE:

```
              AAAAAAAAAAAAAAA
```

```
AAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAA

 AAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAA

      BBBBBBBBBBBBBBB

      BBBBBBBBBBBBBBB
      BBBBBBBBBBBBBBB
      BBBBBBBBBBBBBBB

      BBBBBBBBBBBBBBB
      BBBBBBBBBBBBBBB
```

6.2 ECHO TESTS

These tests are designed as a test of the keyboard and an aid in isolating troubles within the terminal. At the beginning of each test, the test number will be printed indicating which test is being executed. Typing a "RUBOUT" or "DELETE" at any time, whether in keyboard control or not, will exit the current Echo test and print a test termination message. If in keyboard control, the select test message will be printed and the program will await a test selection as usual. In switch register control, the program will halt (at SELLHLT) waiting for control via the switch register. A detailed description of each test follows:

6.2.1 Test 20 - Character Echo Test

This test is designed to operate the terminal in a simulated local mode. Any character typed on the keyboard (except a "rubout") will be echoed to the printer.

If the LA36 terminal is in half duplex with the Auto Line Feed Option available, typing a carriage return may cause a garbled response on the terminal during this test.

6.2.2 Test 21 - Line Echo Test, Fast Rate

This test continually sends full lines of any character up to the maximum column width. The test prints a "0" character when started until a key is typed on the keyboard. The program will then send the typed character until another character is typed or the test is terminated by typing a "rubout". The characters are transmitted at the maximum rate with a carriage return-line feed inserted after

every 132(10)
printable characters.

If the LA36 is in half duplex when runn
ing this test, characters may
be lost or garbled whenever a character is type
d on the keyboard.

With the Auto Line Feed Option set to produce an automatic
line feed
after every carriage return, there will be a blank line between
each
printed line.

6.2.3 Test 22 - Line Echo Test, Slow Rate

This te
st is identical to Test 21 except a delay of 1.8 seconds is
inserted betwe
en each character to allow the print head to perform an
LCV between characters.

6.2.4 Test 23 - Character/Code Echo Test

This test will print the octal code received by the processor followed by the character or the mnemonic of the character every time a key is pressed on the keyboard. The parity of the received code will be indicated as either odd or even. Allow sufficient time between characters for the line to be printed.

With the Auto Line Feed Option set to produce an automatic line feed after every received carriage return, there will be a blank line between each printed line.

EXAMPLE:

```
             301    A        ODD
             263    3        ODD
             215    CR       E
VEN
             240    SP       EVEN
```

6.2.5 Test 24 - Selected Pattern Echo Test

This test is designed to give maintenance the flexibility to choose their own patterns for isolating any specific problems which may arise in the field.

Type any characters (except control-C and rubout) and each character will be echoed as typed. A maximum of 256(10) characters may be inputted. No carriage returns or line feeds are inserted by the program, all characters must be inputted by the operator. To terminate the input string type a control-C, the program will then continually echo the inputted pattern. To stop the printing, type control-C. The program will stop printing the pattern and will wait for either another pattern input terminated by a control-C, or the same pattern may be used again by typing control-C. To exit the test at any time, type a "rubout".

When any options are available, be careful what characters or character sequences are selected.

6.2.6 Test 25 - Bell Echo Test

This test is designed to test the bell on column 64 if typing has occurred on the line. The test prints a message:

```
     TYPE ANY PRINTABLE CHARACTER AND LISTEN FOR BELL ............
```

After the test message is printed, type any printable character on the keyboard. The character will be echoed and the bell should ring. The message will then be typed again. Type the "rubout" key to terminate the test at any time.

6.4 STANDARD I/O TESTS

These tests are designed as a brief check of the console terminal interface logic. E ach check is structured as an independent test and the switch register control s may be used. A description of each test is given in the program listing. Any errors encountered during the I/O tests will cause a halt at location "ER RHLT" if switch 14 is down.

```
CZLACEO LA36 TERM (DL11 & KL11) MACRO M1110  25-AUG-78 10:13
TABLE OF CONTENTS
```

```
        .TITLE  CZLACEO LA36 TERM (DL11 & KL11)
;
;LA36 DIAGNOSTIC (DL11 & KL11 INTERFACE)
;
;
;AUTHORS:       ROBERT W. BAKER
;               R. QUENNEVILLE
;               RALPH A. SCHAUBER
;               JOHN V. CHATALIAN
;
;
;COPYRIGHT 1974,1977,1978  DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
;
;
;
        .SBTTL  SWITCH REGISTER OPTIONS
;
;SWITCH         POSITION        FUNCTION
;
;  15           UP (1)          HALT AT COMPLETION OF CURRENT TEST
;               DOWN (0)        CONTINUE NORMAL TEST SEQUENCE
;
;  14           UP (1)          CONTINUE ON ERROR
;               DOWN (0)        HALT ON ERROR
;
;  13           UP (1)          DRIVE ONLY CONSOLE TERMINAL
;               DOWN (0)        DRIVE ALL TERMINALS
;
;  11           UP (1)          LOOP ON INDIVIDUAL TEST
;               DOWN (0)        NORMAL TEST SEQUENCE
;
;  09           UP (1)          CPU TYPE IS AN LSI-11, PDP-11/03
;               DOWN (0)        ALL OTHER PDP-11 CPU'S
;
;  08           UP (1)          HALT TO SELECT TEST AT END OF CURRENT TEST
;               DOWN (0)        LOOP ON TEST SEQUENCE
;
; 05-00                         TEST # SELECTION
;
; 07-00                         # OF COLUMNS AT START-UP
```

```
        .SBTTL  SPECIAL OPERATIONAL INFORMATION
;
;1.--   THE STANDARD CONSOLE TERMINAL INTERRUPT VECTOR AND REGISTER
;       ADDRESSES ARE USED. TO REDEFINE THE LOCATION OF THE CONSOLE
;       TERMINAL THE SYMBOLIC LOCATIONS "CONADD" AND "CONVEC" SHOULD
;       BE CHANGED BEFORE START UP.
;
;2.--   BEFORE START UP REFER TO THE DESCRIPTION OF THE ROUTINE "DLY".
;       TIMING IS A FUNCTION OF THE PDP-11 MODEL AND MEMORY TYPE AND
;       SHOULD BE SET UP BEFORE RUNNING THE DIAGNOSTIC.
;
;3.--   IF CPU IS A PDP-11/03 , LSI-11 SET SWITCH REGISTER
;       BIT 09 TO A 1. SPECIAL TESTS ARE RUN ON THE DLV11.
;
;4.--   SYSTEMS WITHOUT A HARDWARE SWITCH REGISTER SHOULD USE
;       MEMORY LOCATION 176 AS A SOFTWARE SWITCH REGISTER.
;
;5.--   THIS DIAGNOSTIC IS FOR VERIFICATION OF BASIC TERMINAL
;       FUNCTIONS ONLY. IF THE TERMINAL UNDER TEST HAS HARDWARE
;       OPTIONS INSTALLED RUN DIAGNOSTIC MAINDEC-11-DZLAF-A, THE
;       LA36 TERMINAL OPTIONS TEST.
```

```
    6700                                                .SBTTL  SYSTEM EQUATES
    6800                                        ;
    6900                                        ;
    7000                                        ;       REGISTER EQUATES
    7100                                        ;
    7200        000000                                  R0=%0
    7300        000001                                  R1=%1
    7400        000002                                  R2=%2
    7500        000003                                  R3=%3
    7600        000004                                  R4=%4
    7700        000005                                  R5=%5
    7800        000006                                  SP=%6
    7900        000007                                  PC=%7
    8000        177776                                  PSW=177776
    8100                                        ;
    8200                                        ;       SYSTEM EQUATES
    8300                                        ;
    8400        000001                                  BIT0=1
    8500        000002                                  BIT1=2
    8600        000004                                  BIT2=4
    8700        000010                                  BIT3=10
    8800        000020                                  BIT4=20
    8900        000040                                  BIT5=40
    9000        000100                                  BIT6=100
    9100        000200                                  BIT7=200
    9200        000400                                  BIT8=400
    9300        001000                                  BIT9=1000
    9400        002000                                  BIT10=2000
    9500        004000                                  BIT11=4000
    9600        010000                                  BIT12=10000
    9700        020000                                  BIT13=20000
    9800        040000                                  BIT14=40000
    9900        100000                                  BIT15=100000
   10000        000000                                  OPEN=0
   10100        040000                                  SCOPSW=BIT14                ;SCOPE SWITCH
   10200        004000                                  NITRSW=BIT11                ;TEST LOOP SWITCH
   10300        005726                                  POPSP=5726                  ;POP STACK ONCE
   10400        022626                                  POPSP2=22626                ;POP STACK TWICE
   10500        000340                                  PRTY7=340                   ;PRIORITY LEVEL DEFINITIONS
   10600        000200                                  PRTY4=200
   10700        000200                                  ACRLF=200
   10800        001000                                  LSI11=BIT9                  ;FLAG FOR LSI-11,11/03
   10900                                        ;
   11000                                        ;       PROGRAM TRAP EQUATES
   11100                                        ;
   11200        104000                                  TYPE=EMT+0
   11300        104001                                  ERROR=EMT+1
   11400        104002                                  EHALT=EMT+2
   11500        104003                                  STRDRV=EMT+3
   11600        104004                                  STPCRV=EMT+4
   11700        104005                                  CHAIN=EMT+5
   11800        104006                                  CHALT=EMT+6
   11900        104007                                  TYPEM=EMT+7
   12000        104010                                  DELAY=EMT+10
   12100        104011                                  TTYCTL=EMT+11
   12200        104012                                  CRLF=EMT+12
   12300        104013                                  SCRLF=EMT+13
```

```
   12400        104014                                  LF=EMT+14
   12500        104015                                  PRINTC=EMT+15
   12600        104016                                  PRTHDR=EMT+16
   12700        104017                                  PRNT=EMT+17
   12800        104020                                  READ=EMT+20
   12900        104021                                  AREAD=EMT+21
   13000        104022                                  CR=EMT+22
   13100        104023                                  BTOASC=EMT+23
   13200        104024                                  FORWD=EMT+24
   13300        104025                                  READC=EMT+25
```

```
                                        .SBTTL  TRAP CATCHER & STARTING ADDRESSES
                                ;
000000                                  .ENABL  ABS,AMA
000000                                  .ASECT

        000000                          .=0

000000  000002                          .+2                     ;UNASSIGNED TRAP
000002  000000                          HALT
000004  000006                  MACHER: .+2                     ;SP OVERFLOW, BUS ERROR TRAP
000006  000000                          HALT
000010  000012                          .+2                     ;RESERVED INSTRUCTION TRAP
000012  000000                          HALT
000014  000016                          .+2                     ;TRACE TRAP
000016  000000                          HALT
000020  000022                          .+2                     ;TRAP TO CALL IOX
000022  000000                          HALT
000024  000026                          .+2                     ;POWER FAIL TRAP
000026  000000                          HALT
000030  002722                          EMTINT                  ;EMT TRAP
000032  000340                          PRTY7

        000042                          .=42

000042  000000                          0

        000046                          .=46

000046  011522                          LOGICAL

        000052                          .=52

000052  010000                          010000

        000174                          .=174

000174  000000                  DISPREG:        .WORD   0       ;SOFTWARE DISPLAY
000176  000000                  SWREG:          .WORD   0       ;SOFTWARE SWITCH REGISTER

000200  000167  000604                  JMP     START           ;START UP WITH I/O TESTS RUNNING
000204  000167  000526                  JMP     START1          ;START UP, SKIP ALL I/O TESTS
000210  000167  000540                  JMP     START2          ;START UP TERMINAL CONTROL WITH I/O TESTS
000214  000167  000552                  JMP     START3          ;START UP TERMINAL CONTROL WITHOUT I/O TESTS


        000600                          .=600

000600  000000                  SPBOT:  0                       ;BOTTOM OF STACK
```

```
                                        .SBTTL  SYMBOL DEFINITIONS
                                ;
                                ;
                                ;
000602  177560                  CONADD: 177560                  ;ADDR OF CONSOLE RECEIVER STATUS REG
000604  000060                  CONVEC: 60                      ;CONSOLE TERMINAL INTERRUPT VECTOR
000606  176500                  DLADR:  176500                  ;ADDRESS OF FIRST DL11, DEFAULT TO DL11-A,B
                                                                ;IF DL11-C,D,E,, THEN
                                                                ;SET TO 175610 FOR FIRST 16 (OF 31) OR
                                                                ;SET TO 176000 FOR LAST 16 (OF 31)
                                                                ;OR SET OTHER DESIRED START ADDRESS
000610  000020                  DLNR:   16.                     ;# OF DL11'S TO BE INITIALLY ASSUMED
000612  177560                  TKS:    177560                  ;CONSOLE RECEIVER STATUS REG
000614  177562                  TKB:    177562                  ;CONSOLE RECEIVER BUFFER
000616  177564                  TPS:    177564                  ;CONSOLE TRANSMITTER STATUS REG
000620  177566                  TPB:    177566                  ;CONSOLE TRANSMITTER BUFFER
000622  000060                  TKVTR:  60                      ;C.T. RECEIVER INTERRUPT VECTOR
000624  000200                  TKLVL:  PRTY4                   ;C.T. RECEIVER PRIORITY LEVEL
000626  000064                  TPVTR:  64                      ;C.T. TRANSMITTER INTERRUPT VECTOR
000630  000200                  TPLVL:  PRTY4                   ;C.T. TRANSMITTER PRIORITY LEVEL
000632  000000                  FSTDL:  OPEN                    ;ADDRESS OF FIRST ACTIVE DL11
000634  000000                  CNTLSW: OPEN                    ;CONSOLE TERMINAL CONTROL SWITCH
000636  000000                  RTNNO:  OPEN                    ;CONTAINS CURRENT TEST NUMBER
000640  000000                  NXTST:  OPEN                    ;CONTAINS ADDRESS OF NEXT TEST
000642  000000                  SCOPTR: OPEN                    ;CONTAINS ADDRESS OF TEST SCOPE ENTRY
000644  000000                  PRGID:  OPEN                    ;CONTAINS TEST PROGRAM INDICATORS
000646  000000                  CRBUF:  OPEN
000650  000000                  CTRA:   OPEN
000652  000000                  WIDTH:  OPEN                    ;CURRENT PAPER WIDTH, BINARY
000654  000000                  LEVEL:  OPEN                    ;LEVEL OF EXECUTION
000656  000000                  DLCNT:  OPEN                    ;# OF MULTIPLE DL11S
000660  000000                  ICTR:   OPEN                    ;I/O TEST ITERATION COUNT
000662  000000                  REPT:   OPEN                    ;TEMP STORAGE FOR TESTS E021 & E022
000664  000000                  BRCTR:  OPEN                    ;COUNTER FOR ROUTINE "AREAD"
000666  000000                  COUNT3: OPEN                    ;COUNTER FOR ROUTINE "PRINTC"
000670  000000                  XCSR:   OPEN                    ;ADDRESS OF MULTIPLE DL11 STATUS
000672  000251                  TIMER:  251                     ;1 MSEC COUNTER FOR ROUTINE "DELAY"
000674  000000                  SPCNT:  OPEN                    ;COUNTER FOR TEST ROUTINE "PT3"
000676  000000                  CURTST: OPEN                    ;ADDRESS OF CURRENT TEST
000700  000000                  TEMPCH: OPEN                    ;TEMP STOR FOR ECHO TESTS
000702  000000                  PARITY: OPEN                    ;PARITY FLAG FOR RECEIVED CHAR
000704  000000                  PCHAR:  OPEN                    ;CHAR CODE WITH PARITY BIT
000706  000000                  LFCNT:  OPEN                    ;COUNTER FOR TEST ROUTINE "PT4"
000710  000000                  INCHK:  OPEN                    ;CHECK FOR INPUT FLAG
000712  000000                  TEMP:   OPEN                    ;TEMPORARY WORKING STORAGE
000714  177570                  SR:     177570                  ;SW REG ADDRESS
000716  000000                  CNTR:   OPEN                    ;TIME COUNTER FOR LSI-11 TESTS
```

```
                                                .SBTTL  PROGRAM INITIALIZATION & CONTROL

                                        ;********
                                        ;COMMON HALT---WHEN IN SWITCH REGISTER CONTROL THE CPU
                                        ;       WILL BE ADVANCED TO THIS COMMON HALT WHERE
                                        ;       A NEW TEST WILL BE EXPECTED TO BE STARTED
                                        ;********
 000720  005767  177730                 CHLT:   TST     LEVEL           ;TEST CURRENT LEVEL
 000724  001403                                 BEQ     SELHLT          ;BRANCH IF 0, DO NOT HALT
 000726  011600                                 MOV     @SP,R0          ;PUT ADDRESS OF CALLER INTO R0
 000730  005740                                 TST     -(R0)
 000732  000000                                 HALT
 000734  000002                         SELHLT: RTI                     ;RETURN FROM INTERRUPT
 000736  012767  177777  004324         START1: MOV     #177777,ATOX    ;FORCE END OF I/O TESTS
 000744  012767  104006  000726                 MOV     #CHALT,WAITF    ;FORCE SR CONTROL
 000752  000424                                 BR      STARTX
 000754  012767  104011  000716         START2: MOV     #TTYCTL,WAITF   ;FORCE TERMINAL CONTROL
 000762  012767  005320  004300                 MOV     #AT1,ATOX       ;FORCE ALL I/O TESTS
 000770  000415                                 BR      STARTX
 000772  012767  104011  000700         START3: MOV     #TTYCTL,WAITF   ;FORCE TERMINAL CONTROL
 001000  012767  177777  004262                 MOV     #177777,ATOX    ;FORCE END OF I/O TESTS
 001006  000406                                 BR      STARTX
 001010  012767  005320  004252         START:  MOV     #AT1,ATOX       ;FORCE ALL I/O TESTS
 001016  012767  104006  000654                 MOV     #CHALT,WAITF    ;FORCE SR CONTROL
 001024  012706  000600                 STARTX: MOV     #SPBOT,SP       ;SET STACK POINTER
 001030  016746  176752                         MOV     6,-(SP)         ;SAVE CURRENT VECTOR
 001034  016746  176744                         MOV     4,-(SP)
 001040  012767  001054  176736                 MOV     #10$,4          ;SET UP TIMEOUT VECTOR
 001046  005777  177642                         TST     @SR             ;TRY TO REFERENCE HARDWARE SW REG
 001052  000411                                 BR      11$             ;BRANCH IF NO TIMEOUT TRAP OCCURS
 001054  012767  000176  177632         10$:    MOV     #SWREG,SR       ;POINT TO SOFTWARE SWITCH REGISTER
 001062  022626                                 CMP     (SP)+,(SP)+     ;RESTORE STACK
 001064  104000                                 TYPE                    ;TELL OPERATOR TO USE SOFTWARE
 001066  014424                                 NOSWR                   ;SWITCH REG AT LOC 176
 001070  012767  000202  177574                 MOV     #202,TIMER      ;ADJUST TIMER FOR LSI-11
 001076  012667  176702                 11$:    MOV     (SP)+,4         ;RESTORE TIMEOUT VECTOR
 001102  012667  176700                         MOV     (SP)+,6
 001106  005067  177576                         CLR     INCHK           ;ALLOW INPUT CHECKING
 001112  012767  000006  176664                 MOV     #6,MACHER       ;CLEAN UP
 001120  005067  177520                         CLR     PRGID           ;INITIALIZE PROGRAM FLAGS
 001124  005067  177504                         CLR     CNTLSW          ;INITIALIZE TERMINAL CONTROL SWITCH
 001130  005067  177520                         CLR     LEVEL           ;INITIALIZE LEVEL
 001134  012767  003460  176662                 MOV     #PFAIL,24       ;SET ADDR POWER FAIL ROUTINE
 001142  004767  002542                         JSR     PC,CONIT        ;SET UP CONSOLE TERMINAL ADDRESS
```

```
                                ;**********
                                ;READ THE PAPER WIDTH, NUMBER OF COLUMNS,
                                ;FROM SWITCH REGISTER POSITIONS 0-7. SAVE AND
                                ;CONVERT TO 3 ASCII CHARACTERS. A WIDTH GT132
                                ;OR LT30 COLUMNS (DECIMAL) WILL BE ABORTED TO 132.
                                ;THE SWITCHES MAY BE CHANGED ONCE THE PROGRAM TITLE OR THE DL11 COUNT
                                ;MESSAGE HAS STARTED TO PRINT.
                                ;**********

001146  017701  177542                  MOV     @SR,R1          ;PUT (SR) INTO R1
001152  042701  177400                  BIC     #177400,R1      ;SAVE ONLY BITS 0-7
001156  020127  000204                  CMP     R1,#204         ;TEST NO. COLUMN GT132
001162  003003                          BGT     2$              ;COLUMNS GT132, DEFAULT TO 132
001164  020127  000035          1$:     CMP     R1,#35          ;CHECK IF NO. COLUMNS LT 30
001170  101002                          BHI     3$              ;NOT LT 30 NOR GT 132
001172  012701  000204          2$:     MOV     #204,R1         ;COLUMNS LT 30 OR GT 132, DEFAULT
001176  010167  177450          3$:     MOV     R1,WIDTH        ;SAVE NO. COLUMNS IN WIDTH
001202  012700  014127                  MOV     #HDR0,R0        ;ADDR TO STORE ASC11 COLUMN VALUE
001206  012702  000003                  MOV     #3,R2           ;DO A 3 CHAR. CONVERSION
001212  104023                          BTOASC                  ;CONVERT NO. COLUMNS TO ASCII
001214  000401                  4$:     BR      5$
001216  000410                          BR      6$
001220  012700  000000          5$:     MOV     #0,R0           ;TRANSMIT A
001224  104015                          PRINTC                  ;NULL CODE
001226  104007                          TYPEM                   ;TYPE PROGRAM TITLE FIRST TIME RUN
001230  013676                          STARTM
001232  012767  000240  177754          MOV     #NOP,4$
```

```
                                ;**********
                                ;THIS NEXT PART CHECKS THE PRESENCE OF DL11-A OR DL11-C
                                ;STARTING AT 776500. A MESSAGE WILL BE PRINTED INDICATING THE NUMBER
                                ;PRESENT. THE PRINTER DIAGNOSTIC WILL ADDRESS EACH OF
                                ;THE MULTIPLE DL11S IN THE SYSTEM IF SWITCH 13 IS DOWN (0).
                                ;**********

001240  012767  001320  176536  6$:     MOV     #END2A,MACHER   ;INITIALIZE TIME OUT TRAP
001246  016700  177334                  MOV     DLADR,R0        ;ADDRESS OF FIRST DL11 TO R0
001252  016701  177332                  MOV     DLNR,R1         ;SET DL CHECK COUNT
001256  005067  177374                  CLR     DLCNT           ;INITIALIZE DLCNT
001262  005710                  END3:   TST     (R0)            ;IS DL PRESENT?
001264  012767  001332  176512          MOV     #END2,MACHER    ;YES, RESET TIME OUT TRAP
001272  010067  177334                  MOV     R0,FSTDL        ;STORE ADDRESS OF FIRST DL11
001276  000401                          BR      2$              ;CONTINUE
001300  005710                  1$:     TST     (R0)            ;IS DL11 PRESENT
001302  062700  000010          2$:     ADD     #10,R0          ;POINTER AND DL11 ADDRESS
001306  005267  177344                  INC     DLCNT           ;INCREMENT COUNT OF DL11'S
001312  005301                          DEC     R1              ;DECREMENT DL CHECK COUNT, DONE?
001314  001407                          BEQ     END4            ;BRANCH IF DONE
001316  000770                          BR      1$              ;CHECK PRESENCE OF NEXT DL11
001320  005301                  END2A:  DEC     R1              ;DONE DL CHECK?
001322  001404                          BEQ     END4            ;YES, EXIT
001324  062700  000010                  ADD     #10,R0          ;NO, CHECK NEXT DL
001330  000754                          BR      END3            ;CONTINUE
001332  022626                  END2:   POPSP2                  ;DL11 NOT PRESENT
001334  016701  177316          END4:   MOV     DLCNT,R1        ;GET # DL11'S
001340  012700  014064                  MOV     #DL11S1,R0      ;ADR OF ASCII CHAR STORAGE
001344  012702  000002                  MOV     #2,R2           ;# OF ASCII CHARS
001350  104023                          BTOASC                  ;CONVERT NUMBER
001352  104007                          TYPEM                   ;TYPE MESSAGE
001354  014051                          DL11S

                                ;**********
                                ;EXECUTE THE STRING OF CONSOLE TERMINAL I/O TESTS
                                ;THEN EITHER HALT AT LOCATION SELHLT OR CONTINUE WITH
                                ;PRINTER TESTS AS A FUNCTION OF SR BIT 8.
                                ;**********

001356  005067  177254                  CLR     RTNNO           ;SET ROUTINE NO = 0
001362  005067  177266                  CLR     LEVEL           ;SET LEVEL = 0
001366  026727  003676  177777          CMP     ATOX,#177777    ;SEE IF I/O IS TO BE SKIPPED
001374  001515                          BEQ     SKIP
001376  012767  005266  177234          MOV     #ATO,NXTST      ;ADDRESS OF FIRST I/O TEST
001404  104024                          FORWD                   ;SET UP TEST PARAMETERS
001406  000177  177264                  JMP     @CURTST         ;GO TO I/O TEST ROUTINE
```

```
                                ;**********
                                ;CHAINN--   THIS PORTION IS THE COMMON RETURN
                                ;           FOR ALL THREE CLASSES OF TESTS.
                                ;
                                ;        1--IF AN ERROR OCCURRED DURING AN I/O TEST THE
                                ;           OPERATOR CAN CAUSE THAT TEST TO BE LOOPED
                                ;           WITHOUT ANY FURTHER ERROR HALTS BY
                                ;           SETTING THE "SCOPE" BIT (#14) ON THE SR=1.
                                ;           RESETTING SR BIT 14 TO 0 WILL ALLOW THE
                                ;           ERROR HALT TO OCCUR AGAIN IF IT STILL EXISTS.
                                ;
                                ;        2--IF THE OPERATOR IS IN THE MAINTENANCE
                                ;           MODE (BIT 8 SET = 1 AT START UP TIME), THE
                                ;           SELECTED PROGRAM WILL LOOP CONTINUOUSLY
                                ;           IF SR BIT 11 IS SET=1. IF BIT 11 IS = 0
                                ;           THEN THE PROGRAM WILL BE ADVANCED TO
                                ;           THE NEXT TEST IN IT'S CLASS IF BIT 8=0.
                                ;           AS LONG AS BIT 11 AND
                                ;           BIT 8 ARE 0, THE CLASS OF TESTS SELECTED
                                ;           WILL BE CONTINUOUSLY SEQUENCED THROUGH.
                                ;           IF BIT 11 IS 0 AND BIT 8=1, THEN THE CPU
                                ;           WILL HALT AT LOCATION SELHLT AND WAIT FOR THE
                                ;           NEXT TEST NUMBER TO BE SET IN THE
                                ;**********SWITCH REGISTER.

001412 032767 000001 177214     CHAINN: BIT     #1,CNTLSW       ;CHECK IF TERMINAL CONTROL
001420 001401                           BEQ     1$              ;BRANCH IF NOT
001422 104011                           TTYCTL                  ;GO TO TERMINAL CONTROL
001424 005767 177214            1$:     TST     PRGID           ;TEST ERROR BIT IN PRGID
001430 100016                           BPL     3$              ;BRANCH IF ERROR BIT NOT SET
001432 032777 040000 177254             BIT     #SCOPSW,@SR     ;ERROR, CHECK IF SCOPE OPTION ON
001440 001407                           BEQ     2$              ;BRANCH IF NO SCOPING
001442 022767 177777 177172             CMP     #-1,SCOPTR      ;YES, CHECK IF OK TO SCOPE THIS TEST
001450 001403                           BEQ     2$              ;BRANCH IF NOT OK
001452 017716 177164                    MOV     @SCOPTR,@SP     ;PUT ADDR OF SCOPE ENTRY INTO STACK
001456 000002                           RTI                     ;GO TO SCOPE ENTRY IN TEST
001460 042767 100000 177156     2$:     BIC     #BIT15,PRGID    ;CLEAR ERROR IND. IN PRGID
001466 005767 177162            3$:     TST     LEVEL           ;CHECK LEVEL
001472 001405                           BEQ     4$              ;BRANCH IF LEVEL=0
001474 032777 004000 177212             BIT     #NITRSW,@SR     ;TEST LOOP SWITCH ON (=1)
001502 001405                           BEQ     5$              ;BRANCH IF NO LOOP TEST
001504 000002                           RTI                     ;GO BACK TO TEST
001506 005367 177146            4$:     DEC     ICTR            ;DECREMENT TEST ITERATION COUNT
001512 001407                           BEQ     6$              ;BRANCH IF COUNT=0
001514 000002                           RTI                     ;NOT ZERO, REPEAT TEST
001516 032777 000400 177170     5$:     BIT     #BIT8,@SR       ;TEST IF SEQUENCE TEST (BIT8)
001524 001402                           BEQ     6$              ;BRANCH TO NEXT TEST IF BIT8=0
001526 000167 000146                    JMP     WAITF           ;GO WAIT FOR MORE INPUT
001532 022626                   6$:     POPSP2                  ;POP 2 OFF STACK
001534 000240                   CHAINY: NOP                     ;THIS FORMERLY WAS RESET
001536 005777 177152                    TST     @SR             ;CHECK SR
001542 100003                           BPL     1$              ;BRANCH IF NO HALT WANTED
001544 116700 177066                    MOVB    RTNNO,R0        ;CURRENT TEST NUMBER TO R0
001550 000000                           HALT                    ;HALT (NOT FOR TEST SELECTION)
001552 005767 177076            1$:     TST     LEVEL           ;TEST THE CURRENT LEVEL
001556 001420                           BEQ     3$              ;BRANCH IF 0
```

```
001560 012767 000006 176216             MOV     #6,MACHER       ;CLEAN UP
001566 012706 000600                    MOV     #SPBOT,SP       ;SET UP STACK POINTER
001572 104024                           FORWD                   ;SET UP VALUES FOR NEXT TEST
001574 022767 177777 177036             CMP     #-1,NXTST       ;END OF I/O TESTS (=-1)
001602 001004                           BNE     2$              ;BRANCH IF NOT END
001604 012767 005266 177026             MOV     #AT0,NXTST      ;RESET NXTST TO FIRST I/O TEST
001612 104024                           FORWD                   ;SET UP VALUES FOR NEXT TEST
001614 000177 177056            2$:     JMP     @CURTST         ;GO TO TEST
001620 022767 177777 177012     3$:     CMP     #-1,NXTST       ;END OF I/O TESTS (=-1)
001626 001012                           BNE     NEXT            ;BRANCH IF NOT
001630 032777 000400 177056     SKIP:   BIT     #BIT8,@SR       ;TEST IF WANT TEST SELECTION RIGHT AWAY
001636 001016                           BNE     NEXT1           ;BRANCH IF NOT
001640 052767 000200 176776             BIS     #BIT7,PRGID     ;BYPASS SCOPING
001646 012767 007372 176764             MOV     #PT0,NXTST      ;PROD TESTING, GO TO PRINTER TESTS
001654 012767 000006 176122     NEXT:   MOV     #6,MACHER       ;CLEAN UP
001662 012706 000600                    MOV     #SPBOT,SP       ;SET UP STACK POINTER
001666 104024                           FORWD                   ;SET UP NEXT TEST PARAMETERS
001670 000177 177002                    JMP     @CURTST         ;GO TO ROUTINE
001674 005267 176754            NEXT1:  INC     LEVEL

                                ;**********
                                ;WAIT FOR FURTHER INSTRUCTIONS:
                                ;           -LOAD PROGRAM NUMBER INTO BITS 0-5 OF THE SR
                                ;           -SET SR BIT 11=1 TO LOOP ON SELECTED TEST
                                ;           -SET SR BIT 11=0 AND BIT 8=0 TO LOOP THROUGH
                                ;            SEQUENCE OF SELECTED TESTS.
                                ;           -SET SR BIT 11=0 AND BIT 8=1 TO HALT AGAIN AFTER
                                ;            EXECUTING TEST ONCE
                                ;**********

001700 104006                   WAITF:  CHALT                   ;OR TTYCTL IF START WAS AT 210
001702 012767 000006 176074             MOV     #6,MACHER       ;CLEAN UP
001710 012706 000600                    MOV     #SPBOT,SP       ;SET UP STACK POINTER
001714 017700 176774                    MOV     @SR,R0          ;GET CURRENT SW REG
001720 042700 177700                    BIC     #177700,R0
001724 020027 000037                    CMP     R0,#37          ;TEST IF PROG NO. IS I/O TEST
001730 101403                           BLOS    1$              ;BRANCH IF EQ OR LT 37. AN ECHO OR PRINTER
001732 005067 176706                    CLR     PRGID           ;I/O TEST, CLEAR PRGID
001736 000403                           BR      2$
001740 052767 000200 176676     1$:     BIS     #BIT7,PRGID     ;BYPASS SCOPING
001746 000241                   2$:     CLC                     ;CLEAR C BIT
001750 006100                           ROL     R0              ;GET PROGRAM ADDRESS OUT OF
001752 016067 002522 176660             MOV     PRGTAB(R0),NXTST;PROGRAM ADDRESS TABLE
001760 026727 176654 001700             CMP     NXTST,#WAITF    ;TEST IF LEGAL TEST NO.
001766 001744                           BEQ     WAITF           ;BRANCH IF ILLEGAL
001770 104024                           FORWD                   ;SET UP TEST PARAMETERS
001772 000177 176700                    JMP     @CURTST         ;GO TO TEST
```

```
                                ;**********
                                ;TTY1-- THIS SECTION IS USED WHEN THE DIAGNOSTIC IS BEING CONTROLLED BY
                                ;       THE CONSOLE TERMINAL, IT IS EFFECTIVE ONLY WHEN THE DIAGNOSTIC
                                ;       STARTING ADDRESS IS 210 AND SR BIT 8 WAS SET AT START TIME.
                                ;       THE RESPONSE TO THE MESSAGE "SELECT TEST NO." MUST BE THE 2
                                ;       DIGIT OCTAL TEST NUMBER FOLLOWED BY :
                                ;               "L" TO LOOP ON TEST
                                ;               "S" TO LOOP ON SEQUENCE
                                ;               "." TO EXECUTE TEST ONCE
                                ;       ALL SPACES WILL BE IGNORED. AN ILLEGAL INPUT WILL BE FLAGGED BY A "?"
                                ;       AND THE RETYPING OF THE ABOVE MESSAGE.
                                ;**********

001776  022626                  TTY1:   POPSP2                  ;POP 2 FROM STACK
002000  105777  176606                  TSTB    @TKS            ;TEST IF ANY INPUT
002004  100013                          BPL     1$              ;BRANCH IF NOT
002006  017705  176602                  MOV     @TKB,R5         ;GET CHAR
002012  042705  177600                  BIC     #177600,R5      ;MASK BITS
002016  020527  000177                  CMP     R5,#177         ;CHECK IF RUBOUT
002022  001004                          BNE     1$              ;BRANCH IF NOT
002024  042767  004400  176602          BIC     #4400,CNTLSW    ;CLEAR LOOP BITS
002032  000413                          BR      TTY1B
002034  032767  004000  176572  1$:     BIT     #NITRSW,CNTLSW  ;CHECK IF LOOP ON TEST
002042  001401                          BEQ     2$              ;BRANCH IF NO LOOP ON TEST
002044  000002                          RTI                     ;LOOP ON TEST
002046  032767  000400  176560  2$:     BIT     #BIT8,CNTLSW    ;TEST IF LOOP ON SEQUENCE
002054  001402                          BEQ     TTY1B           ;BRANCH IF NO LOOP ON SEQUENCE
002056  000167  177452                  JMP     CHAINY          ;CHAIN TO NEXT TEST
002062  012767  177777  176620  TTY1B:  MOV     #-1,INCHK       ;STOP INPUT CHECKING
002070  012700  000036                  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
002074  104010                          DELAY
002076  104007                          TYPEM
002100  014341                          MESG3                   ;TYPE MESSAGE
002102  005067  176602                  CLR     INCHK           ;ALLOW INPUT CHECKING AGAIN
002106  104020                  1$:     READ                    ;WAIT FOR INPUT
002110  026727  176564  000040          CMP     TEMPCH,#40      ;TEST IF CHAR IS A SPACE
002116  001773                          BEQ     1$              ;BRANCH IF YES
002120  012700  000036                  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
002124  104010                          DELAY
002126  104017                          PRNT                    ;READY?
002130  117777  176460  176462          MOVB    @TKB,@TPB       ;ECHO CHAR
002136  004767  000316                  JSR     PC,TESTC        ;CHECK IF CHAR IS OK
002142  000541                          BR      8$              ;NO, ERROR
002144  010005                          MOV     R0,R5           ;OK, PUT CHAR INTO R5
002146  006305                          ASL     R5              ;SHIFT INTO POSITION 5-3
002150  006305                          ASL     R5
002152  006305                          ASL     R5
002154  104020                  2$:     READ                    ;WAIT FOR NEXT CHAR
002156  026727  176516  000040          CMP     TEMPCH,#40      ;CHECK IF A SPACE
002164  001773                          BEQ     2$              ;BRANCH IF SPACE
002166  012700  000036                  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
002172  104010                          DELAY
002174  104017                          PRNT                    ;READY?
002176  117777  176412  176414          MOVB    @TKB,@TPB       ;ECHO CHAR
002204  004767  000250                  JSR     PC,TESTC        ;CHECK IF CHAR IS OK
002210  000516                          BR      8$              ;ERROR IN CHAR
002212  060005                          ADD     R0,R5           ;OK,R5 NOW = OCTAL TEST NO.
```

```
002214  104020                  3$:     READ                    ;WAIT FOR TERMINATION CHARACTER
002216  026727  176456  000040          CMP     TEMPCH,#40      ;CHECK IF SPACE
002224  001773                          BEQ     3$              ;BRANCH IF SPACE
002226  012700  000036                  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
002232  104010                          DELAY
002234  104017                          PRNT                    ;READY?
002236  117777  176352  176354          MOVB    @TKB,@TPB       ;ECHO CHAR
002244  012767  004001  176362          MOV     #4001,CNTLSW    ;SET BITS 11 & 0
002252  026727  176422  000114          CMP     TEMPCH,#114     ;NO, IS IT AN "L" ?
002260  001427                          BEQ     5$              ;BRANCH IF YES
002262  026727  176412  000154          CMP     TEMPCH,#154     ;CHECK LOWER CASE
002270  001423                          BEQ     5$
002272  026727  176402  000123          CMP     TEMPCH,#123     ;NO, IS IT AN "S"
002300  001414                          BEQ     4$              ;BRANCH IF YES
002302  026727  176372  000163          CMP     TEMPCH,#163     ;CHECK LOWER CASE
002310  001410                          BEQ     4$
002312  026727  176362  000056          CMP     TEMPCH,#56      ;NO, IS IT A "." ?
002320  001052                          BNE     8$              ;NO, ERROR
002322  012767  000001  176304          MOV     #1,CNTLSW       ;YES SET ONLY BIT 0 IN CONTROL WD
002330  000403                          BR      5$
002332  012767  000401  176274  4$:     MOV     #401,CNTLSW     ;SET BITS 8 & 0
002340  012767  000006  175436  5$:     MOV     #6,MACHER       ;CLEAN UP
002346  012706  000600                  MOV     #SPBOT,SP       ;INIT SP
002352  020527  000040                  CMP     R5,#40          ;IS THIS AN I/O TEST
002356  103033                          BHIS    8$              ;BRANCH IF YES
002360  020527  000030                  CMP     R5,#30          ;IS THIS AN OPTION TEST?
002364  103007                          BHIS    6$              ;SKIP IF YES
002366  020527  000020                  CMP     R5,#20          ;IS THIS AN ECHO TEST
002372  103404                          BLO     6$              ;BRANCH IF NOT
002374  012767  000001  176232          MOV     #1,CNTLSW       ;FORCE ECHO TEST TO A SINGLE RUN
002402  000402                          BR      7$              ;LEAVE THIS TERMINAL AS CONSOLE
002404  004767  001300          6$:     JSR     PC,CONIT        ;RESET CONSOLE TERMINAL ADDRESS
002410  052767  000200  176226  7$:     BIS     #BIT7,PRGID     ;BYPASS SCOPING
002416  000241                          CLC                     ;CLEAR C BIT
002420  006105                          ROL     R5
002422  016567  002522  176210          MOV     PRGTAB(R5),NXTST;ADDR OF TEST TO NXTST
002430  026727  176204  001700          CMP     NXTST,#WAITF    ;CHECK IF TEST EXISTS
002436  001403                          BEQ     8$              ;BRANCH IF NOT
002440  104024                          FORWD                   ;SET UP TEST PARAMETERS
002442  000177  176230                  JMP     @CURTST         ;GO TO TEST
002446  104017                  8$:     PRNT                    ;CHECK IF PRINTER IS READY
002450  112777  000077  176142          MOVB    #77,@TPB        ;SEND A "?"
002456  000601                          BR      TTY1B           ;TRY AGAIN
```

```
32800                                 ;TESTC--CHECKS THAT THE INPUTTED CHARACTER IS BETWEEN 0 AND 7 INCLUSIVE
32900
33000 002460  026727  176214  000060  TESTC:  CMP     TEMPCH,#60      ;CHECK IF NUMERIC AND EQ OR GT 0
33100 002466  103001                          BHIS    1$              ;BRANCH IF OK
33200 002470  000207                          RTS     PC              ;ERROR RETURN
33300 002472  026727  176202  000067  1$:     CMP     TEMPCH,#67      ;CHECK IF EQ OR LT 7
33400 002500  101401                          BLOS    2$              ;BRANCH IF OK
33500 002502  000207                          RTS     PC              ;ERROR RETURN
33600 002504  062716  000002          2$:     ADD     #2,@SP          ;SET UP RETURN ADDRESS
33700 002510  016700  176164                  MOV     TEMPCH,R0       ;GET CHAR
33800 002514  042700  177770                  BIC     #177770,R0      ;SAVE ONLV THE DIGIT
33900 002520  000207                          RTS     PC              ;NORMAL RETURN
```

```
34100 002522  007372                  PRGTAB: PT0                     ;DATA PATH TEST
34200 002524  007446                          PT1                     ;PRINTER CHARACTER TEST
34300 002526  007570                          PT2                     ;NON-PRINTING CHARACTER TEST
34400 002530  010164                          PT3                     ;CARRIAGE RETURN TEST
34500 002532  010304                          PT4                     ;MULTIPLE LINE FEED TEST
34600 002534  010462                          PT5                     ;SINGLE LINE FEED TEST
34700 002536  010666                          PT6                     ;BACKSPACE TEST
34800 002540  011054                          PT7                     ;OVERPRINT TEST
34900 002542  011266                          PT10                    ;PRINTING FREQUENCY SWEEP TEST
35000 002544  011424                          PT11                    ;RIBBON FEED TEST
35100 002546  011456                          PT12                    ;PRINTER BELL TEST
35200 002550  001700                          WAITF                   ;SPARE
35300 002552  001700                          WAITF                   ;SPARE
35400 002554  001700                          WAITF                   ;SPARE
35500 002556  001700                          WAITF                   ;SPARE
35600 002560  011546                          PT17                    ;LIFE TEST
35700 002562  012116                          EO20                    ;CHARACTER ECHO TEST
35800 002564  012166                          EO21                    ;LINE ECHO TEST, FAST RATE
35900 002566  012224                          EO22                    ;LINE ECHO TEST, SLOW RATE
36000 002570  012476                          EO23                    ;CHARACTER/CODE ECHO TEST
36100 002572  013020                          EO24                    ;SELECTIVE PATTERN ECHO TEST
36200 002574  013566                          EO25                    ;BELL ECHO TEST
36300 002576  001700                          WAITF                   ;SPARE
36400 002600  001700                          WAITF                   ;SPARE
36500 002602  001700                          WAITF           ;SPARE
36600 002604  001700                          WAITF           ;SPARE
36700 002606  001700                          WAITF           ;SPARE
36800 002610  001700                          WAITF           ;PRARE
36900 002612  001700                          WAITF           ;SPARE
37000 002614  001700                          WAITF           ;SPARE
37100 002616  001700                          WAITF                   ;SPARE
37200 002620  001700                          WAITF                   ;SPARE
37300 002622  005266                          AT0                     ;I/O TEST NO. 40
37400 002624  005320                          AT1                     ;I/O TEST NO. 41
37500 002626  005352                          AT2                     ;I/O TEST NO. 42
37600 002630  005404                          AT3                     ;I/O TEST NO. 43
37700 002632  005436                          AT4                     ;I/O TEST NO. 44
37800 002634  005526                          AT5                     ;I/O TEST NO. 45
37900 002636  005604                          AT6                     ;I/O TEST NO. 46
38000 002640  005674                          AT7                     ;I/O TEST NO. 47
38100 002642  005744                          AT10                    ;I/O TEST NO. 50
38200 002644  006002                          AT11                    ;I/O TEST NO. 51
38300 002646  006042                          AT12                    ;I/O TEST NO. 52
38400 002650  006116                          AT13                    ;I/O TEST NO. 53
38500 002652  006176                          AT14                    ;I/O TEST NO. 54
38600 002654  006262                          AT15                    ;I/O TEST NO. 55
38700 002656  006362                          AT16                    ;I/O TEST NO. 56
38800 002660  006430                          AT17                    ;I/O TEST NO. 57
38900 002662  006500                          AT20                    ;I/O TEST NO. 60
39000 002664  006572                          AT21                    ;I/O TEST NO. 61
39100 002666  006672                          AT22                    ;I/O TEST NO. 62
39200 002670  007000                          AT23                    ;I/O TEST NO. 63
39300 002672  007112                          AT24                    ;LSI TEST NO. 64
39400 002674  007212                          AT25                    ;LSI TEST NO. 65
39500 002676  007270                          AT26                    ;LSI TEST NO. 66
39600 002700  001700                          WAITF                   ;SPARE
39700 002702  001700                          WAITF                   ;SPARE
```

```
002704  001700                          WAITF                   ;SPARE
002706  001700                          WAITF                   ;SPARE
002710  001700                          WAITF                   ;SPARE
002712  001700                          WAITF                   ;SPARE
002714  001700                          WAITF                   ;SPARE
002716  001700                          WAITF                   ;SPARE
002720  001700                          WAITF                   ;SPARE

                                ;**********
                                ;EMTINT ------------SERVICE ROUTINE FOR TRAPS THROUGH
                                ;                   LOCATION 30.
                                ;**********

002722  011646                  EMTINT: MOV     @SP,-(SP)       ;PUSH STACKED PC TO GET A WORK COPY.   (Q)
002724  162716  000002                  SUB     #2,@SP          ;SUB 2 TO POINT TO CALLING TRAP INSTR.
002730  017616  000000                  MOV     @(SP),@SP       ;PLACE TRAP INSTR INTO THIS STACK WORK AREA.
002734  121627  000035                  CMPB    @SP,#35         ;EXAMINE ITS RIGHT SIDE.  (Q)
002740  101402                          BLOS    2$              ;BRANCH IF WITHIN RANGE OF ESTABLISHED TABLE.
002742  000000                  1$:     HALT                    ;ELSE HALT.
002744  000776                          BR      1$
002746  006116                  2$:     ROL     @SP             ;MULT INSTR BY 2 TO GET WORD DISPLACEMENT.
002750  042716  177001                  BIC     #177001,@SP     ;STRIP OFF OP CODE AND LS BIT.
002754  062716  002776                  ADD     #EMTTAB,@SP     ;ADD IN STARTING ADDRESS OF TABLE.
002760  017616  000000                  MOV     @(SP),@SP       ;FROM TABLE GET OUT DESIRED POINTER.
002764  005046                          CLR     -(SP)           ;PUSH A ZERO PSW.
002766  012746  002774                  MOV     #3$,-(SP)       ;PUSH A PC = TO #3$ OF THIS ROUTINE.
002772  000002                          RTI                     ;DO RTI (POP-POP) TO ESTABLISH THE ZERO PSW.
002774  000136                  3$:     JMP     @(SP)+          ;JMP TO ROUTINE LEAVING STACK AS FOUND.

002776  003076                  EMTTAB: TYP                     ;MESSAGE OUTPUT ROUTINE
003000  003320                          ERR                     ;I/O TEST ERROR ROUTINE
003002  003346                          EHLT                    ;UNCONDITIONAL HALT
003004  003356                          STLSRV                  ;KEYBOARD VECTOR/PRIORITY SETUP
003006  003406                          STLSPV                  ;PRINTER VECTOR/PRIORITY SETUP
003010  001412                          CHAINN                  ;COMMON TEST EXIT
003012  000720                          CHLT                    ;SR BIT 15 HALT
003014  003164                          TYPM                    ;MESSAGE OUTPUT ROUTINE, MULTI DEVICES
003016  003436                          DLY                     ;DELAY ROUTINE
003020  001776                          TTY1                    ;CONSOLE TERMINAL CONTROL
003022  003214                          $CRLF                   ;CARRIAGE RETURN-LINE FEED TO ALL DL11'S
003024  003142                          $SCRLF                  ;CARRIAGE RETURN-LINE FEED TO CONSOLE
003026  003216                          $LF                     ;LINE FEED ONLY (TO ALL)
003030  004324                          $PRTC                   ;PRINT CHAR
003032  003236                          $PRHDR                  ;PRINT TEST HEADER
003034  004314                          $PRNT                   ;PRINTER READY
003036  004112                          $READ                   ;READ CHAR
003040  003640                          $AREAD                  ;I/O TEST READ ROUTINE
003042  003226                          $CR                     ;CARRIAGE RETURN ONLY (TO ALL)
003044  004006                          $BTASC                  ;BINARY TO ASCII CONVERSION
003046  003562                          $FORWD                  ;FORWARD ROUTINE ( BETWEEN TESTS )
003050  004204                          $READC                  ;READ CONSOLE KYBD ONLY
003052  003072                          SPARET                  ;SPARE EMT
003054  003072                          SPARET                  ;SPARE EMT
003056  003072                          SPARET                  ;SPARE EMT
003060  003072                          SPARET                  ;SPARE EMT
003062  003072                          SPARET                  ;SPARE EMT
003064  003072                          SPARET                  ;SPARE EMT
```

```
003066  003072                          SPARET                  ;SPARE EMT
003070  003072                          SPARET                  ;SPARE EMT
003072  000000                  SPARET: HALT                    ;HALT IF TRAP TO UNDEFINED
003074  000776                          BR      SPARET          ;EMT IS ATTEMPTED.



                                        .SBTTL  COMMON ROUTINES USED BY LA36 TESTS

                                ;**********
                                ;
                                ;THIS SECTION CONTAINS MOST ROUTINES CALLED BY
                                ;THE VARIOUS TESTS EITHER BY TRAPPING THROUGH LOCATION
                                ;30 OR BY SUBROUTINE CALLS (JSR PC,***)
                                ;
                                ;**********
                                ;
                                ;TYPE-- A COMMON ROUTINE USED TO TYPE MESSAGES ON THE
                                ;       CONSOLE TERMINAL ONLY. THE NULL CHARACTER TERMINATES
                                ;       THE MESSAGE. CALLED THROUGH AN EMT TRAP.
                                ;       CALLING SEQUENCE
                                ;       TYPE
                                ;       MESG    ;ADDRESS OF MESSAGE
                                ;
                                ;**********
003076  010046                  TYP:    MOV     R0,-(SP)        ;SAVE R0
003100  016601  000002                  MOV     2(SP),R1        ;GET POINTER TO ADDR. OF MESG.
003104  062766  000002  000002          ADD     #2,2(SP)
003112  011101                          MOV     (R1),R1         ;ADDR. OF MESG TO R1
003114  112100                  1$:     MOVB    (R1)+,R0        ;GET CHAR
003116  100403                          BMI     2$              ;BRANCH IF WANT AUTO CR-LF
003120  001004                          BNE     3$              ;PRINT CHAR IF NOT NULL
003122  012600                          MOV     (SP)+,R0        ;RESTORE R0
003124  000002                          RTI                     ;EXIT IF NULL CHAR
003126  104013                  2$:     SCRLF                   ;YES, SEND CR-LF
003130  000771                          BR      1$              ;GET NEXT CHAR
003132  104017                  3$:     PRNT                    ;PRINTER READY?
003134  110077  175460                  MOVB    R0,@TPB         ;LOAD PRINTER BUFFER WITH CHAR
003140  000765                          BR      1$              ;GO GET NEXT CHAR

003142  104017                  $SCRLF: PRNT                    ;PRINTER READY?
003144  112777  000015  175446          MOVB    #15,@TPB        ;SEND CR
003152  104017                          PRNT                    ;PRINTER READY?
003154  112777  000012  175436          MOVB    #12,@TPB        ;SEND LF
003162  000002                          RTI                     ;RETURN TO CALLER
```

```
                                ;XXXXXXXXXX
                                ;
                                ;TYPM---MULTI TYPE-A COMMON ROUTINE TO OUTPUT
                                ;       A MESSAGE ON ALL DL11S IF THE MULTI TEST
                                ;       SWITCH (BIT 13) IS RESET. THIS ROUTINE IS USED BY
                                ;       THE PRINTER TESTS TO TYPE HEADINGS.  IF A UNIT
                                ;       IS NOT READY, THE CHARACTER WILL NOT BE TYPED.
                                ;
                                ;XXXXXXXXXX

003164  011601                  TYPM:   MOV     (SP),R1         ;GET POINTER TO ADDR OF MESG
003166  062716  000002                  ADD     #2,@SP
003172  011101                          MOV     (R1),R1         ;ADDR OF MESG TO R1
003174  112100                  1$:     MOVB    (R1)+,R0        ;GET CHAR
003176  100402                          BMI     2$              ;BRANCH IF WANT AUTO CR-LF
003200  001003                          BNE     3$              ;CONTINUE IF NOT NULL
003202  000002                          RTI                     ;RETURN
003204  104012                  2$:     CRLF                    ;YES, SEND CR-LF
003206  000772                          BR      1$              ;NEXT CHAR
003210  104015                  3$:     PRINTC                  ;PRINT CHAR
003212  000770                          BR      1$              ;GO GET NEXT CHAR.

003214  104022                  $CRLF:  CR                      ;SEND CR
003216  012700  000012          $LF:    MOV     #12,R0          ;SET LF CHAR
003222  104015                          PRINTC                  ;SEND IT
003224  000002                          RTI                     ;RETURN TO CALLER

003226  012700  000015          $CR:    MOV     #15,R0          ;SET CR CHAR
003232  104015                          PRINTC                  ;SEND IT
003234  000002                          RTI                     ;RETURN

                                ;**********
                                ;
                                ;ROUTINE TO PRINT TEST HEADER
                                ;
                                ;**********

003236  012700  000000          $PRHDR: MOV     #0,R0           ;TRANSMIT
003242  104015                          PRINTC                  ;NUL CODE.
003244  104007                          TYPEM                   ;PRINT MESSAGE
003246  014113                          HDRMSG
003250  016700  175362                  MOV     RTNNO,R0        ;GET TEST NUMBER
003254  006200                          ASR     R0              ;GET FIRST DIGIT
003256  006200                          ASR     R0
003260  006200                          ASR     R0
003262  042700  177770                  BIC     #177770,R0      ;MASK FIRST DIGIT
003266  062700  000060                  ADD     #60,R0          ;MAKE ASCII
003272  104015                          PRINTC                  ;PRINT DIGIT
003274  016700  175336                  MOV     RTNNO,R0        ;GET TEST NUMBER AGAIN
003300  042700  177770                  BIC     #177770,R0      ;MASK LAST DIGIT
003304  062700  000060                  ADD     #60,R0          ;MAKE ASCII
003310  104015                          PRINTC                  ;PRINT DIGIT
003312  104012                          CRLF                    ;CR-LF
003314  104014                          LF                      ;BLANK LINE
003316  000002                          RTI                     ;RETURN
```

```
                                ;**********
                                ;
                                ;ERRA-- COMMON ERROR RETURN FROM I/O TESTS. HALTS
                                ;       WITH ADDRESS OF ERROR IN R0. TO CONTINUE
                                ;       ON SAME TEST BUT NOT HALTING ON ERROR,
                                ;       SET THE SCOPE BIT (14) = 1 AND PRESS CONTINUE
                                ;
                                ;**********

003320  032777  040000  175366  ERR:    BIT     #SCOPSW,@SR     ;CHECK SCOPE SWITCH
003326  001404                          BEQ     1$              ;BRANCH IF NO SCOPE
003330  005767  175310                  TST     PRGID           ;SCOPING WANTED, FIRST ERROR?
003334  100001                          BPL     1$              ;BRANCH AND HALT ON FIRST ERROR
003336  000002                          RTI                     ;SCOPE EXIT
003340  052767  100000  175276  1$:     BIS     #BIT15,PRGID    ;SET ERROR INDICATOR
003346  011600                  EHLT:   MOV     @SP,R0
003350  005740                          TST     -(R0)           ;ADDRESS OF CALL INTO R0
003352  000000                          HALT
003354  000002                  ERRHLT: RTI                     ;RETURN TO TEST FOLLOWING CALL

                                ;**********
                                ;
                                ;STLSRV--- THIS ROUTINE SETS UP KEYBOARD INTERRUPT
                                ;           VECTOR AND PRIOITY. CALLING SEQUENCE
                                ;
                                ;       STRDRV
                                ;       AT20C   ;LOCATION OF NEW INTERRUPT VECTOR
                                ;
                                ;**********

003356  017667  000000  000012  STLSRV: MOV     @(SP),STPRA+2   ;SET RETURN ADR AND VECTOR
003364  062716  000002                  ADD     #2,@SP
003370  016701  175226                  MOV     TKVTR,R1
003374  012721  000000          STPRA:  MOV     #0,(R1)+
003400  016721  175220                  MOV     TKLVL,(R1)+
003404  000002                          RTI

                                ;**********
                                ;
                                ;STLSPV-- THIS ROUTINE SETS UP PRINTER INTERRUPT
                                ;
                                ;       VECTOR AND PRIORITY CALLING SEQUENCE
                                ;       STPCHV
                                ;       AT35E   ;LOCATION OF NEW INTERRUPT VECTOR
                                ;
                                ;**********

003406  017667  000000  000012  STLSPV: MOV     @(SP),STPPA+2   ;SET RETURN ADR AND VECTOR
003414  062716  000002                  ADD     #2,@SP
003420  016701  175202                  MOV     TPVTR,R1
003424  012721  000000          STPPA:  MOV     #0,(R1)+
003430  016721  175174                  MOV     TPLVL,(R1)+
003434  000002                          RTI                     ;RETURN TO CALLER
```

```
                                        ;**********
                                        ;
                                        ;DELAY--A COMMON ROUTINE TO DELAY PROCESSING
                                        ;       A GIVEN NUMBER OF MSEC.
                                        ;       CALLING SEQUENCE;
                                        ;       MOV   #5,R0    ;R0 CONTAINS THE NUMBER OF MSEC DELAY DESIRED
                                        ;       DELAY
                                        ;
                                        ;       THE DELAY IS EFFECTED BY THE EXECUTION OF THE LOOP;
                                        ;          1$:  DEC  R1
                                        ;               BNE  1$
                                        ;
                                        ;       SINCE THE EXECUTION TIMES OF THE PDP11 LINE DOES VARY FROM
                                        ;       MACHINE TO MACHINE, THE VALUE AT SYMBOLIC LOCATION
                                        ;       "TIMER" MUST BE CHANGED TO THE APPROPRIATE VALUE AS SHOWN BELOW
                                        ;       BEFORE STARTING THE DIAGNOSTIC. "TIMER" IS INITIALIZED
                                        ;       FOR AN 11/05,11/10(=251).
                                        ;
                                        ;MACHINE         05&10   35&40   15&20   LSI&03              11/45 & 11/70
                                        ;                                                   BIPOLAR        MOS       CORE
                                        ;
                                        ;LOOP: DEC R1    3.4      .99    2.3                 .30           .51       .90
                                        ;      BNE LOOP  2.5     1.76    2.6                 .60           .98      1.13
                                        ;  TIME=         5.9USEC 2.75    4.9     7.7        .90USEC       1.49USEC             2.03USEC
                                        ; SET TIMER      251     554     314     202        2127          1237       755
                                        ;
                                        ;XXXXXXXXXX

003436  010146                          DLY:    MOV     R1,-(SP)          ;SAVE R1
003440  016701  175226                  1$:     MOV     TIMER,R1          ;MOV 1 MSEC LOOP CNT TO R1
003444  005301                          2$:     DEC     R1                ;DECREMENT COUNT
003446  001376                                  BNE     2$                ;BRANCH IF NOT ZERO
003450  005300                                  DEC     R0                ;DEC NO. OF MSEC DELAY
003452  001372                                  BNE     1$                ;DELAY AGAIN IF NOT ZERO
003454  012601                                  MOV     (SP)+,R1          ;ALL DONE RESTORE R1
003456  000002                                  RTI
```

```
                                        ;**********
                                        ;
                                        ;PFAIL--POWER FAIL ROUTINE
                                        ;       SAVE ALL REGISTERS AND SET RESTART ADDRESS
                                        ;       INTO LOCATION 24
                                        ;
                                        ;RESTART--POWER FAIL RECOVERY
```

```
                                          ;       RESTORE ALL REGISTERS AND GO TO START
```

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

 003460  010046                           PFAIL:  MOV     R0,-(SP)
 003462  010146                                   MOV     R1,-(SP)
 003464  010246                                   MOV     R2,-(SP)
 003466  010346                                   MOV     R3,-(SP)
 003470  010446                                   MOV     R4,-(SP)
 003472  010546                                   MOV     R5,-(SP)
 003474  016746  174324                           MOV     24,-(SP)
 003500  010667  000010                           MOV     SP,SAVR6        ;SAVE STACK POSITION
 003504  012767  003516  174312                   MOV     #RESTRT,24      ;STORE RESTART ADDRESS
 003512  000000                                   HALT
 003514  000000                           SAVR6:  .WORD   0
 003516  104007                           RESTRT: TYPEM
 003520  003552                                   1$
 003522  016706  177766                           MOV     SAVR6,SP        ;RESTORE STACK POINTER
 003526  012667  174272                           MOV     (SP)+,24        ;RESTORE PFAIL ADDRESS
 003532  012605                                   MOV     (SP)+,R5
 003534  012604                                   MOV     (SP)+,R4
 003536  012603                                   MOV     (SP)+,R3
 003540  012602                                   MOV     (SP)+,R2
 003542  012601                                   MOV     (SP)+,R1
 003544  012600                                   MOV     (SP)+,R0
 003546  000167  175236                           JMP     START

 003552     200     120     117           1$:     .ASCIZ  <ACRLF>/POWER/<ACRLF>
 003555     127     105     122
 003560     200     000
                                                  .EVEN
```

```
                                ;**********
                                ;
                                ;FORWARD--THIS ROUTINE TRANSFERS THE 2 OR 4 ARGUMENTS
                                ;       FROM THE TEST ROUTINE. THEY ARE;
                                ;
                                ;       1- ROUTINE NUMBER
                                ;       2- ADDRESS OF NEXT TEST
                                ;       3- ITERATION COUNT (I/O TESTS ONLY)
                                ;       4- SCOPE ENTRY ADDRESS (I/O TESTS ONLY)
                                ;
                                ;**********

003562  016705  175052          $FORWD: MOV     NXTST,R5        ;ADDR OF NEXT TEST TO R5
003566  012567  175044                  MOV     (R5)+,RTNNO     ;GET NUMBER OF NEXT TEST
003572  012567  175042                  MOV     (R5)+,NXTST     ;GET ADDR OF FOLLOWING TEST
003576  105767  175042                  TSTB    PRGID           ;CHECK IF I/O TEST
003602  100407                          BMI     FORWDB          ;SKIP THE FETCH OF ITER CNT AND SCOPE
003604  012567  175050                  MOV     (R5)+,ICTR      ;GET ITERATION COUNT
003610  012567  175026                  MOV     (R5)+,SCOPTR    ;GET SCOPE ENTRY POINT
003614  010567  175056          FORWDA: MOV     R5,CURTST       ;ENTRY POINT TO TEST IN CUR TST
003620  000002                          RTI                     ;EXIT
003622  012767  177777  175012  FORWDB: MOV     #-1,SCOPTR      ;FORCE NO SCOPE
003630  012767  000001  175022          MOV     #1,ICTR         ;FORCE INTERATION COUNT OF 1
003636  000766                          BR      FORWDA
```

```
                                ;**********
                                ;
                                ;AREAD--A ROUTINE WHICH, THROUGH THE FACILITY OF
                                ;       THE MAINTENANCE BIT, OUTPUTS TO THE
                                ;       PRINTER BUFFER AND READS THE KEYBOARD
                                ;       STATUS DONE. IF THE DONE IS NOT SET
                                ;       WITHIN 600 MSEC, THE CPU WILL HALT WITH
                                ;       THE LOCATION OF THE ERROR IN R0. PRESS
                                ;       CONTINUE TO CONTINUE WITH TESTS.
                                ;
                                ;**********

003640  012767  000600  175016  $AREAD: MOV     #600,BRCTR      ;SET UP 600 MSEC DELAY
003646  052777  000004  174742          BIS     #4,@TPS         ;SET MAINTENANCE BIT
003654  005077  174740                  CLR     @TPB            ;LOAD PRINTER BUFFER
003660  105777  174726          1$:     TSTB    @TKS            ;CHECK DONE BIT
003664  100410                          BMI     2$              ;BRANCH IF DONE
003666  012700  000001                  MOV     #1,R0           ;ONE TO R0
003672  104010                          DELAY                   ;DELAY 1 MSEC.
003674  005367  174764                  DEC     BRCTR           ;600 MSEC OVER
003700  001367                          BNE     1$              ;BRANCH IF NO
003702  104002                          EHALT
003704  000755                          BR      $AREAD          ;TRY AGAIN
003706  000002                  2$:     RTI                     ;RETURN TO TEST
                                ;**********
                                ;
                                ;CONIT--THIS ROUTINE SETS UP THE DEVICE ADDRESSES
                                ;       AND INTERRUPT VECTORS FOR THE CONSOLE
                                ;       TERMINAL.
                                ;
                                ;**********

003710  016700  174666          CONIT:  MOV     CONADD,R0       ;CONSOLE KEYBOARD STATUS ADDR TO R0
003714  010067  174672          CONSET: MOV     R0,TKS          ;KEYBOARD STATUS ADDRESS (777560) TO TKS
003720  005720                          TST     (R0)+           ;INCREMENT R0 BY TWO
003722  010067  174666                  MOV     R0,TKB          ;KEYBOARD DATA ADDR (777562) TO TKB
003726  005720                          TST     (R0)+           ;INCREMENT R0 BY TWO
003730  016767  174662  000044          MOV     TPS,TPSS        ;SAVE TPS OF LAST TERMINAL
003736  010067  174654                  MOV     R0,TPS          ;PRINTER STATUS ADDR(777564) TO TPS
003742  005720                          TST     (R0)+           ;INCREMENT R0 BY TWO
003744  016767  174650  000032          MOV     TPB,TPBS        ;SAVE TPB OF LAST TERMINAL
003752  010067  174642                  MOV     R0,TPB          ;PRINTER DATA ADDR (777566) TO TPB
003756  016767  174622  174636          MOV     CONVEC,TKVTR    ;KEYBOARD INTERRUPT VECTOR (60) TO TKVTR
003764  016767  174614  174634          MOV     CONVEC,TPVTR
003772  062767  000004  174626          ADD     #4,TPVTR        ;PRINTER INTERRUPT VECTOR (64) TO TPVTR
004000  000207                          RTS     PC

004002  000000                  TPSS:   .WORD   0               ;LAST TERM STATUS REG ADR
004004  000000                  TPBS:   .WORD   0               ;LAST TERM BUFFER REG ADR
```

```
10600                                 ;**********
10700                                 ;
10800                                 ;BINARY TO ASCII CONVERSION (1 TO 5 ASCII CHARACTERS)
10900                                 ;CALLING SEQUENCE
11000                                 ;       MOV     ADDRESS OF LOC. TO STORE FIRST ASCII CHAR. INTO R0
11100                                 ;       MOV     BINARY NUMBER TO BE CONVERTED INTO R1
11200                                 ;       MOV     NUMBER TO BE CONVERTED AS A POWER OF TEN INTO R2
11300                                 ;       BTOASC
11400                                 ;
11500                                 ;**********
11600
11700 004006  010267  000060          $BTASC: MOV     R2,CNVCTR       ;SAVE TEN POWER
11800 004012  006302                          ASL     R2              ;R2*2
11900 004014  062702  004100                  ADD     #ADTENP,R2      ;CALCULATE ADDRESS OF
12000                                                                 ;STARTING TEN POWER
12100 004020  014267  000052          1$:     MOV     -(R2),TENPWR    ;POWER OF TEN VALUE TO TEN PWR
12200 004024  005067  000044                  CLR     DIGIT           ;CLEAR CURRENT DIGIT
12300 004030  166701  000042          2$:     SUB     TENPWR,R1       ;SUBTRACT TEN POWER FROM BINARY VALUE
12400 004034  103403                          BCS     3$              ;BRANCH IF END
12500 004036  005267  000032                  INC     DIGIT
12600 004042  000772                          BR      2$
12700 004044  066701  000026          3$:     ADD     TENPWR,R1       ;RESTORE SUBTRACTED VALUE
12800 004050  062767  000060  000016          ADD     #60,DIGIT       ;CONVERT (DIGIT) TO ASCII
12900 004056  116720  000012                  MOVB    DIGIT,(R0)+     ;PUT ASCII CHAR INTO USER BUFFER
13000 004062  005367  000004                  DEC     CNVCTR          ;FINISHED ALL CHARS. CALLED FOR
13100 004066  001354                          BNE     1$              ;BRANCH IF NOT FINISHED
13200 004070  000002                          RTI                     ;YES, EXIT
13300 004072  000000                  CNVCTR: .WORD   0               ;CONVERSION CHARACTER COUNT
13400 004074  000000                  DIGIT:  .WORD   0               ;CONVERTED CHARACTER
13500 004076  000000                  TENPWR: .WORD   0               ;CURRENT TEN POWER
13600 004100  000001  000012  000144  ADTENP: .WORD   1.,10.,100.,1000.,10000.
      004106  001750  023420
```

```
13800                                 ;XXXXXXXXXX
13900                                 ;
14000                                 ;READ-- A COMMON ROUTINE WHICH CHECKS THE KEYBOARD
14100                                 ;       DONE FLAG & SETS A FLAG INDICATING CHAR PARITY
14200                                 ;
14300                                 ;XXXXXXXXXX
14400
14500 004112  004767  177572          $READ:  JSR     PC,CONIT        ;RESET CONSOLE ADR AND VECTORS
14600 004116  005767  174534                  TST     DLCNT           ;CHECK IF MULTI DL11'S AVAILABLE
14700 004122  001430                          BEQ     $READC          ;NONE, WAIT FOR CONSOLE INPUT
14800 004124  016767  174526  174534  1$:     MOV     DLCNT,COUNT3    ;SET DL11 COUNT
14900 004132  016767  174474  174530          MOV     FSTDL,XCSR      ;ADDRESS OF FIRST DL11 INTO XCSR
15000 004140  105777  174524          2$:     TSTB    @XCSR           ;TEST IF ANY INPUT
15100 004144  100005                          BPL     3$              ;CONTINUE IF NO INPUT
15200 004146  016700  174516                  MOV     XCSR,R0         ;SET THIS DL11 AS CONSOLE
15300 004152  004767  177536                  JSR     PC,CONSET
15400 004156  000415                          BR      READ1           ;READ CHAR AND RETURN
15500 004160  005367  174502          3$:     DEC     COUNT3          ;DECREMENT DL11 COUNT
15600 004164  001404                          BEQ     4$              ;TEST CONSOLE WHEN DONE DL11'S
15700 004166  062767  000010  174474          ADD     #10,XCSR        ;NEXT DL11 ADDRESS
15800 004174  000761                          BR      2$              ;CONTINUE
15900 004176  105777  174410          4$:     TSTB    @TKS            ;CHECK CONSOLE
16000 004202  100350                          BPL     1$              ;WAIT, NO INPUT
16100 004204  105777  174402          $READC: TSTB    @TKS            ;CHECK KEYBOARD DONE FLAG
16200 004210  100375                          BPL     $READC          ;BRANCH IF NOT SET
16300 004212  117767  174376  174460  READ1:  MOVB    @TKB,TEMPCH     ;SAVE CHARACTER
16400 004220  116767  174454  174456          MOVB    TEMPCH,PCHAR    ;SAVE CODE WITH PARITY BIT
16500 004226  042767  177400  174450          BIC     #177400,PCHAR   ;MASK UNWANTED BITS
16600 004234  116767  174440  174441          MOVB    TEMPCH,PARITY+1 ;SAVE CHAR WITH PARITY BIT
16700 004242  042767  177600  174430          BIC     #177600,TEMPCH  ;MAKE IT 7 BIT ASCII
16800 004250  026727  174424  000004          CMP     TEMPCH,#4       ;DISREGARD EOT
16900 004256  001715                          BEQ     $READ
17000 004260  012700  000011                  MOV     #11,R0          ;SET SHIFT COUNT
17100 004264  042767  000377  174410          BIC     #377,PARITY     ;CLEAR PARITY FLAG
17200 004272  005300                  1$:     DEC     R0              ;DECREMENT SHIFT COUNT
17300 004274  001406                          BEQ     2$              ;EXIT IF DONE
17400 004276  106367  174401                  ASLB    PARITY+1        ;SHIFT CODE
17500 004302  103373                          BCC     1$              ;CONTINUE IF BIT WAS ZERO
17600 004304  105167  174372                  COMB    PARITY          ;CHANGE PARITY FLAG IF BIT WAS ONE
17700 004310  000770                          BR      1$              ;CONTINUE
17800 004312  000002                  2$:     RTI                     ;SET, RET. TO CALLER
17900
18000                                 ;XXXXXXXXXX
18100                                 ;
18200                                 ;PRINT-- A COMMON ROUTINE TO CHECK THE PRINTER READY FLAG
18300                                 ;
18400                                 ;XXXXXXXXXX
18500
18600 004314  105777  174276          $PRNT:  TSTB    @TPS            ;CHECK PRINTER READY FLAG
18700 004320  100375                          BPL     $PRNT           ;BRANCH IF NOT SET
18800 004322  000002                          RTI                     ;SET, RETURN
```

```
                                        ;**********
                                        ;
                                        ;PRINTC--SENDS A CHARACTER AT A TIME FIRST TO THE
                                        ;         CONSOLE  DL11 THEN TO ALL MULTIPLE DL11S IF
                                        ;         SR BIT 13 IS = 0. IF THE REFERENCED PRINTER
                                        ;         READY BIT IS NOT SET, THE CHARACTER WILL NOT BE
                                        ;         SENT TO THAT PRINTER. ENTER WITH CHARACTER IN RO.
                                        ;         CALL:   PRINTC
                                        ;
                                        ;********

004324  016767  174252  174360  $PRTC:  MOV     CONADD,TEMP         ;SET CONSOLE ADR
004332  062767  000004  174352          ADD     #4,TEMP
004340  105777  174346          1$:     TSTB    @TEMP
004344  100375                          BPL     1$                  ;WAIT FOR CONSOLE READY
004346  062767  000002  174336          ADD     #2,TEMP             ;SET ADR
004354  010077  174332                  MOV     R0,@TEMP            ;LOAD CONSOLE PRINTER BUFFER
004360  032777  020000  174326          BIT     #BIT13,@SR          ;CHECK SW 13
004366  001003                          BNE     2$                  ;SEND ALL TERMS IF SW13 DOWN
004370  005767  174262                  TST     DLCNT               ;CHECK IF MULTIPLE DL11'S
004374  001002                          BNE     3$                  ;CHECK FOR INPUT IF THERE
004376  000167  000432          2$:     JMP     18$
004402  016767  174250  174256  3$:     MOV     DLCNT,COUNT3        ;PUT NO. DL11'S INTO COUNT3
004410  016767  174216  174252          MOV     FSTDL,XCSR          ;ADDR OF FIRST DL INTO XCSR
004416  005767  174266          4$:     TST     INCHK               ;CHECK FOR INPUT?
004422  001140                          BNE     13$
004424  026727  174206  000020          CMP     RTNNO,#20           ;PRINTING TEST?
004432  002004                          BGE     5$                  ;BRANCH IF NOT
004434  022767  104011  175236          CMP     #TTYCTL,WAITF       ;KEYBOARD CONTROL?
004442  001130                          BNE     13$                 ;SKIP INPUT CHECK IF NOT
004444  105777  174220          5$:     TSTB    @XCSR               ;TEST IF ANY INPUT
004450  100125                          BPL     13$                 ;CONTINUE IF NO INPUT
004452  062767  000002  174210          ADD     #2,XCSR             ;SET BUFFER ADDRESS
004460  017767  174204  174212          MOV     @XCSR,TEMPCH
004466  042767  177600  174204          BIC     #177600,TEMPCH
004474  026727  174200  000003          CMP     TEMPCH,#3           ;CHECK IF CONTROL-C
004502  001006                          BNE     6$                  ;CONTINUE IF NOT
004504  026727  174126  000024          CMP     RTNNO,#24           ;CHECK IF TEST 24
004512  001002                          BNE     6$                  ;CONTINUE IF NOT CONTROL-C
004514  000167  000420                  JMP     20$
004520  026727  174154  000177  6$:     CMP     TEMPCH,#177         ;CHECK IF RUBOUT
004526  001427                          BEQ     9$                  ;YES, CHECK TEST NUMBER
004530  026727  174102  000017          CMP     RTNNO,#17           ;TEST 17?
004536  001003                          BNE     7$                  ;BRANCH IF NOT
004540  016703  174134                  MOV     TEMPCH,R3           ;SAVE CHARACTER
004544  000461                          BR      12$                 ;CONTINUE
004546  026727  174064  000021  7$:     CMP     RTNNO,#21           ;TEST 21?
004554  001004                          BNE     8$                  ;BRANCH IF NOT
004556  016767  174116  174076          MOV     TEMPCH,REPT         ;SAVE CHARACTER
004564  000451                          BR      12$                 ;CONTINUE
004566  026727  174044  000022  8$:     CMP     RTNNO,#22           ;TEST 22?
004574  001056                          BNE     14$                 ;CONTINUE IF NOT
004576  016767  174076  174056          MOV     TEMPCH,REPT         ;SAVE CHARACTER
004604  000441                          BR      12$                 ;CONTINUE
004606  026727  174024  000021  9$:     CMP     RTNNO,#21           ;CHECK IF TEST 21
004614  001011                          BNE     10$                 ;NO, CHECK IF TEST 22
004616  022626                          POPSP2                      ;ASJUST STACK
```

```
004620  012700  000036                  MOV     #30.,R0             ;DELAY FOR HALF DUPLEX
004624  104010                          DELAY
004626  104007                          TYPEM                       ;YES, TEST 21
004630  014272                          ECOEND                      ;PRINT TERMINATION MESSAGE
004632  104005                          CHAIN                       ;CHAIN TO NEXT TEST
004634  000167  005334                  JMP     E021A               ;REPEAT TEST IF LOOP ON TEST SW SET
004640  026727  173772  000022  10$:    CMP     RTNNO,#22           ;CHECK IF TEST 22
004646  001011                          BNE     11$                 ;NO, CHECK IF TEST 24
004650  022626                          POPSP2                      ;ADJUST STACK
004652  012700  000036                  MOV     #30.,R0             ;DELAY FOR HALF DUPLEX
004656  104010                          DELAY
004660  104007                          TYPEM                       ;YES, PRINT TERMINATION MESSAGE
004662  014272                          ECOEND
004664  104005                          CHAIN                       ;CHAIN TO NEXT TEST
004666  000167  005340                  JMP     E022A               ;REPEAT TEST IF LOOP ON TEST SW SET
004672  026727  173740  000024  11$:    CMP     RTNNO,#24           ;TEST 24?
004700  001133                          BNE     22$                 ;WAIT FOR NEXT TEST IF NOT TEST 24
004702  022626                          POPSP2                      ;RESET STACK
004704  000167  006244                  JMP     TERM                ;TERMINATE TEST
004710  012700  000036          12$:    MOV     #30.,R0             ;DELAY FOR HALF DUPLEX
004714  104010                          DELAY
004716  016700  173756                  MOV     TEMPCH,R0           ;SET NEW CHARACTER
004722  000403                          BR      14$                 ;CONTINUE
004724  062767  000002  173736  13$:    ADD     #2,XCSR             ;SET STATUS ADDRESS IN XCSR
004732  062767  000002  173730  14$:    ADD     #2,XCSR
004740  016767  173636  173744          MOV     CONADD,TEMP         ;CHECK IF CONSOLE TERMINAL
004746  062767  000004  173736          ADD     #4,TEMP             ;IS THIS DL
004754  026767  173732  173706          CMP     TEMP,XCSR
004762  001420                          BEQ     17$
004764  105777  173700          15$:    TSTB    @XCSR               ;TEST PRINTER READY
004770  100375                          BPL     15$                 ;WAIT FOR READY
004772  062767  000002  173670          ADD     #2,XCSR             ;SET XCSR TO PRINTER BUFFER
005000  010077  173664                  MOV     R0,@XCSR            ;LOAD CHARACTER INTO BUFFER
005004  005367  173656          16$:    DEC     COUNT3              ;DECREASE COUNT OF DL11'S
005010  001411                          BEQ     18$                 ;ALL DONE,EXIT
005012  062767  000002  173650          ADD     #2,XCSR             ;SET XCSR TO NEXT DL11 PRINTER STATUS
005020  000167  177372                  JMP     4$                  ;GO TEST NEXT DL11 READY FLAG
005024  062767  000002  173636  17$:    ADD     #2,XCSR             ;SET XCSR TO PRINTER BUFFER
005032  000764                          BR      16$                 ;DO NOT LOAD BUFFER
005034  005767  173650          18$:    TST     INCHK               ;WANT INPUT CHECK?
005040  001111                          BNE     26$                 ;NO, BRANCH
005042  026727  173570  000020          CMP     RTNNO,#20           ;PRINTING TEST?
005050  002004                          BGE     19$                 ;BRANCH IF NOT
005052  022767  104011  174620          CMP     #TTYCTL,WAITF       ;KEYBOARD CONTROL?
005060  001101                          BNE     26$                 ;SKIP INPUT CHECK IF NOT
005062  105777  173514          19$:    TSTB    @CONADD             ;TEST IF ANY INPUT
005066  100076                          BPL     26$                 ;BRANCH IF NONE
005070  016767  173506  173614          MOV     CONADD,TEMP         ;SET ADR
005076  062767  000002  173606          ADD     #2,TEMP
005104  117767  173602  173566          MOVB    @TEMP,TEMPCH
005112  042767  177600  173560          BIC     #177600,TEMPCH      ;MASK UNWANTED BITS
005120  026727  173554  000003          CMP     TEMPCH,#3           ;CHARACTER = CONTROL-C?
005126  001013                          BNE     21$                 ;CONTINUE IF NOT
005130  026727  173502  000024          CMP     RTNNO,#24           ;TEST 24?
005136  001007                          BNE     21$                 ;CONTINUE IF NOT
005140  012700  000036          20$:    MOV     #30.,R0             ;DELAY FOR HALF DUPLEX
005144  104010                          DELAY
```

```
 30400 005146  104012                              CRLF                    ;SEND CR-LF
 30500 005150  022626                              POPSP2                  ;RESET STACK
 30600 005152  000167  005650                      JMP     E024B           ;RETURN TO TEST
 30700 005156  026727  173516  000177     21$:     CMP     TEMPCH,#177     ;CHECK IF RUBOUT
 30800 005164  001006                              BNE     23$             ;BRANCH IF NO
 30900 005166  000607                              BR      9$
 31000 005170  012767  000001  173436     22$:     MOV     #1,CNTLSW       ;CLEAR LOOP AND SEQUENCE BITS
 31100 005176  000167  174660                      JMP     TTY1B           ;GO WAIT FOR NEXT TEST
 31200 005202  010046                     23$:     MOV     R0,-(SP)        ;SAVE R0
 31300 005204  012700  000036                      MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
 31400 005210  104010                              DELAY
 31500 005212  012600                              MOV     (SP)+,R0        ;RESTORE R0
 31600 005214  026727  173416  000017              CMP     RTNNO,#17       ;CHECK IF TEST 17
 31700 005222  001002                              BNE     24$             ;BRANCH IF NOT TEST 17
 31800 005224  016703  173450                      MOV     TEMPCH,R3       ;STORE INPUTTED CHARACTER
 31900 005230  026727  173402  000021     24$:     CMP     RTNNO,#21       ;CHECK IF TEST 21
 32000 005236  001003                              BNE     25$             ;BRANCH IF NOT TEST 21
 32100 005240  016767  173434  173414              MOV     TEMPCH,REPT     ;STORE INPUTTED CHARACTER
 32200 005246  026727  173364  000022     25$:     CMP     RTNNO,#22       ;CHECK IF TEST 22
 32300 005254  001003                              BNE     26$             ;BRANCH IF NOT TEST 22
 32400 005256  016767  173416  173376              MOV     TEMPCH,REPT     ;STORE INPUTTED CHARACTER
 32500 005264  000002                     26$:     RTI                     ;RETURN TO TEST
 32600
```

```
 32800
```

```
                                                .SBTTL  I/O LOGIC TESTS
                                        ;
                                        ;**********
                                        ;
                                        ;ONLY THE CONSOLE TERMINAL IS TESTED.
                                        ;UPON COMPLETION, THE CPU WILL EITHER HALT IF SR
                                        ;BIT8 IS = 1 AND AWAIT FUTHER INSTRUCTIONS OR CONTINUE
                                        ;AND EXECUTE THE PRINTER TESTS CONTINOUSLY
                                        ;IF AN I/O TEST FAILS, THE CPU WILL HALT AT ERRHLT
                                        ;WITH THE ADDRESS OF THE ERROR IN R0 (LOC 777700). PRESSING
                                        ;THE CONTINUE SWITCH WILL CAUSE THE I/O TEST TO
                                        ;CONTINUE WITH THE NEXT TEST. HOWEVER IF SWITCH 14
                                        ;WERE SET, OR IS SET BEFORE THE CONTINUE SWITCH IS
                                        ;PRESSED, THE FAILED TEST WILL LOOP ON ITSELF
                                        ;WITHOUT FURTHER HALTS
                                        ;
                                        ;**********
                                        ;AT0--  TEST #40--TESTS THE ABILITY TO REFERENCE THE
                                        ;       RECEIVER STATUS WORD (TKS) WITHOUT TRAPPING.
                                        ;**********

005266  000040                          AT0:    40                      ;TEST NUMBER
005270  005320                          AT0X:   AT1                     ;NEXT TEST
005272  000012                                  10.                     ;ITERATION COUNT
005274  005304                                  1$                      ;SCOPE ENTRY
005276  012767  005314  172500                  MOV     #3$,MACHER      ;SET UP MACHINE ERROR TRAP
005304  005777  173302                  1$:     TST     @TKS            ;REFERENCE RECEIVER STATUS WORD
005310  104005                          2$:     CHAIN                   ;CHAIN TO NEXT TEST
005312  000774                                  BR      1$              ;REPEAT TEST
005314  104001                          3$:     ERROR                   ;ERROR TRAPPED WHEN REFERENCING
005316  000774                                  BR      2$              ;RECEIVER STATUS WORD (TKS)

                                        ;**********
                                        ;AT1--TEST #41--TESTS THE ABILITY TO  REFERENCE THE
                                        ;       RECEIVER BUFFER (TKB) WITHOUT TRAPPING.
                                        ;**********

005320  000041                          AT1:    41                      ;TEST NUMBER
005322  005352                                  AT2                     ;NEXT TEST
005324  000012                                  10.                     ;ITERATION COUNT
005326  005336                                  1$                      ;SCOPE ENTRY
005330  012767  005346  172446                  MOV     #3$,MACHER      ;SET UP MACHINE ERROR TRAP
005336  005777  173252                  1$:     TST     @TKB            ;REFERENCE RECEIVER BUFFER
005342  104005                          2$:     CHAIN                   ;CHAIN TO NEXT TEST
005344  000774                                  BR      1$              ;REPEAT TEST
005346  104001                          3$:     ERROR                   ;TRAPPED WHEN REFERENCING
005350  000774                                  BR      2$              ;RECEIVER BUFFER (TKB)
```

```
                                        ;**********
                                        ;AT2--TEST #42--TESTS THE ABILITY TO REFERENCE THE
                                        ;       TRANSMITTER STATUS WORD (TPS) WITHOUT TRAPPING.
                                        ;**********

005352  000042                          AT2:    42                      ;TEST NUMBER
005354  005404                                  AT3                     ;NEXT TEST
005356  000012                                  10.                     ;ITERATION COUNT
005360  005370                                  1$                      ;SCOPE ENTRY
005362  012767  005400  172414                  MOV     #3$,MACHER      ;SET UP MACHINE ERROR TRAP
005370  005777  173222                  1$:     TST     @TPS            ;REFERENCE TRANSMITTER STATUS
005374  104005                          2$:     CHAIN                   ;CHAIN TO NEXT TEST
005376  000774                                  BR      1$              ;REPEAT TEST
005400  104001                          3$:     ERROR                   ;TRAPPED WHEN REFERENCING
005402  000774                                  BR      2$              ;TRANSMITTER STATUS WORD

                                        ;**********
                                        ;AT3--  TEST #43--TESTS THE ABILITY TO REFERENCE THE
                                        ;       TRANSMITTER BUFFER (TPB) WITHOUT TRAPPING.
                                        ;**********

005404  000043                          AT3:    43                      ;TEST NUMBER
005406  005436                                  AT4                     ;NEXT TEST
005410  000012                                  10.                     ;ITERATION COUNT
005412  005422                                  1$                      ;SCOPE ENTRY
005414  012767  005432  172362                  MOV     #3$,MACHER      ;SET UP ERROR TRAP
005422  005777  173172                  1$:     TST     @TPB            ;REFERENCE TRANSMITTER BUFFER
005426  104005                          2$:     CHAIN                   ;CHAIN TO NEXT TEST
005430  000774                                  BR      1$              ;REPEAT TEST
005432  104001                          3$:     ERROR                   ;TRAPPED WHEN REFERENCING
005434  000774                                  BR      2$              ;TRANSMITTER BUFFER.
```

```
                                        ;**********
                                        ;AT4--   TEST #44--TESTS THE ABILITY TO SET AND CLEAR THE
                                        ;        RECEIVER INTERRUPT ENABLE BIT.
                                        ;**********

005436  000044                          AT4:    44                              ;TEST NUMBER
005440  005526                                  AT5                             ;NEXT TEST
005442  000012                                  10.                             ;ITERATION COUNT
005444  005460                                  1$                              ;SCOPE ENTRY
005446  012746  000340                          MOV     #PRTY7,-(SP)            ;SET PRIORITY 7
005452  012746  005460                          MOV     #1$,-(SP)
005456  000002                                  RTI
005460  052777  000100  173124          1$:     BIS     #BIT6,@TKS              ;SET INTERRUPT ENABLE BIT
005466  032777  000100  173116                  BIT     #BIT6,@TKS              ;CHECK IF BIT IS SET
005474  001002                                  BNE     3$                      ;BRANCH IF SET
005476  104001                          2$:     ERROR                           ;NOT SET, ERROR
005500  000410                                  BR      5$                      ;CHAIN TO NEXT TEST
005502  042777  000100  173102          3$:     BIC     #BIT6,@TKS              ;CLEAR INTERRUPT ENABLE BIT
005510  032777  000100  173074                  BIT     #BIT6,@TKS              ;CHECK IF BIT IS CLEARED
005516  001401                                  BEQ     5$                      ;BRANCH IF CLEARED
005520  104001                          4$:     ERROR                           ;NOT CLEARED, ERROR
005522  104005                          5$:     CHAIN                           ;CHAIN TO NEXT TEST
005524  000755                                  BR      1$                      ;DO TEST AGAIN

                                        ;**********
                                        ;AT5--   TEST #45--CHECKS THAT THE RECEIVER INTERRUPT
                                        ;        ENABLE BIT CAN BE CLEARED WITH RESET INSTRUCTION.
                                        ;**********
                                        ;
005526  000045                          AT5:    45                              ;TEST NUMBER
005530  005604                                  AT6                             ;NEXT TEST
005532  000012                                  10.                             ;ITERATION COUNT
005534  005550                                  1$                              ;SCOPE ENTRY
005536  012746  000340                          MOV     #PRTY7,-(SP)            ;SET PRIORITY TO 7
005542  012746  005550                          MOV     #1$,-(SP)
005546  000002                                  RTI
005550  052777  000100  173034          1$:     BIS     #BIT6,@TKS              ;SET INTERRUPT ENABLE BIT
005556  105777  173034                  3$:     TSTB    @TPS                    ;BE SURE PRINTER IS DONE WITH DL11S1 MESSAGE
005562  001775                                  BEQ     3$                      ;BEFORE ALLOWING FOLLOWING RESET.
005564  000005                                  RESET                           ;RESET
005566  032777  000100  173016                  BIT     #BIT6,@TKS              ;TEST INTERRUPT ENABLE BIT
005574  001401                                  BEQ     2$                      ;BRANCH IF CLEARED
005576  104001                                  ERROR                           ;STILL SET,ERROR
005600  104005                          2$:     CHAIN                           ;CHAIN TO NEXT ROUTINE
005602  000762                                  BR      1$                      ;REPEAT TEST
```

```
                                        ;**********
                                        ;AT6--   TEST#46--TESTS THE ABILITY TO SET AND CLEAR
                                        ;        TRANSMITTER INTERRUPT ENABLE BIT.
                                        ;**********

005604  000046                          AT6:    46                              ;TEST NUMBER
005606  005674                                  AT7                             ;NEXT TEST
005610  000012                                  10.                             ;ITERATION COUNT
005612  005626                                  1$                              ;SCOPE ENTRY
005614  012746  000340                          MOV     #PRTY7,-(SP)            ;SET PRIORITY TO 7
005620  012746  005626                          MOV     #1$,-(SP)
005624  000002                                  RTI
005626  052777  000100  172762          1$:     BIS     #BIT6,@TPS              ;SET INTERRUPT ENABLE BIT
005634  032777  000100  172754                  BIT     #BIT6,@TPS              ;CHECK THAT BIT IS SET
005642  001002                                  BNE     2$                      ;BRANCH IF SET
005644  104001                                  ERROR                           ;NOT SET, ERROR
005646  000410                                  BR      3$                      ;CHAIN TO NEXT TEST
005650  042777  000100  172740          2$:     BIC     #BIT6,@TPS              ;CLEAR INTERRUPT ENABLE BIT
005656  032777  000100  172732                  BIT     #BIT6,@TPS              ;CHECK IF BIT IS CLEARED
005664  001401                                  BEQ     3$                      ;BRANCH IF CLEARED
005666  104001                                  ERROR                           ;NOT CLEARED, ERROR
005670  104005                          3$:     CHAIN                           ;CHAIN TO NEXT TEST
005672  000755                                  BR      1$                      ;DO AGAIN

                                        ;**********
                                        ;AT7--   TEST #47--TESTS THE ABILITY TO CLEAR TRANSMITTER
                                        ;        INTERRUPT ENABLE BIT WITH RESET INSTRUCTION.
                                        ;**********

005674  000047                          AT7:    47                              ;TEST NUMBER
005676  005744                                  AT10                            ;NEXT TEST
005700  000012                                  10.                             ;ITERATION COUNT
005702  005716                                  1$                              ;SCOPE ENTRY
005704  012746  000340                          MOV     #PRTY7,-(SP)            ;SET PRIORITY TO 7
005710  012746  005716                          MOV     #1$,-(SP)
005714  000002                                  RTI
005716  052777  000100  172672          1$:     BIS     #BIT6,@TPS              ;SET INTERRUPT BIT
005724  000005                                  RESET                           ;RESET
005726  032777  000100  172662                  BIT     #BIT6,@TPS              ;CHECK IF BIT IS CLEARED
005734  001401                                  BEQ     2$                      ;BRANCH IF CLEARED
005736  104001                                  ERROR                           ;ERROR, RESET DID NOT CLEAR BIT
005740  104005                          2$:     CHAIN                           ;CHAIN TO NEXT ROUTINE
005742  000765                                  BR      1$                      ;REPEAT TEST
```

```
                                                ;**********
                                                ;AT10-- TEST #50--CHECKS THAT RESET SETS THE TRANSMITTER
                                                ;       READY BIT AND THAT THE READY BIT CAN BE READ RELIABLY.
                                                ;**********

005744  000050                          AT10:   50                      ;TEST NUMBER
005746  006002                                  AT11                    ;NEXT TEST
005750  000012                                  10.                     ;ITERATION COUNT
005752  005754                                  1$                      ;SCOPE ENTRY
005754  032777  001000  172732  1$:     BIT     #LSI11,@SR      ;SKIP TEST IF AN LSI-11
005762  001005                                  BNE     2$
005764  000005                                  RESET                   ;RESET
005766  105777  172624                          TSTB    @TPS            ;CHECK TRANSMIT READY BIT
005772  100401                                  BMI     2$              ;BRANCH IF SET
005774  104001                                  ERROR                   ;ERROR, RESET DID NOT SET READY BIT
005776  104005                          2$:     CHAIN                   ;CHAIN TO NEXT TEST
006000  000765                                  BR      1$              ;DO AGAIN

                                                ;**********
                                                ;AT11-- TEST #51--TESTS THAT THE TRANSMITTER READY RESETS
                                                ;       BY LOADING THE TRANSMITTER BUFFER.
                                                ;**********

006002  000051                          AT11:   51                      ;TEST NUMBER
006004  006042                                  AT12                    ;NEXT TEST
006006  000012                                  10.                     ;ITERATION COUNT
006010  006012                                  1$                      ;SCOPE ENTRY
006012  012700  000226                  1$:     MOV     #226,R0
006016  104010                                  DELAY                   ;DELAY 150 MSEC.
006020  000005                                  RESET                   ;RESET
006022  005077  172572                          CLR     @TPB            ;LOAD TRANSMITTER BUFFER
006026  105777  172564                          TSTB    @TPS            ;CHECK TRANSMIT READY BIT
006032  100001                                  BPL     2$              ;BRANCH IF CLEARED
006034  104001                                  ERROR                   ;NOT CLEARED, ERROR
006036  104005                          2$:     CHAIN                   ;CHAIN TO NEXT TEST
006040  000764                                  BR      1$              ;REPEAT TEST
```

```
                                                ;**********
                                                ;AT12-- TEST #52--CHECKS THAT THE TRANSMIT READY BIT CAN
                                                ;       CAUSE AN INTERRUPT
                                                ;**********

006042  000052                          AT12:   52                      ;TEST NUMBER
006044  006116                                  AT13                    ;NEXT TEST
006046  000012                                  10.                     ;ITERATION COUNT
006050  006056                                  1$                      ;SCOPE ENTRY
006052  104004                                  STPCHV                  ;SET UP TRANSMITTER INTERRUPT VECTOR
006054  006112                                  4$                      ;TO 4$
006056  000005                          1$:     RESET                   ;SEE CHAINY COMMENT
006060  005077  172532                          CLR     @TPS            ;DISABLE TRANSMIT INTERRUPT
006064  005046                                  CLR     -(SP)           ;SET PRIORITY TO ZERO
006066  012746  006074                          MOV     #2$,-(SP)
006072  000002                                  RTI
006074  052777  000100  172514  2$:     BIS     #BIT6,@TPS      ;ENABLE TRANSMIT INTERRUPT
006102  000240                                  NOP
006104  104001                                  ERROR                   ;TRANSMIT READY DID NOT CAUSE INTERRUPT
006106  104005                          3$:     CHAIN                   ;CHAIN TO NEXT TEST
006110  000762                                  BR      1$              ;REPEAT TEST
006112  022626                          4$:     POPSP2                  ;INTERRUPT OCCURRED, CLEAN STACK
006114  000774                                  BR      3$              ;CHAIN TO NEXT TEST

                                                ;**********
                                                ;AT13-- TEST#53--TESTS THAT THE TRANSMIT READY DOES NOT CAUSE AN
                                                ;       INTERRUPT WHEN THE PROCESSOR IS AT THE SAME LEVEL
                                                ;**********

006116  000035                          AT13:   35                      ;TEST NUMBER
006120  006176                                  AT14                    ;NEXT TEST
006122  000012                                  10.                     ;ITERATION COUNT
006124  006132                                  1$                      ;SCOPE ENTRY
006126  104004                                  STPCHV                  ;SET UP TRANSMIT INTERRUPT
006130  006170                                  4$                      ;VECTOR TO 4$
006132  016746  172472                  1$:     MOV     TPLVL,-(SP)     ;SET PROCESSOR TO SAME LEVEL AS XMITTER
006136  012746  006144                          MOV     #2$,-(SP)
006142  000002                                  RTI
006144  005077  172446                  2$:     CLR     @TPS            ;DISABLE TRANSMITTER INTERRUPTS
006150  052777  000100  172440                  BIS     #BIT6,@TPS      ;ENABLE TRANSMITTER INTERRUPTS
006156  000240                                  NOP
006160  005077  172432                  3$:     CLR     @TPS            ;OK, NO INTERRUPT OCCURRED
006164  104005                                  CHAIN                   ;CHAIN TO NEXT TEST
006166  000761                                  BR      1$              ;REPEAT TEST
006170  022626                          4$:     POPSP2                  ;INTERRUPT OCCURRED,ERROR,CLEAN
006172  104001                                  ERROR                   ;UP STACK
006174  000771                                  BR      3$              ;CHAIN TO NEXT TEST
```

```
                                      ;**********
                                      ;AT14--  TEST#54--TESTS THAT THE TRANSMIT READY DOES CAUSE AN
                                      ;        INTERRUPT WHEN THE PROCESSOR IS AT A PRIORITY LEVEL
                                      ;        ONE LOWER THAN THE TRANSMIT INTERRUPT REQUEST LEVEL
                                      ;**********

006176  000054                        AT14:   54                      ;TEST NUMBER
006200  006262                                AT15                    ;NEXT TEST
006202  000012                                10.                     ;ITERATION COUNT
006204  006212                                1$                      ;SCOPE ENTRY
006206  104004                                STPCHV                  ;SET UP TRANSMIT INTERRUPT
006210  006250                                3$                      ;VECTOR TO 3$
006212  005077  172400                1$:     CLR     @TPS            ;DISABLE TRANSMIT INTERRUPTS
006216  016746  172406                        MOV     TPLVL,-(SP)     ;SET PROCESSOR PRIORITY ONE
006222  162716  000040                        SUB     #40,(SP)        ;LEVEL LOWER THAN TRANSMITTER
006226  012746  006234                        MOV     #2$,-(SP)
006232  000002                                RTI
006234  052777  000100  172354        2$:     BIS     #BIT6,@TPS      ;ENABLE TRANSMITTER INTERRUPTS
006242  000240                                NOP
006244  104001                                ERROR                   ;NO INTERRUPT, ERROR
006246  000401                                BR      4$              ;CHAIN TO NEXT TEST
006250  022626                        3$:     POPSP2                  ;INTERRUPT OCCURED, OK, CLEAN STACK
006252  005077  172340                4$:     CLR     @TPS            ;DISABLE TRANSMITTER INTERRUPTS
006256  104005                                CHAIN                   ;CHAIN TO NEXT TEST
006260  000754                                BR      1$              ;REPEAT TEST
```

```
                                      ;**********
                                      ;AT15--  TEST#55--TESTS THAT THE TRANSMIT READY DOES NOT
                                      ;        REINTERRUPT AFTER AN RTI WHEN THE READY BIT HAS
                                      ;        NOT BEEN RESET.
                                      ;**********

006262  000055                        AT15:   55                      ;TEST NUMBER
006264  006362                                AT16                    ;NEXT TEST
006266  000012                                10.                     ;ITERATION COUNT
006270  006272                                1$                      ;SCOPE ENTRY
006272  104004                        1$:     STPCHV                  ;SET TRANSMIT INTERRUPT VECTOR
006274  006334                                4$                      ;TO 4$
006276  005077  172314                        CLR     @TPS            ;DISABLE TRANSMITTER INTERRUPTS
006302  005046                                CLR     -(SP)           ;SET PROCESSOR PRIORITY TO ZERO
006304  012746  006312                        MOV     #2$,-(SP)
006310  000002                                RTI
006312  052777  000100  172276        2$:     BIS     #BIT6,@TPS      ;ENABLE TRANSMITTER INTERRUPTS
006320  000240                                NOP
006322  104001                                ERROR                   ;ERROR1, TRANSMITTER FAILED TO INTERRUPT
006324  005077  172266                3$:     CLR     @TPS            ;DISABLE TRANSMITTER INTERRUPTS
006330  104005                                CHAIN                   ;CHAIN TO NEXT TEST
006332  000757                                BR      1$              ;REPEAT TEST
006334  012777  006354  172264        4$:     MOV     #6$,@TPVTR      ;INTERRUPT OCCURRED, CHANGE INTERRUPT
006342  012716  006350                        MOV     #5$,@SP         ;VECTOR TO 6$ AND RETURN TO 5$
006346  000002                                RTI                     ;RETURN FROM INTERRUPT
006350  000240                        5$:     NOP
006352  000764                                BR      3$              ;CHAIN TO NEXT TEST
006354  022626                        6$:     POPSP2                  ;ERROR2, TRANSMITTER REINTERRUPTED
006356  104001                                ERROR                   ;AFTER RTI WITH READY BIT LEFT ON.
006360  000761                                BR      3$              ;CLEAN STACK, CHAIN TO NEXT TEST.

                                      ;**********
                                      ;AT16--TEST#56--CHECKS THAT RESET CLEARS THE RECEIVER DONE BIT
                                      ;**********

006362  000056                        AT16:   56                      ;TEST NUMBER
006364  006430                                AT17                    ;NEXT TEST
006366  000012                                10.                     ;ITERATION COUNT
006370  006372                                1$                      ;SCOPE ENTRY
006372  032777  001000  172314        1$:     BIT     #LSI11,@SR      ;SKIP TEST IF LSI-11
006400  001011                                BNE     3$
006402  012700  000226                        MOV     #226,R0
006406  104010                                DELAY                   ;DELAY 150 MSEC.
006410  104021                        2$:     AREAD                   ;ENABLE RECEIVER
006412  000005                                RESET                   ;RESET
006414  105777  172172                        TSTB    @TKS            ;TEST DONE BIT
006420  100001                                BPL     3$              ;BRANCH IF DONE IS CLEARED
006422  104001                                ERROR                   ;NOT CLEARED, ERROR
006424  104005                        3$:     CHAIN                   ;CHAIN TO NEXT TEST
006426  000761                                BR      1$              ;REPEAT TEST
```

```
  33300                                             ;**********
  33400                                             ;AT17-- TEST#57--CHECKS THAT REFERENCING THE RECEIVER BUFFER
  33500                                             ;       CLEARS THE DONE BIT.
  33600                                             ;**********
  33700
  33800 006430  000057                              AT17:   57                              ;TEST NUMBER
  33900 006432  006500                                      AT20                            ;NEXT TEST
  34000 006434  000012                                      10.                             ;ITERATION COUNT
  34100 006436  006440                                      1$                              ;SCOPE ENTRY
  34200 006440  032777  001000  172246  1$:     BIT     #LSI11,@SR              ;CHECK FOR LSI-11
  34300 006446  001012                                      BNE     3$                      ;SKIP TEST IF SET
  34400 006450  012700  000226                              MOV     #226,R0
  34500 006454  104010                                      DELAY                           ;DELAY 150 MSEC.
  34600 006456  104021                              2$:     AREAD                           ;ENABLE RECEIVER
  34700 006460  105777  172130                              TSTB    @TKB                    ;REFERENCE RECEIVER BUFFER
  34800 006464  105777  172122                              TSTB    @TKS                    ;TEST DONE BIT
  34900 006470  100001                                      BPL     3$                      ;BRANCH IF NOT SET
  35000 006472  104001                                      ERROR                           ;DONE BIT IS SET, ERROR
  35100 006474  104005                              3$:     CHAIN                           ;CHAIN TO NEXT TEST
  35200 006476  000760                                      BR      1$                      ;REPEAT TEST
  35300
  35400                                             ;**********
  35500                                             ;AT20-- TEST#60--CHECK THAT THE RECEIVER DONE BIT IS ABLE TO
  35600                                             ;       CAUSE AN INTERRUPT.
  35700                                             ;**********
  35800
  35900 006500  000060                              AT20:   60                              ;TEST NUMBER
  36000 006502  006572                                      AT21                            ;NEXT TEST
  36100 006504  000012                                      10.                             ;ITERATION COUNT
  36200 006506  006514                                      1$                              ;SCOPE ENTRY
  36300 006510  104003                                      STRDRV                          ;SET UP RECEIVER INTERRUPT
  36400 006512  006564                                      4$                              ;VECTOR TO 4$
  36500 006514  032777  001000  172172  1$:     BIT     #LSI11,@SR              ;CHECK FOR LSI-11
  36600 006522  001021                                      BNE     5$                      ;SKIP TEST IF SET
  36700 006524  012700  000226                              MOV     #226,R0
  36800 006530  104010                                      DELAY                           ;DELAY 150 MSEC
  36900 006532  104021                                      AREAD                           ;ENABLE RECEIVER
  37000 006534  005077  172052                      2$:     CLR     @TKS                    ;DISABLE RECEIVER INTERRUPTS
  37100 006540  005046                                      CLR     -(SP)                   ;SET PROCESS STATUS TO ZERO
  37200 006542  012746  006550                              MOV     #3$,-(SP)
  37300 006546  000002                                      RTI
  37400 006550  052777  000100  172034  3$:     BIS     #BIT6,@TKS              ;ENABLE RECEIVER INTERRUPT
  37500 006556  000240                                      NOP
  37600 006560  104001                                      ERROR                           ;ERROR,RECEIVER FAILED TO INTERRUPT
  37700 006562  000401                                      BR      5$                      ;CHAIN TO NEXT TEST
  37800 006564  022626                              4$:     POPSP2                          ;OK, CLEAN STACK
  37900 006566  104005                              5$:     CHAIN                           ;CHAIN TO NEXT TEST
  38000 006570  000751                                      BR      1$                      ;REPEAT TEST
```

```
  38200                                             ;**********
  38300                                             ;AT21-- TEST#61--TESTS THAT THE RECEIVER DONE DOES NOT CAUSE AN
  38400                                             ;       INTERRUPT WHEN THE PROCESSOR IS AT THE SAME LEVEL AS
  38500                                             ;       THE RECEIVER'S INTERRUPT REQUEST LEVEL.
  38600                                             ;**********
  38700
  38800 006572  000061                              AT21:   61                              ;TEST NUMBER
  38900 006574  006672                                      AT22                            ;NEXT TEST
  39000 006576  000012                                      10.                             ;ITERATION COUNT
  39100 006600  006606                                      1$                              ;SCOPE ENTRY
  39200 006602  104003                                      STRDRV                          ;SET RECEIVER VECTOR TO 5$
  39300 006604  006664                                      5$
  39400 006606  032777  001000  172100  1$:     BIT     #LSI11,@SR              ;CHECK FOR LSI-11
  39500 006614  001017                                      BNE     4$                      ;SKIP TEST IF SET
  39600 006616  012700  000226                              MOV     #226,R0
  39700 006622  104010                                      DELAY                           ;DELAY 150 MSEC
  39800 006624  104021                                      AREAD                           ;ENABLE RECEIVER
  39900 006626  005077  171760                      2$:     CLR     @TKS                    ;DISABLE RECEIVER INTERRUPTS
  40000 006632  016746  171766                              MOV     TKLVL,-(SP)             ;SET PROCESSOR PRIORITY TO SAME LEVEL AS RECEIVER
  40100 006636  012746  006644                              MOV     #3$,-(SP)
  40200 006642  000002                                      RTI
  40300 006644  052777  000100  171740  3$:     BIS     #BIT6,@TKS              ;ENABLE RECEIVER INTERRUPTS
  40400 006652  000240                                      NOP
  40500 006654  005077  171732                      4$:     CLR     @TKS                    ;OK, NO INTERRUPT OCCURRED
  40600 006660  104005                                      CHAIN                           ;CHAIN TO NEXT TEST
  40700 006662  000751                                      BR      1$                      ;REPEAT TEST
  40800 006664  022626                              5$:     POPSP2                          ;ERROR, RECEIVER INTERRUPTED, CLEAN STACK
  40900 006666  104001                                      ERROR
  41000 006670  000771                                      BR      4$                      ;BRANCH 4$
```

```
                                          ;**********
                                          ;AT22--  TEST#62--TESTS THAT THE RECEIVER DONE DOES CAUSE AN
                                          ;        INTERRUPT WHEN THE PROCESSOR IS AT A PRIORITY ONE
                                          ;        LEVEL LOWER THAN THE RECEIVER'S INTERRUPT
                                          ;        REQUEST LEVEL
                                          ;**********

006672  000062                            AT22:   62                      ;TEST NUMBER
006674  007000                                    AT23                    ;NEXT TEST
006676  000012                                    10.                     ;ITERATION COUNT
006700  006706                                    1$                      ;SCOPE ENTRY
006702  104003                                    STRDRV                  ;SET RECEIVER INTERRUPT
006704  006766                                    4$                      ;VECTOR TO 4$
006706  032777  001000  172000            1$:     BIT     #LSI11,@SR      ;CHECK FOR LSI11
006714  001025                                    BNE     5$              ;SKIP TEST IF SET
006716  012700  000226                            MOV     #226,R0
006722  104010                                    DELAY                   ;DELAY 150 MSEC
006724  104021                                    AREAD                   ;ENABLE RECEIVER
006726  005077  171660                    2$:     CLR     @TKS            ;DISABLE READER INTERRUPTS
006732  016746  171666                            MOV     TKLVL,-(SP)     ;SET PROCESSOR PRIORITY ONE LEVEL
006736  012746  006744                            MOV     #3$,-(SP)
006742  000002                                    RTI
006744  162767  000040  171024            3$:     SUB     #40,PSW         ;LOWER THAN READER
006752  052777  000100  171632                    BIS     #BIT6,@TKS      ;ENABLE INTERRUPTS
006760  000240                                    NOP
006762  104001                                    ERROR                   ;FAILED TO INTERRUPT
006764  000401                                    BR      5$              ;CHAIN TO NEXT TEST
006766  022626                            4$:     POPSP2                  ;OK, CLEAN STACK
006770  005077  171616                    5$:     CLR     @TKS            ;DISABLE RECEIVER INTERRUPTS
006774  104005                                    CHAIN                   ;CHAIN TO NEXT TEST
006776  000743                                    BR      1$              ;REPEAT TEST
```

```
                                          ;**********
                                          ;AT23--  TEST#63--CHECKS THAT THE RECEIVER DONE DOES NOT
                                          ;        REINTERRUPT AFTER RTI INSTRUCTION WHEN DONE
                                          ;        BIT IS LEFT SET.
                                          ;**********

007000  000063                            AT23:   63                      ;TEST NUMBER
007002  007112                                    AT24                    ;NEXT TEST
007004  000012                                    10.                     ;ITERATION COUNT
007006  007010                                    1$                      ;SCOPE ENTRY
007010  032777  001000  171676            1$:     BIT     #LSI11,@SR      ;CHECK FOR LSI-11
007016  001015                                    BNE     3$              ;SKIP TEST IF SET
007020  012700  000226                    2$:     MOV     #226,R0
007024  104010                                    DELAY                   ;DELAY 150 MSEC
007026  104021                                    AREAD                   ;ENABLE RECEIVER
007030  104003                                    STRDRV                  ;SET RECEIVER INTERRUPT
007032  007064                                    4$                      ;VECTOR TO 4$
007034  005077  171552                            CLR     @TKS            ;DISABLE RECEIVER INTERRUPTS
007040  052777  000100  171544                    BIS     #BIT6,@TKS      ;ENABLE RECEIVER INTERRUPT
007046  000240                                    NOP                     ;
007050  104001                                    ERROR                   ;NO INTERRUPT, ERROR
007052  005077  171534                    3$:     CLR     @TKS            ;DISABLE RECEIVER INTERRUPTS
007056  000005                                    RESET                   ;RESET AFTER LAST INTERRUPT
007060  104005                                    CHAIN                   ;CHAIN TO NEXT TEST
007062  000752                                    BR      1$              ;REPEAT TEST
007064  012777  007104  171530            4$:     MOV     #6$,@TKVTR      ;INTERRUPT, OK, CHANGE VECTOR TO 6$
007072  012716  007100                            MOV     #5$,@SP         ;CHANGE RET ADDR TO 5$
007076  000002                                    RTI                     ;RETURN
007100  000240                            5$:     NOP
007102  000763                                    BR      3$              ;OK, NO ADDITIONAL INTERRUPT
007104  022626                            6$:     POPSP2                  ;ERROR, ADDITIONAL INTERRUPT
007106  104001                                    ERROR
007110  000760                                    BR      3$              ;CHAIN TO NEXT TEST
```

```
                                      ;**********
                                      ;AT24--TEST#64--HAVE OPERATOR TYPE A CHARACTER ON THE
                                      ;               KEYBOARD, THEN CHECK FOR RECEIVER DONE.
                                      ;
                                      ;               ALLOW 12 SECONDS FOR OPERATOR RESPONSE.
                                      ;**********

007112  000064                        AT24:   64                          ;TEST NUMBER
007114  007212                                AT25                        ;NEXT TEST
007116  000001                                1                           ;ITERATION COUNT
007120  007122                                1$                          ;SCOPE ENTRY
007122  032777  001000  171564        1$:     BIT     #LSI11,@SR          ;SKIP TEST IF NOT AN LSI-11
007130  001426                                BEQ     5$
007132  005777  171454                        TST     @TKS                ;SHOULD BE CLEAR
007136  001401                                BEQ     2$
007140  104001                                ERROR                       ;RECEIVER STATUS NOT =0
007142  012700  000600                2$:     MOV     #600,R0             ;1/2 SEC DELAY
007146  012767  000030  171542                MOV     #30,CNTR            ;SET UP FOR 12 SEC WAIT
007154  104000                                TYPE
007156  014401                                OPMSG                       ;MESSAGE TO TYPE A CHARACTER
007160  104010                        3$:     DELAY                       ;1/2 SECOND
007162  105777  171424                        TSTB    @TKS                ;CHECK DONE BIT
007166  100407                                BMI     5$                  ;SET - EXIT LOOP
007170  005367  171522                        DEC     CNTR
007174  001403                                BEQ     4$                  ;TIME HAS RUN OUT...
007176  012700  000600                        MOV     #600,R0             ;ANOTHER 1/2 SEC
007202  000766                                BR      3$                  ;CONTINUE WAIT
007204  104001                        4$:     ERROR                       ;NO RECEIVER DONE, OR
                                                                          ;OPERATOR DID NOT RESPOND
007206  104005                        5$:     CHAIN                       ;CHAIN TO NEXT TEST
007210  000744                                BR      1$
```

```
                                      ;**********
                                      ;AT25--TEST#65--CHECK THAT RECEIVER DONE CAUSES AN INTERRUPT
                                      ;               WHEN BIT 6 (INTERRUPT ENABLE) IS SET.
                                      ;**********

007212  000065                        AT25:   65                          ;TEST NUMBER
007214  007270                                AT26                        ;NEXT TEST
007216  000001                                1                           ;ITERATION COUNT
007220  007222                                1$                          ;SCOPE ENTRY
007222  032777  001000  171464        1$:     BIT     #LSI11,@SR          ;SKIP TEST IF NOT AN LSI-11
007230  001415                                BEQ     6$
007232  105777  171354                2$:     TSTB    @TKS                ;DONE SHOULD BE SET
007236  001001                                BNE     3$
007240  104001                                ERROR                       ;RECEIVER DONE NOT SET
007242  104003                        3$:     STRDRV                      ;SET RECEIVER INTERRUPT
007244  007262                                5$                          ;VECTOR TO 5$
007246  052777  000100  171336                BIS     #BIT6,@TKS          ;ENABLE INTERRUPT
007254  000240                                NOP
007256  000240                                NOP
007260  104001                        4$:     ERROR                       ;RECEIVER DID NOT INTERRUPT
007262  022626                        5$:     POPSP2                      ;CLEAN UP THE STACK
007264  104005                        6$:     CHAIN                       ;CHAIN TO NEXT TEST

007266  000755                                BR      1$
```

```
  53500                                          ;**********
  53600                                          ;AT26--TEST#66--CHECK THAT READING TKB CLEARS DONE BIT
  53700                                          ;               AND THAT DONE CLEARED DOES NOT CAUSE AN INTERRUPT
  53800                                          ;**********
  53900
  54000 007270  000066                           AT26:   66                              ;TEST NUMBER
  54100 007272  177777                                   -1                              ;LAST TEST
  54200 007274  000001                                   1                               ;ITERATION COUNT
  54300 007276  007300                                   1$                              ;SCOPE ENTRY
  54400 007300  032777  001000  171406   1$:     BIT     #LSI11,@SR      ;SKIP TEST IF NOT AN LSI-11
  54500 007306  001422                                   BEQ     5$
  54600 007310  105777  171276           2$:     TSTB    @TKS            ;MAKE SURE DONE IS STILL SET
  54700 007314  001001                                   BNE     3$
  54800 007316  104001                                   ERROR                           ;RECEIVER DONE NOT SET
  54900 007320  017767  171270  171370   3$:     MOV     @TKB,CNTR       ;READ DATA BUFFER
  55000 007326  105777  171260                   TSTB    @TKS            ;CHECK THE DONE BIT
  55100 007332  100001                                   BPL     4$              ;OK
  55200 007334  104001                                   ERROR                           ;READING DATA BUFFER DID NOT CLEAR DONE
  55300 007336  104003                   4$:     STRDRV                          ;SET RECEIVER INTERRUPT
  55400 007340  007364                                   6$                              ;VECTOR TO 6$
  55500 007342  052777  000100  171242           BIS     #BIT6,@TKS      ;ENABLE INTERRUPT
  55600 007350  000240                                   NOP
  55700 007352  000240                                   NOP
  55800 007354  005077  171232           5$:     CLR     @TKS            ;OK- CLEAN UP
  55900 007360  104005                                   CHAIN                           ;EXIT TESTS
  56000 007362  000746                                   BR      1$
  56100 007364  104001                   6$:     ERROR                           ;DLV INTERRUPTED WITH DONE CLEAR
  56200 007366  022626                                   POPSP2                          ;CLEAN UP THE STACK
  56300 007370  000771                                   BR      5$              ;EXIT TESTS
```

```
  56500
```

```
                                          .SBTTL  LA36 PRINTER TESTS
                            ;
                            ;THE LA36 PRINTER TESTS WILL BE EXECUTED IN A
                            ;CONTINOUS LOOP OUTPUTTING TO ALL MULTIPLE DL11'S
                            ;IF SR BIT 8 IS SET TO ZERO AT START UP TIME.  IF
                            ;BIT 8 IS SET TO 1 AT START UP THEY MAY BE EXECUTED
                            ;INDIVIDUALLY ONCE OR CONTINUALLY LOOPED, OR
                            ;BECOME THE FIRST OF THE ENTIRE SEQUENCE OF PRINTER
                            ;TESTS.  REFERENCE INTRUCTIONS IN THE INTRODUCTION
                            ;FOR PROPER MODE OF OPERATION.
                            ;
                            ;
                            ;XXXXXXXXXX
                            ;
                            ;PT0 -- DATA PATH TEST---FOUR LINES OF ALTERNATING
                            ;       "*" AND "U" ARE PRINTED, OUT TO THE GIVEN PAPER
                            ;       WIDTH.  THE PATTERN WILL APPEAR AS FOLLOWS.
                            ;
                            ;       *U*U*U*U*U*U
                            ;       U*U*U*U*U*U*
                            ;       *U*U*U*U*U*U
                            ;       U*U*U*U*U*U*
                            ;
                            ;XXXXXXXXXX
                            ;
007372  000000              PT0:    0                         ;TEST NUMBER
007374  007446                      PT1                       ;NEXT TEST
007376  104016                      PRTHDR                    ;
007400  104007                      TYPEM                     ;PRINT COLUMN # MESG
007402  014127                      HDR0
007404  012703  025125      1$:     MOV     #"U*,R3           ;SET FIRST CHAR PAIR
007410  012702  000004              MOV     #4,R2             ;SET LINE COUNT
007414  010300              2$:     MOV     R3,R0             ;SET CHAR PAIR
007416  016701  171230              MOV     WIDTH,R1          ;SET COLUMN COUNT
007422  104015              3$:     PRINTC                    ;PRINT CHAR
007424  000300                      SWAB    R0                ;SET NEXT CHAR
007426  005301                      DEC     R1                ;DEC COLUMN COUNT
007430  001374                      BNE     3$                ;FINISH LINE
007432  000303                      SWAB    R3                ;SET NEXT LINE START CHAR
007434  104012                      CRLF                      ;SEND CR-LF
007436  005302                      DEC     R2                ;DEC LINE COUNT
007440  001365                      BNE     2$                ;FINISH TEST
007442  104005                      CHAIN                     ;ALL DONE, EXIT
007444  000757                      BR      1$                ;REPEAT TEST
```

```
                            ;XXXXXXXXXX
                            ;
                            ;PT1 -- PRINTER CHARACTER TEST --- PRINTS ALL PRINTABLE CHARACTERS
                            ;
                            ;XXXXXXXXXX

007446  000001              PT1:    1                         ;TEST NUMBER
007450  007570                      PT2                       ;NEXT TEST
007452  104016                      PRTHDR
007454  012701  000040      1$:     MOV     #40,R1            ;SPACE TO R1
007460  012702  000100              MOV     #100,R2           ;@ TO R2
007464  012703  000140              MOV     #140,R3           ;\ TO R3
007470  110100              2$:     MOVB    R1,R0             ;CHAR TO R0
007472  004767  000042              JSR     PC,SPSP           ;SEND TWO SPACES
007476  110200                      MOVB    R2,R0             ;NEXT CHAR TO R0
007500  004767  000034              JSR     PC,SPSP           ;SEND TWO SPACES
007504  012704  000003              MOV     #3,R4             ;PRINT COUNT TO R4
007510  110300                      MOVB    R3,R0             ;THIRD CHAR TO R0
007512  104015              3$:     PRINTC                    ;PRINT THE CHAR
007514  005304                      DEC     R4                ;THREE TIMES ?
007516  001375                      BNE     3$                ;BRANCH IF NOT
007520  104012                      CRLF                      ;CARRIAGE RETURN LINE FEED
007522  122122                      CMPB    (R1)+,(R2)+       ;NEXT CHARACTERS
007524  105723                      TSTB    (R3)+
007526  020327  000200              CMP     R3,#200           ;CHECK IF ALL DONE
007532  103756                      BLO     2$                ;BRANCH IF NOT
007534  104005                      CHAIN                     ;EXIT TO NEXT TEST
007536  000746                      BR      1$                ;REPEAT TEST
007540  012704  000003      SPSP:   MOV     #3,R4             ;PRINT COUNT TO R4
007544  104015              1$:     PRINTC                    ;PRINT CHAR
007546  005304                      DEC     R4                ;THREE TIMES?
007550  001375                      BNE     1$                ;BRANCH IF NOT
007552  012700  000040      SP2:    MOV     #40,R0            ;SPACE TO R0
007556  104015                      PRINTC                    ;SEND A SPACE
007560  012700  000040      SPC:    MOV     #40,R0            ;SPACE TO R0
007564  104015                      PRINTC                    ;SEND ANOTHER
007566  000207                      RTS     PC                ;RETURN
```

```
                                        ;XXXXXXXXXX
                                        ;
                                        ;PT2 -- NON-PRINTING CHARACTER TEST.  THIS TEST
                                        ;       PRINTS THE OCTAL CODE FOLLOWED BY THE MNEMONIC
                                        ;       OF ALL NON-PRINTING CHARACTERS.  FOLLOWING EACH
                                        ;       MNEMONIC, THE PRINTER IS DRIVEN BY THE NON-PRINTING
                                        ;       CODE (000 THROUGH 037 PLUS 177)
                                        ;       ALL CONTROL CHARACTERS (INCLUDING THOSE FOR OPTIONS
                                        ;       WILL BE SKIPPED, REFER TO THE DOCUMENT FOR A LIST OF THOSE
                                        ;       TESTED.
                                        ;
                                        ;XXXXXXXXXX
                                        ;
007570  000002                          PT2:    2                               ;TEST NUMBER
007572  010164                                  PT3                             ;NEXT TEST
007574  104016                                  PRTHDR                          ;PRINT TEST HEADER
007576  012701  007676                  1$:     MOV     #IDEZ,R1                ;ADDR OF IDENT TO R1
007602  012703  010137                          MOV     #NPCODE,R3              ;ADDR OF NON-PRINT-CODES TO R3
007606  012702  000003                  2$:     MOV     #3,R2                   ;NO. OF ID'S PER LINE TO R2
007612  012704  000010                  3$:     MOV     #10,R4                  ;NO. OF CHARS PER ID TO R4
007616  121327  000055                  4$:     CMPB    (R3),#55                ;ZERO TERMINATOR IN NP TABLE?
007622  001422                                  BEQ     7$                      ;BRANCH IF YES
007624  112100                                  MOVB    (R1)+,R0                ;GET ID CHARACTERS
007626  104015                                  PRINTC                          ;AND PRINT A
007630  005304                                  DEC     R4                      ;GROUP OF
007632  001371                                  BNE     4$                      ;8 CHARACTERS
007634  112300                                  MOVB    (R3)+,R0                ;GET NP CODE FROM TABLE
007636  012704  000003                          MOV     #3,R4                   ;AND
007642  104015                          5$:     PRINTC                          ;TRY TO PRINT IT
007644  005304                                  DEC     R4                      ;THREE
007646  001375                                  BNE     5$                      ;TIMES
007650  005302                                  DEC     R2                      ;MORE TO GO ON THIS LINE ?
007652  001404                                  BEQ     6$                      ;BRANCH IF NO
007654  004767  177672                          JSR     PC,SP2                  ;SEND 3 SPACES
007660  104015                                  PRINTC
007662  000753                                  BR      3$                      ;BRANCH TO CONTINUE LINE
007664  104012                          6$:     CRLF
007666  000747                                  BR      2$                      ;GO DO NEXT LINE
007670  104012                          7$:     CRLF
007672  104005                                  CHAIN                           ;CHAIN TO NEXT TEST
007674  000740                                  BR      1$



007676     060     060     060          IDEZ:   .ASCII  /000  NUL001  SOH002  STX/
007701     040     040     116
007704     125     114     060
007707     060     061     040
007712     040     123     117
007715     110     060     060
007720     062     040     040
007723     123     124     130
007726     060     060     066                  .ASCII  /006  ACK020  DLE021  DC1/
007731     040     040     101
007734     103     113     060
007737     062     060     040
007742     040     104     114
```

```
007745     105     060     062
007750     061     040     040
007753     104     103     061
007756     060     062     062                  .ASCII  /022  DC2023  DC3024  DC4/
007761     040     040     104
007764     103     062     060
007767     062     063     040
007772     040     104     103
007775     063     060     062
010000     064     040     040
010003     104     103     064
010006     060     062     065                  .ASCII  /025  NAK026  SYN027  ETB/
010011     040     040     116
010014     101     113     060
010017     062     066     040
010022     040     123     131
010025     116     060     062
010030     067     040     040
010033     105     124     102
010036     060     063     060                  .ASCII  /030  CAN031  EM 032  SUB/
010041     040     040     103
010044     101     116     060
010047     063     061     040
010052     040     105     115
010055     040     060     063
010060     062     040     040
010063     123     125     102
010066     060     063     064                  .ASCII  /034  FS 035  GS 036  RS /
010071     040     040     106
010074     123     040     060
010077     063     065     040
010102     040     107     123
010105     040     060     063
010110     066     040     040
010113     122     123     040
010116     060     063     067                  .ASCII  /037  US 177  DEL /
010121     040     040     125
010124     123     040     061
010127     067     067     040
010132     040     104     105
010135     114     040
010137     000     002     006          NPCODE: .BYTE   0,2,6,20,21,22,23,24
010142     020     021     022
010145     023     024
010147     025     025     027                  .BYTE   25,25,27,30,31,32,34,35
010152     030     031     032
010155     034     035
010157     036     037     177                  .BYTE   36,37,177,55
010162     055
                                                .EVEN
```

```
                                      ;XXXXXXXXXX
                                      ;
                                      ;PT3 -- CARRIAGE RETURN TEST
                                      ;
                                      ;        THE LINE CONSISTS OF A STRING OF O'S AND
                                      ;        X'S.  FIRST, THE O'S ARE PRINTED OUT TO THE LAST
                                      ;        COLUMN WITH A SPACE SEPARATING EACH.  THEN THE
                                      ;        CARRIAGE IS SPACED TO THE FIRST BLANK SPACE, AN X
                                      ;        IS PRINTED AND THEN RETURNED TO THE MARGIN.  THIS
                                      ;        PROCESS IS CONTINUE UNTIL ALL SPACES BETWEEN
                                      ;        THE ZEROES HAVE BEEN FILLED.
                                      ;
                                      ;XXXXXXXXXX

010164  000003                        PT3:    3                       ;TEST NUMBER
010166  010304                                PT4                     ;NEXT TEST
010170  104016                                PRTHDR                  ;TYPE HEADER
010172  005067  170476                1$:     CLR     SPCNT           ;CLEAR SPACE COUNTER
010176  016701  170450                        MOV     WIDTH,R1        ;POSITION COUNTER TO R1
010202  012700  000117                2$:     MOV     #117,R0         ;"O" TO R0
010206  104015                                PRINTC                  ;PRINT THE "O"
010210  005301                                DEC     R1              ;DECREMENT POSITION COUNTER
010212  001404                                BEQ     3$              ;BRANCH IF 0
010214  004767  177340                        JSR     PC,SPC          ;SEND SPACE
010220  005301                                DEC     R1              ;DECREMENT POSITION COUNTER
010222  001367                                BNE     2$              ;BRANCH IF NOT ZERO
010224  104022                        3$:     CR                      ;SEND A CR
010226  012767  000001  170440                MOV     #1,SPCNT        ;SPACE, COUNTER SET TO 1
010234  016701  170434                4$:     MOV     SPCNT,R1        ;NO. OF SPACES TO R1
010240  004767  177314                5$:     JSR     PC,SPC          ;SEND SPACE
010244  005301                                DEC     R1              ;DECREMENT SPACE COUNTER
010246  001374                                BNE     5$              ;BRANCH IF NOT ZERO
010250  012700  000130                        MOV     #130,R0         ;"X" INTO R0
010254  104015                                PRINTC                  ;PRINT "X"
010256  104022                                CR                      ;PRINT CR
010260  062767  000002  170406                ADD     #2,SPCNT        ;INCREMENT SPACE COUNT BY 2
010266  026767  170402  170356                CMP     SPCNT,WIDTH     ;COMPARE POSITION COUNTER WITH COLM. COUNT
010274  103757                                BLO     4$              ;BRANCH IF LOWER
010276  104014                                LF                      ;SEND LF
010300  104005                                CHAIN                   ;CHAIN TO NEXT TEST
010302  000733                                BR      1$              ;REPEAT TEST
```

```
                                      ;XXXXXXXXXX
                                      ;
                                      ;PT4 -- MULTIPLE LINE FEED TEST -- 63 LINE FEEDS ARE
                                      ;        SENT WITH A REFERENCE LINE AT THE START AND END.
                                      ;        A NUMBER IS PRINTED WHICH INDICATES THE NUMBER OF LINE
                                      ;        FEEDS THAT WILL BE ISSUED BEFORE THE NEXT
                                      ;        NUMBER OR REFERENCE LINE IS PRINTED.
                                      ;
                                      ;XXXXXXXXXX

010304  000004                        PT4:    4                       ;TEST NUMBER
010306  010462                                PT5                     ;NEXT TEST
010310  104016                                PRTHDR                  ;TYPE HEADER
010312  012767  000001  170366        1$:     MOV     #1,LFCNT        ;LINE FEED COUNT TO 1
010320  016701  170326                        MOV     WIDTH,R1        ;COLUMN COUNT TO R1
010324  012702  010444                        MOV     #LINE3,R2       ;ADDR OF NUMBER FIELD TO R2
010330  004767  000060                        JSR     PC,REF          ;PRINT REFERENCE LINE
010334  016701  170346                2$:     MOV     LFCNT,R1        ;LINE FEED COUNT TO R1
010340  104014                        3$:     LF                      ;SEND LF
010342  005301                                DEC     R1              ;DECREMENT COUNTER
010344  001375                                BNE     3$              ;BRANCH IF NOT YET 0
010346  006367  170334                        ASL     LFCNT           ;DOUBLE LINE FEED COUNT
010352  022767  000100  170326                CMP     #BIT6,LFCNT     ;TEST IF COUNT IS 32
010360  001406                                BEQ     4$              ;BRANCH IF =32, END
010362  112200                                MOVB    (R2)+,R0        ;NUMBER TO R0
010364  104015                                PRINTC                  ;PRINT IT
010366  112200                                MOVB    (R2)+,R0        ;NUMBER TO R0
010370  104015                                PRINTC                  ;PRINT IT
010372  104022                                CR                      ;PRINT CR
010374  000757                                BR      2$              ;DRIVE THE LINEFEEDS
010376  016701  170250                4$:     MOV     WIDTH,R1        ;COLUMN COUNT TO R1
010402  004767  000006                        JSR     PC,REF          ;SEND END REFERENCE LINE
010406  104014                                LF                      ;ADVANCE PAPER
010410  104005                                CHAIN
010412  000737                                BR      1$              ;REPEAT TEST
010414  112200                        REF:    MOVB    (R2)+,R0        ;NUMBER TO R0
010416  104015                                PRINTC                  ;PRINT IT
010420  112200                                MOVB    (R2)+,R0        ;NUMBER TO R0
010422  104015                                PRINTC                  ;PRINT IT
010424  005741                                TST     -(R1)           ;DECREASE COUNTER BY 2
010426  012700  000137                        MOV     #137,R0         ;DASH (-) TO R0
010432  104015                        1$:     PRINTC                  ;PRINT IT
010434  005301                                DEC     R1              ;DECREMENT COLUMN COUNTER
010436  001375                                BNE     1$              ;BRANCH IF NO ZERO
010440  104022                                CR                      ;PRINT CR
010442  000207                                RTS     PC              ;RETURN

010444     060     061     060        LINE3:  .ASCII  /01020408163200/
010447     062     060     064
010452     060     070     061
010455     066     063     062
010460     060     060
```

```
   23100                                         ;XXXXXXXXXX
   23200                                         ;PT5--   SINGLE LINE FEED TEST --  TESTS THE LINE FEED
   23300                                         ;        CAPABILITY FROM ALL COLUMNS.
   23400                                         ;XXXXXXXXXX
   23500
   23600 010462  000005                          PT5:     5                           ;TEST NUMBER
   23700 010464  010666                                   PT6                         ;NEXT TEST
   23800 010466  104016                                   PRTHDR                      ;TYPE HEADER
   23900 010470  016701  170156                  1$:      MOV     WIDTH,R1            ;COLUMN COUNT TO R1
   24000 010474  005741                                   TST     -(R1)               ;DECREASE BY 2
   24100 010476  012700  000060                           MOV     #60,R0              ;"0" TO R0
   24200 010502  104015                          2$:      PRINTC                      ;SEND 0
   24300 010504  005301                                   DEC     R1                  ;DECREMENT COLUMN COUNTER
   24400 010506  001375                                   BNE     2$                  ;BRANCH IF NOT ZERO
   24500 010510  012700  000062                           MOV     #62,R0              ;SEND A 2
   24600 010514  104015                                   PRINTC
   24700 010516  104015                                   PRINTC                      ;SEND A SECOND TWO
   24800 010520  026727  170126  000204                   CMP     WIDTH,#132.         ;COMPARE COLUMN COUNT
   24900 010526  001404                                   BEQ     3$                  ;BRANCH IF EQ 132
   25000 010530  012700  003410                           MOV     #3410,R0            ;DELAY 1.8 SEC
   25100 010534  104010                                   DELAY
   25200 010536  000407                                   BR      5$
   25300 010540  012700  000063                  3$:      MOV     #63,R0              ;3"S TO R0
   25400 010544  012701  000100                           MOV     #100,R1             ;64 TO COUNTER
   25500 010550  104015                          4$:      PRINTC                      ;SEND CHARACTER
   25600 010552  005301                                   DEC     R1                  ;DECREMENT COUNT
   25700 010554  001375                                   BNE     4$                  ;BRANCH IF NOT ZERO
   25800 010556  104012                          5$:      CRLF                        ;SEND A CR,LF
   25900 010560  016701  170066                           MOV     WIDTH,R1            ;NO. COLUMNS TO R1
   26000 010564  012700  000134                  6$:      MOV     #134,R0             ;BACKSLASH TO R0
   26100 010570  104015                                   PRINTC                      ;SEND IT
   26200 010572  104014                                   LF                          ;PRINT LF
   26300 010574  005301                                   DEC     R1                  ;DECREMENT COUNTER
   26400 010576  001372                                   BNE     6$                  ;BRANCH IF NOT ZERO.
   26500 010600  104022                                   CR                          ;SEND CR
   26600 010602  004767  000022                           JSR     PC,PT5AL            ;SEND REF LINE #1
   26700 010606  104012                                   CRLF                        ;SEND A CR,LF
   26800 010610  012700  001750                           MOV     #1750,R0            ;DELAY 1 SEC
   26900 010614  104010                                   DELAY
   27000 010616  004767  000006                           JSR     PC,PT5AL            ;SEND A SECOND REF. LINE
   27100 010622  104012                                   CRLF                        ;SEND A CR,LF
   27200 010624  104005                                   CHAIN                       ;CHAIN TO NEXT TEST
   27300 010626  000720                                   BR      1$                  ;REPEAT TEST
   27400 010630  016701  170016                  PT5AL:   MOV     WIDTH,R1            ;COLUMN COUNT TO R1
   27500 010634  012700  000061                           MOV     #61,R0              ;"1" TO R0
   27600 010640  104015                          1$:      PRINTC                      ;PRINT R0
   27700 010642  005301                                   DEC     R1                  ;DECREMENT COUNTER
   27800 010644  001407                                   BEQ     2$                  ;BRANCH IF=0
   27900 010646  005200                                   INC     R0                  ;INCREMENT CHARACTER
   28000 010650  020027  000071                           CMP     R0,#71              ;COMP CHAR TO "9"
   28100 010654  101771                                   BLOS    1$                  ;BRANCH IF LOWER OR SAME
   28200 010656  012700  000060                           MOV     #60,R0              ;RESET CHAR TO "0"
   28300 010662  000766                                   BR      1$                  ;CONTINUE
   28400 010664  000207                          2$:      RTS     PC                  ;FINISHED, RETURN TO CALLER
```

```
   28600                                         ;XXXXXXXXXX
   28700                                         ;PT6--   BACKSPACE TEST -- A REFERENCE LINE SUCH AS IN
   28800                                         ;        TEST PT5 IS PRINTED.  THE SECOND LINE CONSISTS
   28900                                         ;        OF PRINTING A BACKSLASH, BACKSPACE AND FORWARD
   29000                                         ;        SLASH COMBINATION OUT TO THE GIVEN COLUMN WIDTH.
   29100                                         ;        THIS LINE IS THEN FOLLOWED BY THE SAME TWO REFERENCE
   29200                                         ;        LINES AS PRINTED IN TEST PT5.
   29300                                         ;XXXXXXXXXX
   29400
   29500 010666  000006                          PT6:     6                           ;TEST NUMBER
   29600 010670  011054                                   PT7                         ;NEXT TEST
   29700 010672  104016                                   PRTHDR                      ;PRINT HEADER
   29800 010674  104007                                   TYPEM                       ;PRINT COLUMN # MESG
   29900 010676  014127                                   HDR0
   30000 010700  016701  167746                  1$:      MOV     WIDTH,R1            ;COLUMN COUNT TO R1
   30100 010704  005741                                   TST     -(R1)               ;DECREMENT BY 2
   30200 010706  012700  000060                           MOV     #60,R0              ;"0" TO R0
   30300 010712  104015                          2$:      PRINTC                      ;SEND 0
   30400 010714  005301                                   DEC     R1                  ;DECREMENT COLUMN COUNTER
   30500 010716  001375                                   BNE     2$                  ;BRANCH IF NOT ZERO
   30600 010720  012700  000062                           MOV     #62,R0              ;"2" TO R0
   30700 010724  104015                                   PRINTC                      ;SEND A "2"
   30800 010726  104015                                   PRINTC                      ;SEND A SECOND "2"
   30900 010730  026727  167716  000204                   CMP     WIDTH,#132.         ;COMPARE COLUMN COUNT
   31000 010736  001404                                   BEQ     3$
   31100 010740  012700  003410                           MOV     #3410,R0            ;DELAY 1.8 SEC
   31200 010744  104010                                   DELAY
   31300 010746  000407                                   BR      5$
   31400 010750  012700  000063                  3$:      MOV     #63,R0              ;3"S TO R0
   31500 010754  012701  000100                           MOV     #100,R1             ;64 TO CHAR COUNT
   31600 010760  104015                          4$:      PRINTC                      ;SEND CHAR
   31700 010762  005301                                   DEC     R1                  ;DECREMENT CHAR COUNT
   31800 010764  001375                                   BNE     4$                  ;CONTINUE IF NOT DONE
   31900 010766  104012                          5$:      CRLF                        ;SEND A CR,LF
   32000 010770  016701  167656                           MOV     WIDTH,R1            ;COLUMN COUNT TO R1
   32100 010774  012700  000134                  6$:      MOV     #134,R0             ;BACKSLASH TO R0
   32200 011000  104015                                   PRINTC                      ;SEND IT
   32300 011002  012700  000010                           MOV     #10,R0              ;BACKSPACE TO R0
   32400 011006  104015                                   PRINTC                      ;SEND IT
   32500 011010  012700  000057                           MOV     #57,R0              ;FORWORD SLASH TO R0
   32600 011014  104015                                   PRINTC                      ;SEND IT
   32700 011016  005301                                   DEC     R1                  ;END OF PAPER
   32800 011020  001365                                   BNE     6$                  ;BRANCH IF NO
   32900 011022  104014                                   LF                          ;SEND LF
   33000 011024  104022                                   CR                          ;SEND CR
   33100 011026  004767  177576                           JSR     PC,PT5AL            ;SEND REF LINE #1
   33200 011032  104012                                   CRLF                        ;SEND A CR,LF
   33300 011034  012700  001750                           MOV     #1750,R0            ;DELAY 1 SEC
   33400 011040  104010                                   DELAY
   33500 011042  004767  177562                           JSR     PC,PT5AL            ;SEND SECOND REF LINE
   33600 011046  104012                                   CRLF                        ;SEND A CR,LF
   33700 011050  104005                                   CHAIN                       ;CHAIN TO NEXT TEST
   33800 011052  000712                                   BR      1$                  ;REPEAT TEST
```

```
                                ;XXXXXXXXXX
                                ;
                                ;PT7--  OVERPRINT TEST-- A ROW OF ALTERNATING M'S AND
                                ;       SPACES ARE PRINTED, OUT TO THE LAST COLUMN AND OVERPRINTED TWICE.
                                ;       A SECOND LINE OF ALTERNATING SPACES AND "@'S" IS THEN
                                ;       SENT 3 TIMES AS THE FIRST LINE.  THIS IS FOLLOWED
                                ;       BY A THIRD AND FINAL LINE OF ALTERNATING "&"
                                ;       AND SPACES.
                                ;
                                ;XXXXXXXXXX

011054  000007                  PT7:    7                       ;TEST NUMBER
011056  011266                          PT10                    ;NEXT TEST
011060  104016                          PRTHDR                  ;PRINT MESSAGE
011062  012703  000002          1$:     MOV     #2,R3           ;2 COUNT TO R3
011066  016701  167560          2$:     MOV     WIDTH,R1        ;NO. OF COLUMNS TO R1
011072  012700  000115          3$:     MOV     #115,R0         ;"M" TO R0
011076  104015                          PRINTC                  ;SEND IT
011100  005301                          DEC     R1              ;END OF LINE
011102  001404                          BEQ     4$              ;BRANCH IF YES
011104  004767  176450                  JSR     PC,SPC          ;SEND SPACE
011110  005301                          DEC     R1              ;END OF LINE?
011112  001367                          BNE     3$              ;BRANCH IF NO
011114  022703  000002          4$:     CMP     #2,R3           ;TEST R3
011120  001003                          BNE     6$              ;BRANCH IF NOT FIRST TIME
011122  104022                  5$:     CR                      ;SEND CR
011124  005303                          DEC     R3              ;DECREASE LINE COUNTER
011126  000757                          BR      2$              ;REPEAT LINE
011130  005703                  6$:     TST     R3              ;THIRD TIME?
011132  001373                          BNE     5$              ;BRANCH IF NOT
011134  104012                          CRLF                    ;NEXT LINE
011136  005723                          TST     (R3)+           ;REPEAT COUNTER TO R3
011140  016701  167506          7$:     MOV     WIDTH,R1        ;COLUMN COUNT TO R1
011144  004767  176410          8$:     JSR     PC,SPC          ;SEND SPACE
011150  005301                          DEC     R1              ;DECREASE COLUMN COUNT
011152  001405                          BEQ     9$              ;BRANCH IF 0, END OF LINE
011154  012700  000100                  MOV     #100,R0         ;"@" TO R0
011160  104015                          PRINTC                  ;SEND IT
011162  005301                          DEC     R1              ;DECREASE COLUMN COUNT
011164  001367                          BNE     8$              ;BRANCH IF NOT 0 (NOT END)
011166  022703  000002          9$:     CMP     #2,R3           ;END OF LINE, FIRST TIME?
011172  001003                          BNE     11$             ;BRANCH IF NOT
011174  104022                  10$:    CR                      ;SEND CR
011176  005303                          DEC     R3              ;DECREASE LINE COUNTER
011200  000757                          BR      7$              ;REPEAT LINE
011202  005703                  11$:    TST     R3              ;TEST IF THIRD REPEAT
011204  001373                          BNE     10$             ;BRANCH IF NOT
011206  104012                          CRLF                    ;DO NEXT LINE
011210  005723                          TST     (R3)+           ;LINE REPEAT COUNTER TO R3
011212  016701  167434          12$:    MOV     WIDTH,R1        ;COLUMN COUNT TO R1
011216  012700  000046          13$:    MOV     #46,R0          ;"&" TO R0
011222  104015                          PRINTC                  ;SEND IT
011224  005301                          DEC     R1              ;DECREASE COLUMN COUNT

011226  001404                          BEQ     14$             ;BRANCH IF END
011230  004767  176324                  JSR     PC,SPC          ;SEND SPACE
011234  005301                          DEC     R1              ;DECREASE COLUMN COUNT
```

```
011236  001367                          BNE     13$             ;BRANCH IF NOT END
011240  022703  000002          14$:    CMP     #2,R3           ;TEST IF FIRST TIME
011244  001003                          BNE     16$             ;BRANCH IF =2, FIRST TIME
011246  104022                  15$:    CR                      ;SENT CR
011250  005303                          DEC     R3              ;DECREASE REPEAT COUNTER
011252  000757                          BR      12$             ;PRINT LINE AGAIN
011254  005703                  16$:    TST     R3              ;TEST IF END, R3=0
011256  001373                          BNE     15$             ;BRANCH IF NOT END
011260  104012                          CRLF                    ;SEND CR,LF
011262  104005                          CHAIN                   ;CHAIN TO NEXT TEST
011264  000676                          BR      1$              ;REPEAT TEST
```

```
                                ;XXXXXXXXXX
                                ;
                                ;PT10-- PRINTING FREQUENCY TEST-- 120 H'S ARE PRINTED ON 4 LINES
                                ;       30 PER LINE. THE TEST IS SUCH THAT BETWEEN THE FIRST AND SECOND
                                ;       "H" A 30 MSEC DELAY IS INTRODUCED. THIS DELAY IS THEN INCREASED
                                ;       BETWEEN CHARACTERS OUT TO 60 CHARACTERS IN AN EXPONENTIAL
                                ;       MANNER. THE DELAY IS THEN DECREASED IN THE SAME MANNER OUT TO THE
                                ;       120TH CHARACTER. THIS DELAY IS CALCULATED AS FOLLOWS;
                                ;
                                ;               NEW DELAY = OLD DELAY [+ OR -] (OLD DELAY/16 + OLD DELAY/128 )
                                ;
                                ;XXXXXXXXXX

011266 000010                   PT10:   10                      ;TEST NUMBER
011270 011424                           PT11                    ;NEXT TEST
011272 104016                           PRTHDR                  ;TYPE MESSAGE
011274 012701 000036            1$:     MOV     #36,R1          ;SET R1=30
011300 012702 000170                    MOV     #120.,R2        ;SET CHAR COUNT = 120
011304 012767 000036 000010             MOV     #30.,3$+2       ;SET UP DELAY VALUE
011312 012700 000110            2$:     MOV     #110,R0         ;"H" TO R0
011316 104015                           PRINTC                  ;SEND IT
011320 012700 000036            3$:     MOV     #30.,R0
011324 104010                           DELAY                   ;DELAY
011326 005301                           DEC     R1              ;DEC. COUNT OF CHARS PER LINE
011330 001426                           BEQ     6$              ;BRANCH IF 0, END OF LINE
011332 005302                   4$:     DEC     R2              ;DECREMENT CHAR COUNTER
011334 001430                           BEQ     7$              ;BRANCH IF END
011336 016704 177760                    MOV     3$+2,R4         ;GET OLD DELAY
011342 006204                           ASR     R4              ;CAL 1/16 OF OLD DELAY
011344 006204                           ASR     R4
011346 006204                           ASR     R4
011350 006204                           ASR     R4
011352 010405                           MOV     R4,R5           ;SAVE 1/16 IN R5
011354 006204                           ASR     R4              ;CAL 1/128 OF OLD DELAY
011356 006204                           ASR     R4
011360 006204                           ASR     R4
011362 060405                           ADD     R4,R5           ;1/16 +1/128 TO R5
011364 022702 000074                    CMP     #60.,R2         ;TEST WHICH HALF OF THE 120 CHARS.
011370 003403                           BLE     5$              ;BRANCH IF LT OR EQ 60
011372 160567 177724                    SUB     R5,3$+2         ;GT 51, DECREASE DELAY BY 34 MEC.
011376 000745                           BR      2$              ;GO PRINT AGAIN
011400 060567 177716            5$:     ADD     R5,3$+2         ;LT HALF WAY, ADD DELAY OF 34 MEC.
011404 000742                           BR      2$              ;GO PRINT AGAIN
011406 104012                   6$:     CRLF                    ;SEND CRLF
011410 012701 000036                    MOV     #36,R1          ;SET R1=30
011414 000746                           BR      4$
011416 104012                   7$:     CRLF                    ;SEND CR,LF
011420 104005                           CHAIN                   ;CHAIN TO NEXT TEST
011422 000724                           BR      1$              ;REPEAT TEST
```

```
                                ;XXXXXXXXXX
                                ;
                                ;PT11-- RIBBON FEED TEST-- THIS TEST PRINTS A SINGLE COLUMN OF X'S
                                ;       (24 LINES) DOWN THE LEFT MARGIN OF THE PAGE.
                                ;       VISUALLY CHECK THE RIBBON FEED MECHANISM FOR PROPER OPERATION.
                                ;
                                ;XXXXXXXXXX

011424 000011                   PT11:   11                      ;TEST NUMBER
011426 011456                           PT12                    ;NEXT TEST
011430 104016                           PRTHDR                  ;TYPE MESSAGE
011432 012701 000030            1$:     MOV     #30,R1          ;SET R1=24(10), LINE COUNT
011436 012700 000130            2$:     MOV     #130,R0         ;SET CHAR = X
011442 104015                           PRINTC                  ;PRINT X
011444 104012                           CRLF                    ;SEND CR-LF
011446 005301                           DEC     R1              ;DECREMENT LINE COUNT
011450 001372                           BNE     2$              ;CONTINUE IF NOT DONE TEST
011452 104005                           CHAIN                   ;CHAIN TO NEXT TEST
011454 000766                           BR      1$              ;REPEAT TEST



                                ;XXXXXXXXXX
                                ;
                                ;PT12-- PRINTER BELL TEST-- THE LAST TEST IN THE
                                ;       PRINTER TEST SEQUENCE.  THIS TEST OUTPUTS
                                ;       EIGHT BELL SIGNALS TO THE PRINTER
                                ;
                                ;
                                ;XXXXXXXXXX
                                ;
011456 000012                   PT12:   12                      ;THIS TEST
011460 007372                           PT0                     ;NEXT TEST
011462 104016                           PRTHDR                  ;TYPE HEADER
011464 012701 000010            PT12A:  MOV     #10,R1          ;COUNTER TO R1
011470 012700 000007                    MOV     #7,R0           ;BELL TO R0
011474 104015                   1$:     PRINTC                  ;SEND IT
011476 005301                           DEC     R1              ;DECREMENT COUNT
011500 001375                           BNE     1$              ;BRANCH IF NOT ZERO
011502 104014                           LF
011504 012700 003720                    MOV     #3720,R0        ;DELAY 2 SEC BEFORE RESTARTING
011510 104010                           DELAY
011512 013700 000042                    MOV     @#42,R0         ;CHECK IF UNDER ACT11 OR XXDP
011516 001405                           BEQ     HERE            ;CONTINUE TEST SEQUENCE
011520 000240                           NOP                     ;A RESET WAS FORMERLY HERE
011522 004710                   LOGICAL:JSR     PC,(R0)
011524 000240                           NOP
011526 000240                           NOP
011530 000240                           NOP
011532 104005                   HERE:   CHAIN                   ;CHAIN TO NEXT TEST
011534 000753                           BR      PT12A           ;REPEAT TEST
```

```
                                           ;XXXXXXXXXX
                                           ;
                                           ;PT17-- LIFE TEST
                                           ;       THIS TEST PRINTS 2 FULL LINES OF EACH PRINTABLE
                                           ;       CHARACTER AND OVERPRINTS THE SECOND LINE 4 TIMES.
                                           ;       THIS TEST IS CONTINUOUS RUNNING ONCE INITIATED,
                                           ;       LOOPING AUTOMATICALLY ON ITSELF.
                                           ;       END OF PASS COUNT IS CLEARED WHENEVER TEST IS RESTARTED
                                           ;
                                           ;XXXXXXXXXX

011536  000017                             PT17B:  17                              ;TEST NUMBER
011540  011536                                     PT17B                           ;NEXT TEST
011542  000167  000030                             JMP     PT17D                   ;CONTINUE
011546  000017                             PT17:   17                              ;TEST NUMBER
011550  011536                                     PT17B                           ;NEXT TEST
011552  005067  000336                             CLR     PASCNT                  ;CLEAR PASS COUNT
011556  016704  167070                             MOV     WIDTH,R4                ;INITIALIZE R4
011562  012767  000001  000322                     MOV     #1,DIRTN                ;AND DIRECTION OF PRECESS
011570  104016                                     PRTHDR
011572  104007                                     TYPEM                           ;PRINT COLUMN # MESG
011574  014127                                     HDRO
011576  012703  000041                     PT17D:  MOV     #41,R3                  ;SET START CHAR
011602  005267  000306                             INC     PASCNT
011606  026727  000302  000031                     CMP     PASCNT,#31              ;DO 31 TIMES
011614  001003                                     BNE     20$                     ;BRANCH IF NOT DONE
011616  012767  000001  000270                     MOV     #1,PASCNT               ;START OVER
011624  012700  014042                     20$:    MOV     #PASMES,R0              ;SET MESG ADDR
011630  016701  000260                             MOV     PASCNT,R1               ;# TO CONVERT
011634  012702  000002                             MOV     #2,R2                   ;# DIGITS
011640  104023                                     BTOASC                          ;CONVERT PASCNT TO ASCII
011642  016701  167004                     1$:     MOV     WIDTH,R1                ;SET COLUMN COUNT
011646  010300                             2$:     MOV     R3,R0                   ;GET CHARACTER
011650  004767  000110                             JSR     PC,CKPOS                ;TIME TO INSERT PASS # ?
011654  104015                                     PRINTC                          ;SEND CHAR
011656  005301                                     DEC     R1                      ;DECREMENT COUNT
011660  003372                                     BGT     2$                      ;BRANCH IF NOT DONE
011662  004767  000144                             JSR     PC,ADJR4                ;ADJUST R4 POINTER
011666  104012                                     CRLF
011670  012702  000005                             MOV     #5,R2                   ;SET OVERPRINT COUNT
011674  016701  166752                     3$:     MOV     WIDTH,R1                ;SET COLUMN COUNT
011700  010300                             4$:     MOV     R3,R0                   ;GET CHARACTER
011702  004767  000056                             JSR     PC,CKPOS                ;TIME TO INSERT PASS # ?
011706  104015                                     PRINTC                          ;SEND CHAR
011710  005301                                     DEC     R1                      ;DECREMENT COUNT
011712  003372                                     BGT     4$                      ;BRANCH IF NOT DONE
011714  104022                                     CR                              ;SEND CR
011716  005302                                     DEC     R2                      ;DONE OVERPRINTS ?
011720  001365                                     BNE     3$                      ;NO. CONTINUE
011722  004767  000104                             JSR     PC,ADJR4                ;ADJUST R4 POINTER
011726  104014                                     LF                              ;SEND LF
011730  005203                                     INC     R3                      ;SET NEXT CHAR
011732  022703  000177                             CMP     #177,R3                 ;DONE CHAR SET ?
011736  001341                                     BNE     1$                      ;NO. CONTINUE
011740  004767  000066                             JSR     PC,ADJR4                ;OFFSET POINTER 3 PLACES
011744  004767  000062                             JSR     PC,ADJR4                ;TO RETAIN VISUAL ALIGNMENT
011750  004767  000056                             JSR     PC,ADJR4                ;THROUGH END OF PASS
```

```
011754  104007                                     TYPEM                           ;TYPE END OF PASS MESG
011756  014023                                     ENDPAS
011760  104005                                     CHAIN                           ;REPEAT TEST
011762  000705                                     BR      PT17D
```

```

011764  020401                    CKPOS:  CMP     R4,R1           ;IS IT TIME TO INSERT PASS # ?
011766  001020                            BNE     1$              ;BRANCH IF NO
011770  012700  000040                    MOV     #40,R0          ;PRINT A SPACE
011774  104015                            PRINTC
011776  116700  002040                    MOVB    PASMES,R0       ;PRINT MSG OF PASS COUNT
012002  104015                            PRINTC
012004  116700  002033                    MOVB    PASMES+1,R0
012010  104015                            PRINTC
012012  012700  000040                    MOV     #40,R0          ;PRINT A SPACE
012016  104015                            PRINTC
012020  162701  000003                    SUB     #3,R1           ;ADJUST R1 3 POSITIONS
012024  062716  000002                    ADD     #2,(SP)         ;ADJUST RETURN PC OVER PRINTC
012030  000207                    1$:     RTS     PC

012032  005767  000054            ADJR4:  TST     DIRTN           ;TEST DIRECTION OF PRECESS
012036  001013                            BNE     1$              ;BR IF LEFT
012040  005204                            INC     R4              ;INCREASE POSITION CNTR
012042  020467  166604                    CMP     R4,WIDTH        ;IS R4 > WIDTH ?
012046  101420                            BLOS    3$              ;BR IF NOT GREATER
012050  016704  166576                    MOV     WIDTH,R4        ;CHANGE DIRECTION
012054  005304                            DEC     R4              ;       TO
012056  012767  000001  000026            MOV     #1,DIRTN        ;     LEFT.
012064  000411                            BR      3$
012066  005304                    1$:     DEC     R4              ;DECREASE POSITION CNTR
012070  020427  000004                    CMP     R4,#4   ;LESS THAN 4 ?
012074  002401                            BLT     2$              ;BR IF YES
012076  000404                            BR      3$              ;ELSE EXIT
012100  012704  000005            2$:     MOV     #5,R4           ;SET R4 TO POS 5
012104  005067  000002                    CLR     DIRTN           ;CHANGE DIRECTION TO RIGHT
012110  000207                    3$:     RTS     PC              ;EXIT

012112  000000                    DIRTN:  .WORD   0               ;DIRECTION OF PRECESS (0=LEFT)

012114  000000                    PASCNT: .WORD   0
```

```
                                              .SBTTL  LA36 ECHO TESTS
                                    ;
                                    ;XXXXXXXXXX
                                    ;
                                    ;E020--   CHARACTER ECHO TEST-- ALL PRINTABLE AND
                                    ;         NON-PRINTING CHARACTERS TYPED ON THE KEYBOARD
                                    ;         ARE USED TO DRIVE THE PRINTER, ONE CHARACTER AT
                                    ;         A TIME.  A "RUBOUT" WILL CAUSE THE TEST TO BE
                                    ;         TERMINATED.
                                    ;
                                    ;XXXXXXXXXX
                                    ;
012116  000020                      E020:     20                          ;TEST NUMBER
012120  012166                                E021                        ;NEXT TEST
012122  104016                                PRTHDR                      ;TYPE HEADER
012124  104020                      1$:       READ                        ;GO WAIT FOR KEYBOARD INPUT
012126  012700    000036                      MOV      #30.,R0            ;DELAY FOR HALF DUPLEX
012132  104010                                DELAY
012134  022767    000177   166536             CMP      #177,TEMPCH        ;CHECK IF RUBOUT
C12142  001405                                BEQ      2$                 ;BRANCH IF YES
012144  104017                                PRNT                        ;NO, CHECK PRINTER READY
012146  117777    166442   166444             MOVB     @TKB,@TPB          ;READY, ECHO CHARACTER
012154  000763                                BR       1$
012156  104007                      2$:       TYPEM                       ;PRINT TERMINATION MESSAGE
012160  014272                                ECOEND
012162  104005                                CHAIN                       ;CHAIN TO NEXT TEST
012164  000757                                BR       1$                 ;REPEAT TEST
                                    ;
                                    ;XXXXXXXXXX
                                    ;
                                    ;E021--   LINE ECHO TEST, FAST RATE-- THIS TEST WILL
                                    ;         CAUSE THE CONTINUAL PRINTING OF "0" AT THE MAXIMUM
                                    ;         RATE UNTIL EITHER ANOTHER CHARACTER IS SELECTED
                                    ;         BY PRESSING A KEY ON THE KEYBOARD OR TERMINATION BY THE
                                    ;         RUBOUT.
                                    ;
                                    ;XXXXXXXXXX
                                    ;
012166  000021                      E021:     21                          ;TEST NUMBER
012170  012224                                E022                        ;NEXT TEST
012172  104016                                PRTHDR                      ;TYPE HEADER
012174  012767    000060   166460   E021A:    MOV      #60,REPT           ;CHARACTER TO BE REPEATED (0)
012202  016702    166444            1$:       MOV      WIDTH,R2           ;SET COLUMN COUNT
012206  016700    166450            2$:       MOV      REPT,R0            ;GET CHAR
012212  104015                                PRINTC                      ;PRINT CHAR
012214  005302                                DEC      R2                 ;DEC COLUMN COUNT
012216  003373                                BGT      2$                 ;FINISH LINE
012220  104012                                CRLF                        ;SEND A CR AND LF
012222  000767                                BR       1$
```

```
                                    ;XXXXXXXXXX
                                    ;
                                    ;E022--   LINE ECHO TEST, SLOW RATE-- SAME AS E021 EXCEPT
                                    ;         THAT A DELAY IS INTRODUCED BETWEEN CHARACTERS
                                    ;         TO PRODUCE A LCV ACTION
                                    ;
                                    ;XXXXXXXXXX
                                    ;
012224  000022                      E022:     22
012226  012476                                E023
012230  104016                                PRTHDR                      ;TYPE HEADER
012232  012767    000060   166422   E022A:    MOV      #60,REPT           ;LOAD 0 AS INITIAL CHARACTER
012240  016702    166406            1$:       MOV      WIDTH,R2           ;SET COLUMN COUNT
012244  016700    166412            2$:       MOV      REPT,R0            ;GET CHAR
012250  104015                                PRINTC                      ;PRINT CHAR
012252  005302                                DEC      R2                 ;DEC COLUMN COUNT
012254  001404                                BEQ      3$                 ;BRANCH IF DONE LINE
012256  012700    003410                      MOV      #3410,R0
012262  104010                                DELAY                       ;DELAY 1.8 SEC.
012264  000767                                BR       2$                 ;OUTPUT NEW CHAR.
012266  104012                      3$:       CRLF                        ;SEND A CR AND LF
012270  000763                                BR       1$
```

```
 7400                                          ;**********
 7500                                          ;
 7600                                          ;        THIS FOLLOWING TABLE IS USED BY TEST E023
 7700                                          ;
 7800                                          ;**********
 7900
 8000 012272    116     125     114   MONIC:  .ASCII  /NUL /
      012275    040
 8100 012276    123     117     110           .ASCII  /SOH /
      012301    040
 8200 012302    123     124     130           .ASCII  /STX /
      012305    040
 8300 012306    105     124     130           .ASCII  /ETX /
      012311    040
 8400 012312    105     117     124           .ASCII  /EOT /
      012315    040
 8500 012316    105     116     121           .ASCII  /ENQ /
      012321    040
 8600 012322    101     103     113           .ASCII  /ACK /
      012325    040
 8700 012326    102     105     114           .ASCII  /BEL /
      012331    040
 8800 012332    102     123     040           .ASCII  /BS  /
      012335    040
 8900 012336    110     124     040           .ASCII  /HT  /
      012341    040
 9000 012342    114     106     040           .ASCII  /LF  /
      012345    040
 9100 012346    126     124     040           .ASCII  /VT  /
      012351    040
 9200 012352    106     106     040           .ASCII  /FF  /
      012355    040
 9300 012356    103     122     040           .ASCII  /CR  /
      012361    040
 9400 012362    123     117     040           .ASCII  /SO  /
      012365    040
 9500 012366    123     111     040           .ASCII  /SI  /
      012371    040
 9600 012372    104     114     105           .ASCII  /DLE /
      012375    040
 9700 012376    104     103     061           .ASCII  /DC1 /
      012401    040
 9800 012402    104     103     062           .ASCII  /DC2 /
      012405    040
 9900 012406    104     103     063           .ASCII  /DC3 /
      012411    040
10000 012412    104     103     064           .ASCII  /DC4 /
      012415    040
10100 012416    116     101     113           .ASCII  /NAK /
      012421    040
10200 012422    123     131     116           .ASCII  /SYN /
      012425    040
10300 012426    105     124     102           .ASCII  /ETB /
      012431    040
10400 012432    103     101     116           .ASCII  /CAN /
      012435    040
10500 012436    105     115     040           .ASCII  /EM  /
```

```
      012441    040
10600 012442    123     125     102           .ASCII  /SUB /
      012445    040
10700 012446    105     123     103           .ASCII  /ESC /
      012451    040
10800 012452    106     123     040           .ASCII  /FS  /
      012455    040
10900 012456    107     123     040           .ASCII  /GS  /
      012461    040
11000 012462    122     123     040           .ASCII  /RS  /
      012465    040
11100 012466    125     123     040           .ASCII  /US  /
      012471    040
11200 012472    123     120     040           .ASCII  /SP  /
      012475    040
11300
11400                                         .EVEN
```

```
                                ;XXXXXXXXXX
                                ;
                                ;E023--  CHARACTER CODE TEST-- ANY CHARACTER SELECTED
                                ;        WILL BE ECHOED ALONG WITH ITS OCTAL CODE.
                                ;        A MNEMONIC WILL BE PRINTED INSTEAD OF THE CHARACTER
                                ;        IF IT IS A NON-PRINTING CHARACTER.
                                ;        THE PARITY OF THE RECEIVED CODE WILL ALSO BE
                                ;        INDICATED AS EITHER EVEN OR ODD.
                                ;
                                ;XXXXXXXXXX
                                ;
012476  000023                  E023:    23                      ;TEST NUMBER
012500  013020                           E024                    ;NEXT TEST
012502  104016                           PRTHDR                  ;TYPE HEADER
012504  104020                  1$:      READ                    ;GO WAIT FOR CHARACTER
012506  012700  000036                   MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
012512  104010                           DELAY
012514  026727  166160  000041           CMP     TEMPCH,#41      ;TEST IF CHAR IS PRINTABLE
012522  103015                           BHIS    3$              ;BRANCH IF IT IS
012524  004767  000130                   JSR     PC,STRLN        ;STORE CODE INTO MESSAGE
012530  116700  166144                   MOVB    TEMPCH,R0       ;GET CODE AGAIN
012534  006300                           ASL     R0              ;MULT BY 2
012536  006300                           ASL     R0              ;MULT BY 4
012540  062700  012272                   ADD     #MONIC,R0       ;ADD ADDR OF MNEMONIC TABLE
012544  004767  000166                   JSR     PC,MOVNUM       ;MOV MNEMONIC TO MESSAGE
012550  104000                  2$:      TYPE                    ;TYPE CODE AND MNEMONIC
012552  014321                           E023M                   ;ADDRESS OF MESSAGE
012554  000753                           BR      1$              ;GO WAIT FOR NEXT CHARACTER
012556  026727  166116  000177  3$:      CMP     TEMPCH,#177     ;TEST IF CHAR IS A RUBOUT
012564  001421                           BEQ     4$              ;BRANCH IF RUBOUT
012566  012701  013010                   MOV     #MG24,R1
012572  116721  166102                   MOVB    TEMPCH,(R1)+
012576  112721  000040                   MOVB    #40,(R1)+
012602  112721  000040                   MOVB    #40,(R1)+
012606  112721  000040                   MOVB    #40,(R1)+
012612  004767  000042                   JSR     PC,STRLN        ;STORE CODE INTO MESSAGE
012616  012700  013010                   MOV     #MG24,R0        ;ADDR OF CHAR INTO R0
012622  004767  000110                   JSR     PC,MOVNUM       ;MOVE CHAR INTO MESSAGE
012626  000750                           BR      2$              ;TYPE MESSAGE
012630  004767  000024          4$:      JSR     PC,STRLN        ;RUBOUT, CONVERT AND STOR CODE
012634  012700  013014                   MOV     #MG25,R0        ;ADDR. OF DEL INTO R0
012640  004767  000072                   JSR     PC,MOVNUM       ;MOVE DEL INTO MESSAGE
012644  104000                           TYPE                    ;TYPE MESSAGE
012646  014321                           E023M                   ;ADDR OF MESSAGE
012650  104007                           TYPEM
012652  014272                           ECOEND
012654  104005                           CHAIN                   ;CHAIN TO NEXT TEST
012656  000712                           BR      1$              ;REPEAT TEST
012660  012702  000003          STRLN:   MOV     #3,R2           ;COUNT OF 3 TO R2
012664  012701  014323                   MOV     #LINES5,R1      ;ADDR OF MESG TO R1
012670  062701  000003                   ADD     #3,R1           ;POINT TO LAST SPACE IN MESG
012674  016700  166004          1$:      MOV     PCHAR,R0        ;MOVE OCTAL CODE TO R0
012700  042700  177770                   BIC     #177770,R0      ;SAVE LS OCTAL CHAR
012704  062700  000060                   ADD     #60,R0          ;MAKE ASCII
012710  110041                           MOVB    R0,-(R1)        ;MOVE INTO MESG
012712  005302                           DEC     R2              ;DECREMENT CHAR COUNTER
012714  001407                           BEQ     2$              ;BRANCH IF 3 MOVED
```

```
012716  006267  165762                   ASR     PCHAR           ;NOT THREE, SHIFT NEXT OCTAL
012722  006267  165756                   ASR     PCHAR           ;CHARACTER TO THE RIGHT
012726  006267  165752                   ASR     PCHAR           ;
012732  000760                           BR      1$              ;CONVERT AND STORE NEXT CHAR
012734  000207                  2$:      RTS     PC              ;RETURN TO CALLER
012736  012701  014327          MOVNUM:  MOV     #LINE5A,R1      ;ADDR OF LINES IN R1
012742  012702  000004                   MOV     #4,R2           ;COUNT OF 4 TO R2
012746  112021                  1$:      MOVB    (R0)+,(R1)+     ;MOV 4 CHARS TO MESG AREA
012750  005302                           DEC     R2              ;DECREMENT COUNTER
012752  001375                           BNE     1$              ;BRANCH IF NOT ALL DONE
012754  105767  165722                   TSTB    PARITY          ;TEST PARITY FLAG
012760  001003                           BNE     2$              ;BRANCH IF ODD PARITY
012762  012700  014371                   MOV     #EVEN,R0        ;SET ADDRESS FOR EVEN PARITY MESG
012766  000402                           BR      3$              ;CONTINUE
012770  012700  014375          2$:      MOV     #ODD,R0         ;SET ADDRESS FOR ODD PARITY MESG
012774  012702  000004          3$:      MOV     #4,R2           ;COUNT OF 4 TO R2
013000  112021                  4$:      MOVB    (R0)+,(R1)+     ;MOVE 4 CHARS TO MESG AREA
013002  005302                           DEC     R2              ;DECREMENT COUNTER
013004  001375                           BNE     4$              ;BRANCH IF NOT DONE
013006  000207                           RTS     PC              ;RETURN

013010     040     040     040  MG24:    .ASCII  /    /          ;SAVE CHARACTER CODE
013013     040

                                         .EVEN

013014     104     105     114  MG25:    .ASCII  /DEL /          ;MNEMONIC FOR RUBOUT
013017     040

                                         .EVEN
```

```
                                    ;XXXXXXXXXX
                                    ;E024--  SELECTED PATTERN ECHO TEST-- SELECT 1 TO 256
                                    ;        CHARACTERS.  EACH WILL BE ECHOED
                                    ;        AND STORED UNTIL THE CNTL/C IS SELECTED.
                                    ;        AT THAT TIME ALL CHARACTERS WILL BE PRINTED AS
                                    ;        A CONTINOUS STRING UNTIL EITHER THE RUBOUT IS
                                    ;        SELECTED TO TERMINATE OR THE CNTL/C IS SELECTED
                                    ;        AGAIN.  A TERMINATING CNTL/C FOLLOWED BY ANOTHER
                                    ;        CNTL/C WILL ALWAYS CAUSE THE LAST INPUTTED STRING TO
                                    ;        BE PRINTED.  A TERMINATING CNTL/C FOLLOWED BY A CHARACTER OTHER THAN A
                                    ;        RUBOUT WILL CAUSE A NEW STRING TO BE INPUTTED.
                                    ;XXXXXXXXXXX

013020  000024                      E024:   24                        ;TEST NUMBER
013022  013566                              E025                      ;NEXT TEST
013024  104016                              PRTHDR                    ;TYPE TEST HEADER
013026  005001                      E024B:  CLR     R1                ;CLEAR CHARACTER COUNT
013030  012702  013164                      MOV     #BUFR,R2          ;ADDRESS OF BUFFER TO R2
013034  104020                      1$:     READ                      ;WAIT FOR INPUT
013036  012700  000036                      MOV     #30.,R0           ;DELAY FOR HALF DUPLEX
013042  104010                              DELAY
013044  022767  000177  165626              CMP     #177,TEMPCH       ;TEST IF RUBOUT
013052  001440                              BEQ     TERM              ;BRANCH IF RUBOUT
013054  022767  000003  165616              CMP     #3,TEMPCH         ;TEST IF CNTL-C
013062  001413                              BEQ     OUTPUT            ;BRANCH IF CNTL-C
013064  020127  000400                      CMP     R1,#256.          ;YES, CHECK IF CHAR CNT IS EQ, GT 256
013070  103361                              BHIS    1$                ;BRANCH IF YES, IGNORE CHAR
013072  116722  165602                      MOVB    TEMPCH,(R2)+      ;STORE CHAR INTO BUFFER
013076  005201                              INC     R1                ;INCREMENT CHARACTER COUNT
013100  104017                              PRNT                      ;CHECK IF PRINTER READY
013102  116777  165572  165510              MOVB    TEMPCH,@TPB       ;ECHO CHAR
013110  000751                              BR      1$                ;GO WAIT FOR NEXT CHAR
                                    ;
                                    ;SECTION TO OUTPUT CONTINOUS STRING
                                    ;
013112  020227  013164              OUTPUT: CMP     R2,#BUFR          ;CHECK IF POINTER IS AT START OF TABLE
013116  001403                              BEQ     1$                ;YES, BRANCH
013120  116722  165554                      MOVB    TEMPCH,(R2)+      ;NO, STORE ^C IN TABLE
013124  104013                              SCRLF                     ;SEND A CR LF
013126  012702  013164              1$:     MOV     #BUFR,R2          ;BUFFER ADDRESS TO R2
013132  021227  000003                      CMP     (R2),#3           ;CHECK IF FIRST CHAR IS ^C
013136  001733                              BEQ     E024B             ;YES, LOOK FOR INPUT AGAIN
013140  112200                      2$:     MOVB    (R2)+,R0          ;GET CHARACTER
013142  020027  000003                      CMP     R0,#3             ;DONE STRING?
013146  001767                              BEQ     1$                ;YES, RESTART STRING
013150  104015                              PRINTC                    ;PRINT CHAR
013152  000772                              BR      2$                ;CONTINUE
013154  104007                      TERM:   TYPEM                     ;OUTPUT TERMINATION MESSAGE
013156  014272                              ECOEND
013160  104005                              CHAIN                     ;CHAIN TO NEXT TEST
013162  000721                              BR      E024B             ;REPEAT TEST
013164  000003                      BUFR:   3                         ;INITIALIZE FIRST CHAR AS CNTL-C IN TABLE
013166                                      .BLKB   256.              ;256 CHARACTER BUFFER
```

```
                                    ;XXXXXXXXXXX
                                    ;
                                    ;E025--  BELL ECHO TEST-- A MESSAGE IS PRINTED AND
                                    ;        THE TEST WAITS FOR SOME PRINTABLE CHARACTER
                                    ;        TO BE SELECTED ON THE KEYBOARD (GT040).  THIS
                                    ;        TEST IS VALID ONLY IF THE PAPER WIDTH IS GT 64
                                    ;        COLUMNS.  IF LT64 COLUMNS AN ILLEGAL BELL TEST
                                    ;        MESSAGE IS PRINTED.
                                    ;
                                    ;XXXXXXXXXXX
                                    ;
013566  000025                      E025:   25                        ;TEST NUMBER
013570  012116                              E020                      ;NEXT TEST HEADER
013572  104016                              PRTHDR                    ;PRINT HEADER
013574  026727  165052  000101      1$:     CMP     WIDTH,#101        ;TEST IF COLUMN COUNTIS EQ,GT 64
013602  103427                              BLO     4$                ;BRANCH IF NOT
013604  104007                              TYPEM                     ;TYPE TEST MESG
013606  014145                              E025MA
013610  000402                              BR      3$                ;WAIT FOR CHAR
013612  104000                      2$:     TYPE                      ;TYPE TEST MESG ON TERM CHAR RCVD ON
013614  014145                              E025MA
013616  104020                      3$:     READ                      ;WAIT FOR OPERATOR RESPONSE
013620  012700  000036                      MOV     #30.,R0           ;DELAY FOR HALF DUPLEX
013624  104010                              DELAY
013626  026727  165046  000040              CMP     TEMPCH,#40        ;TEST IF PRINTABLE
013634  103770                              BLO     3$                ;BRANCH IF NON-PRINTABLE
013636  022767  000177  165034              CMP     #177,TEMPCH       ;CHECK IF CHAR IS RUBOUT
013644  001410                              BEQ     5$                ;BRANCH IF YES
013646  104017                              PRNT                      ;CHECK IF PRINTER IS READY
013650  116777  165024  164742              MOVB    TEMPCH,@TPB       ;PRINT CHAR. (BELL SHOULD SOUND)
013656  104013                              SCRLF                     ;SEND A CRLF
013660  000754                              BR      2$                ;REPEAT
013662  104007                      4$:     TYPEM                     ;TYPE ERROR MESSAGE
013664  014245                              E025MB
013666  104007                      5$:     TYPEM                     ;PRINT TERMINATION
013670  014272                              ECOEND
013672  104005                              CHAIN                     ;EXIT TO NEXT TEST
013674  000737                              BR      1$                ;REPEAT TEST
```

```
 29500
```

```
 100                                       .SBTTL  MISC. DIAGNOSTIC MESSAGES
 200                                   ;
 300 013676    007    002    200    STARTM: .ASCII  <7><2><ACRLF><17>/CZLACEO LA36 TERM (DL11 & KL11)/<ACRLF>
     013701    017    103    132
     013704    114    101    103
     013707    105    060    040
     013712    114    101    063
     013715    066    040    124
     013720    105    122    115
     013723    040    050    104
     013726    114    061    061
     013731    040    046    040
     013734    113    114    061
     013737    061    051    200
 400 013742    114    101    063            .ASCII  /LA36 TERMINAL DIAGNOSTIC/<ACRLF>
     013745    066    040    124
     013750    105    122    115
     013753    111    116    101
     013756    114    040    104
     013761    111    101    107
     013764    116    117    123
     013767    124    111    103
     013772    200
 500 013773    104    114    061            .ASCIZ  /DL11 & KL11 INTERFACE/<ACRLF><12>
     013776    061    040    046
     014001    040    113    114
     014004    061    061    040
     014007    111    116    124
     014012    105    122    106
     014015    101    103    105
     014020    200    012    000
 600 014023    200    012    105    ENDPAS: .ASCII  <ACRLF><12>/END OF PASS  /
     014026    116    104    040
     014031    117    106    040
     014034    120    101    123
     014037    123    040    040
 700 014042    060    060    060    PASMES: .ASCIZ  /0000/<ACRLF><12>
     014045    060    200    012
     014050    000
 800 014051    200    103    117    DL11S:  .ASCII  <ACRLF>/CONSOLE & /
     014054    116    123    117
     014057    114    105    040
     014062    046    040
 900 014064    060    060    040    DL11S1: .ASCIZ  /00 DL11'S UNDER TEST/<ACRLF><12>
     014067    104    114    061
     014072    061    047    123
     014075    040    125    116
     014100    104    105    122
     014103    040    124    105
     014106    123    124    200
     014111    012    000
1000 014113    007    002    200    HDRMSG: .ASCIZ  <7><2><ACRLF><17><12>/TEST #/
     014116    017    012    124
     014121    105    123    124
     014124    040    043    000
1100 014127    060    060    060    HDR0:   .ASCIZ  /000 COLUMNS/<ACRLF><12>
     014132    040    103    117
     014135      114      125      115
     014140      116      123      200
     014143      012      000
1200 014145      124      131      120  E025MA: .ASCII  /TYPE ANY PRINTABLE CHARACTER /
     014150      105      040      101
     014153      116      131      040
     014156      120      122      111
     014161      116      124      101
     014164      102      114      105
     014167      040      103      110
     014172      101      122      101
     014175      103      124      105
     014200      122      040
1300 014202      101      116      104          .ASCIZ  /AND LISTEN FOR BELL................/
     014205      040      114      111
     014210      123      124      105
     014213      116      040      106
     014216      117      122      040
     014221      102      105      114
     014224      114      056      056
     014227      056      056      056
     014232      056      056      056
     014235      056      056      056
     014240      056      056      056
     014243      056      000
1400 014245      200      116      117  E025MB: .ASCIZ  <ACRLF>/NOT ENOUGH COLUMNS/<ACRLF>
     014250      124      040      105
     014253      116      117      125
     014256      107      110      040
     014261      103      117      114
     014264      125      115      116
     014267      123      200      000
1500 014272      200      105      103  ECOEND: .ASCIZ  <ACRLF>/ECHO TEST TERMINATED/<ACRLF>
     014275      110      117      040
     014300      124      105      123
     014303      124      040      124
     014306      105      122      115
     014311      111      116      101
     014314      124      105      104
     014317      200      000
1600 014321      040      040           E023M:  .ASCII  /  /
1700 014323      040      040      040  LINE5:  .ASCII  /    /           ;MESG FOR TEST E024
     014326      040
1800 014327      040      040      040  LINE5A: .ASCIZ  /        /<ACRLF>
     014332      040      040      040
     014335      040      040      200
     014340      000
1900 014341      200      017      012  MESG3:  .ASCIZ  <ACRLF><17><12>/SELECT TEST NUMBER  /
     014344      123      105      114
     014347      105      103      124
     014352      040      124      105
     014355      123      124      040
     014360      116      125      115
     014363      102      105      122
     014366      040      040      000
2000 014371      105      126      105  EVEN:   .ASCII  /EVEN/
     014374      116
```

```
2100 014375      117      104      104  ODD:    .ASCII  /ODD /
     014400      040
2200 014401      124      131      120  OPMSG:  .ASCIZ  /TYPE ANY CHARACTER/
     014404      105      040      101
     014407      116      131      040
     014412      103      110      101
     014415      122      101      103
     014420      124      105      122
     014423      000
2300 014424      125      123      105  NOSWR:  .ASCIZ  /USE SOFTWARE SWITCH REG AT MEMORY ADDR 176/<7>
     014427      040      123      117
     014432      106      124      127
     014435      101      122      105
     014440      040      123      127
     014443      111      124      103
     014446      110      040      122
     014451      105      107      040
     014454      101      124      040
     014457      115      105      115
     014462      117      122      131
     014465      040      101      104
     014470      104      122      040
     014473      061      067      066
     014476      007      000
2400
2500
2600          000001                           .END
```

```
ACRLF = 000200          CHAINN  001412          EO25MA  014145          PRGTAB  002522          START3  000772
ADJR4   012032          CHAINY  001534          EO25MB  014245          PRINTC= 104015          STLSPV  003406
ADTENP  004100          CHALT = 104006          ERR     003320          PRNT  = 104017          STLSRV  003356
AREAD = 104021          CHLT    000720          ERRHLT  003354          PRTHDR= 104016          STPCHV= 104004
AT0     005266          CKPOS   011764          ERROR = 104001          PRTY4 = 000200          STPPA   003424
AT0X    005270          CNTLSW  000634          EVEN    014371          PRTY7 = 000340          STPRA   003374
AT1     005320          CNTR    000716          FORWD = 104024          PSW   = 177776          STRDRV= 104003
AT10    005744          CNVCTR  004072          FORWDA  003614          PT0     007372          STRLN   012660
AT11    006002          CONADD  000602          FORWDB  003622          PT1     007446          SWREG   000176
AT12    006042          CONIT   003710          FSTDL   000632          PT10    011266          TEMP    000712
AT13    006116          CONSET  003714          HDRMSG  014113          PT11    011424          TEMPCH  000700
AT14    006176          CONVEC  000604          HDRO    014127          PT12    011456          TENPWR  004076
AT15    006262          COUNT3  000666          HERE    011532          PT12A   011464          TERM    013154
AT16    006362          CR    = 104022          ICTR    000660          PT17    011546          TESTC   002460
AT17    006430          CRBUF   000646          IDEZ    007676          PT17B   011536          TIMER   000672
AT2     005352          CRLF  = 104012          INCHK   000710          PT17D   011576          TKB     000614
AT20    006500          CTRA    000650          LEVEL   000654          PT2     007570          TKLVL   000624
AT21    006572          CURTST  000676          LF    = 104014          PT3     010164          TKS     000612
AT22    006672          DELAY = 104010          LFCNT   000706          PT4     010304          TKVTR   000622
AT23    007000          DIGIT   004074          LINE3   010444          PT5     010462          TPB     000620
AT24    007112          DIRTN   012112          LINE5   014323          PT5AL   010630          TPBS    004004
AT25    007212          DISPRE  000174          LINE5A  014327          PT6     010666          TPLVL   000630
AT26    007270          DLADR   000606          LOGICA  011522          PT7     011054          TPS     000616
AT3     005404          DLCNT   000656          LSI11 = 001000          READ  = 104020          TPSS    004002
AT4     005436          DLNR    000610          MACHER  000004          READC = 104025          TPVTR   000626
AT5     005526          DLY     003436          MESG3   014341          READ1   004212          TTYCTL= 104011
AT6     005604          DL11S   014051          MG24    013010          REF     010414          TTY1    001776
AT7     005674          DL11S1  014064          MG25    013014          REPT    000662          TTY1B   002062
BIT0  = 000001          ECOEND  014272          MONIC   012272          RESTRT  003516          TYP     003076
BIT1  = 000002          EHALT = 104002          MOVNUM  012736          RTNNO   000636          TYPE  = 104000
BIT10 = 002000          EHLT    003346          NEXT    001654          SAVR6   003514          TYPEM = 104007
BIT11 = 004000          EMTINT  002722          NEXT1   001674          SCOPSW= 040000          TYPM    003164
BIT12 = 010000          EMTTAB  002776          NITRSW= 004000          SCOPTR  000642          WAITF   001700
BIT13 = 020000          ENDPAS  014023          NOSWR   014424          SCRLF = 104013          WIDTH   000652
BIT14 = 040000          END2    001332          NPCODE  010137          SELHLT  000734          XCSR    000670
BIT15 = 100000          END2A   001320          NXTST   000640          SKIP    001630          $AREAD  003640
BIT2  = 000004          END3    001262          ODD     014375          SPARET  003072          $BTASC  004006
BIT3  = 000010          END4    001334          OPEN  = 000000          SPBOT   000600          $CR     003226
BIT4  = 000020          EO20    012116          OPMSG   014401          SPC     007560          $CRLF   003214
BIT5  = 000040          EO21    012166          OUTPUT  013112          SPCNT   000674          $FORWD  003562
BIT6  = 000100          EO21A   012174          PARITY  000702          SPSP    007540          $LF     003216
BIT7  = 000200          EO22    012224          PASCNT  012114          SP2     007552          $PRHDR  003236
BIT8  = 000400          EO22A   012232          PASMES  014042          SR      000714          $PRNT   004314
BIT9  = 001000          EO23    012476          PCHAR   000704          START   001010          $PRTC   004324
BRCTR   000664          EO23M   014321          PFAIL   003460          STARTM  013676          $READ   004112
BTOASC= 104023          EO24    013020          POPSP = 005726          STARTX  001024          $READC  004204
BUFR    013164          EO24B   013026          POPSP2= 022626          START1  000736          $SCRLF  003142
CHAIN = 104005          EO25    013566          PRGID   000644          START2  000754

. ABS.  014500    000
        000000    001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  2846 WORDS  ( 12 PAGES)
DYNAMIC MEMORY:  3844 WORDS  ( 14 PAGES)
ELAPSED TIME:  00:00:52
CZLACE.BIN,CZLACE.LST/-SP=CZLACE.MAC
```