LA36 DL11 KL11

LA36 TERMINAL (DL11 + KL11
MD-11-DZLAC-D

EP-DZLAC-D-DL

FICHE 1 OF 1

JAN 1978
digital
MADE IN USA

NLIST
REPT 0
LIST

IDENTIFICATION

Product Code:	MAINDEC-11-DZLAC-D-D
Product Name:	LA36 Terminal (DL11 & KL11 INTERFACE)
Date Created:	August 1977
Maintainer:	Diagnostic Group
Authors:	Robert Baker P. Quenneville R. Schauber

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 PDP11 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, ie., (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 corrosponding 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 MDEC-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 program will then be waiting for control via the switch register. the CPU will halt at location SELHLT.

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

SWITCH NUMBER	DESCRIPTION
15	1(up) = HALT 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 START-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 position and pressing 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 2 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 timeing, and the execution of special DLV11 I/O tests. The DLV11 has no maintenance mode and will cause the 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 test sequence. Placing switch 8 up will cause the program to halt (at SELHLT) at the completion of the current test. After the halt set the control switches as desired and set switches 5 to 0 to the next desired test number then 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 - testing the full 132 (10) columns - leave switches 7 - 0 down.

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 only 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 detected in the test selection (illegal test number or control character) a question 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*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.

EXAMPLE:

```
          @@@     ```
          !!!     AAA     aaa
          """     BBB     bbb
          ###     CCC     ccc
          $$$     DDD     ddd
          %%%     EEE     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
          444     TTT     ttt
          555     UUU     uuu
          666     VVV     vvv
          777     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 separating 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 line between each printed line.

SEQ 0013

EXAMPLE:

```
001   SOH  002  STX  006  ACK
020   DLE  021  DC1  022  DC2
023   DC3  024  DC4  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 O's and spaces, starting with a O. 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:

```
OXOXOXOXOXOXOXOXOXOXOXOXOXOX
```

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 O's and spaces, then print a diaginal 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 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 is 63(10) lines between the two dashed 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 Lines
31
     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 miss prints or missing characters during the first 16(10) characters of the middle reference line. Also, any paper feed problems will cause miss-alignment of the slashes forming the diagonal line.

SEQ 0016

EXAMPLE:

```
0000000000000000000000000000022
```

```
12345678901234567890123456789O
12345678901234567890123456789O
```

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. Any 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:

```
0000000000000000000000000000022
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
12345678901234567890123456789O
12345678901234567890123456789O
```

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 characters are each overprinted two times. The rows consist of the following characters alternated across the line:

Row 1	M-SP
Row 2	SP-@
Row 3	&-SP

The resulting pattern will be a checkerboard pattern and the overprinted characters should be aligned properly with the initial characters.

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 overprinted. There will be three lines of each character with a blank line between each group of characters. The characters in each group should 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 logrithmic 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 logrithmic decrease:

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

Look for possible miss-alignment 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
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
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 received carriage return, there will be a blank line between each printed line.

Example:

X
X
X
X
X
X

6.1.11 Test 12 - Printer Bell Test

This test checks the printer bell buffer to insure that eight bells are distinctly 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

This 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 each character is overprinted 4 times to conserve paper. At the end of each complete pass through the character set a message is printed indicating the number of passes executed. If any character (except "Rubout") is typed on the keyboard during this test, the pattern will change and restart with the typed character. This will only happen if keyboard control is in use.

EXAMPLE:

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
```

If the Auto Line Feed Option is set to produce an automatic line feed 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 between 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 running this test, characters may be lost or garbled whenever a character is typed 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 test is identical to Test 21 except a delay of 1.8 seconds is inserted between 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    EVEN
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 occured 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. Each check is structured as an independent test and the switch register controls 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 "ERRHLT" if switch 14 is down.

```
.NLIST
.ENDP
.LIST
```

```
                        00400           .TITLE  MAINDEC-11-DZLAC-D
                        00500   ;
                        00600   ;LA36 DIAGNOSTIC (DL11 & KL11 INTERFACE)
                        00700   ;
                        00800   ;
                        00900   ;AUTHOR:    ROBERT W. BAKER
                        00950   ;           RALPH A. SCHAUBER
                        01000   ;COPYRIGHT 1974,1977 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                        01100
                        01200
                        01300
                        01400           .SBTTL  SWITCH REGISTER OPTIONS
                        01500   ;
                        01600   ;SWITCH          POSITION        FUNCTION
                        01700   ;
                        01800   ;  15            UP (1)          HALT AT COMPLETION OF CURRENT TEST
                        01900   ;                 DOWN (0)        CONTINUE NORMAL TEST SEQUENCE
                        02000   ;
                        02100   ;  14            UP (1)          CONTINUE ON ERROR
                        02200   ;                                 HALT ON ERROR
                        02300   ;
                        02400   ;  13            UP (1)          DRIVE ONLY CONSOLE TERMINAL
                        02500   ;                 DOWN (0)        DRIVE ALL TERMINALS
                        02600   ;
                        02700   ;  11            UP (1)          LOOP ON INDIVIDUAL TEST
                        02800   ;                 DOWN (0)        NORMAL TEST SEQUENCE
                        02900   ;
                        03000   ;  09            UP (1)          CPU TYPE IS AN LSI-11, 11/03
                        03100   ;                 DOWN (0)        ALL OTHER 11 CPU'S
                        03200   ;
                        03300   ;  08            UP (1)          HALT TO SELECT TEST AT END OF CURRENT TEST
                        03400   ;                 DOWN (0)        LOOP ON TEST SEQUENCE
                        03500   ;
                        03600   ;  05-00                         TEST # SELECTION
                        03700   ;
                        03800   ;  07-00                         # OF COLUMNS AT START-UP
```

```
                        04000           .SBTTL  SPECIAL OPERTIONAL INFORMATION
                        04100   ;
                        04200   ;1. --   THE STANDARD CONSOLE TERMINAL INTERRUPT VECTOR AND REGISTER
                        04300   ;        ADDRESSES ARE USED. TO REDEFINE THE LOCATION OF THE CONSOLE
                        04400   ;        TERMINAL THE SYMBOLIC LOCATIONS "CONADD" AND "CONVEC" SHOULD
                        04500   ;        BE CHANGED BEFORE START UP.
                        04600   ;
                        04700   ;2. --   BEFORE START UP REFER TO THE DESCRIPTION OF THE ROUTINE "DLY"
                        04800   ;        TIMING IS A FUNCTION OF THE PDP11 MODEL AND MEMORY TYPE AND
                        04900   ;        SHOULD BE SET UP BEFORE RUNNING THE DIAGNOSTIC.
                        05000   ;
                        05100   ;3. --   IF CPU IS AN 11/03 , LSI-11 SET SWITCH REGISTER
                        05200   ;        BIT 09 TO A 1. SPECIAL TESTS ARE RUN ON THE DLV11.
                        05300   ;
                        05400   ;4. --   SYSTEMS WITHOUT A HARDWARE SWITCH REGISTER SHOULD USE
                        05500   ;        MEMORY LOCATION 176 AS A SOFTWARE SWITCH REGISTER.
                        05600   ;
                        05700   ;5. --   THIS DIAGNOSTIC IS FOR VERIFICATION OF BASIC TERMINAL
                        05800   ;        FUNCTIONS ONLY. IF THE TERMINAL UNDER TEST HAS HARDWARE
                        05900   ;        OPTIONS INSTALLED RUN DIAGNOSTIC MDEC-11-DZLAF-A, THE
                        06000   ;        LA36 TERMINAL OPTIONS TEST.
```

```
                                06100           .SBTTL  SYSTEM EQUATES
                                06200   ;
                                06300   ;
                                06400   ;       REGISTER EQUATES
                                06500   ;
000000                          06600           R0=%0
000001                          06700           R1=%1
000002                          06800           R2=%2
000003                          06900           R3=%3
000004                          07000           R4=%4
000005                          07100           R5=%5
000006                          07200           SP=%6
000007                          07300           PC=%7
177776                          07400           PSW=177776
                                07500   ;
                                07600   ;       SYSTEM EQUATES
                                07700   ;
000001                          07800           BIT0=1
000002                          07900           BIT1=2
000004                          08000           BIT2=4
000010                          08100           BIT3=10
000020                          08200           BIT4=20
000040                          08300           BIT5=40
000100                          08400           BIT6=100
000200                          08500           BIT7=200
000400                          08600           BIT8=400
001000                          08700           BIT9=1000
002000                          08800           BIT10=2000
004000                          08900           BIT11=4000
010000                          09000           BIT12=10000
020000                          09100           BIT13=20000
040000                          09200           BIT14=40000
100000                          09300           BIT15=100000
000000                          09400           OPEN=0
040000                          09500           SCOPSW=BIT14            ;SCOPE SWITCH
004000                          09600           NITRSW=BIT11            ;TEST LOOP SWITCH
005726                          09700           POPSP=5726              ;POP STACK ONCE
022626                          09800           POPSP2=22626            ;POP STACK TWICE
000340                          09900           PRTY7=340               ;PRIORITY LEVEL DEFINITIONS
000200                          10000           PRTY4=200
000200                          10100           ACRLF=200
001000                          10200           LSI11=BIT9              ;FLAG FOR LSI-11,11/03
                                10300   ;
                                10400   ;       PROGRAM TRAP EQUATES
                                10500   ;
104000                          10600           TYPE=EMT+0
104001                          10700           ERROR=EMT+1
104002                          10800           EHALT=EMT+2
104003                          10900           STRDRV=EMT+3
104004                          11000           STPCHV=EMT+4
104005                          11100           CHAIN=EMT+5
104006                          11200           CHALT=EMT+6
104007                          11300           TYPEM=EMT+7
104010                          11400           DELAY=EMT+10
104011                          11500           TTYCTL=EMT+11
104012                          11600           CRLF=EMT+12
```

```
   119          104013                  11700          SCRLF=EMT+13
   120          104014                  11800          LF=EMT+14
   121          104015                  11900          PRINTC=EMT+15
   122          104016                  12000          PRTHDR=EMT+16
   123          104017                  12100          PRNT=EMT+17
   124          104020                  12200          READ=EMT+20
   125          104021                  12300          AREAD=EMT+21
   126          104022                  12400          CR=EMT+22
   127          104023                  12500          BTOASC=EMT+23
   128          104024                  12600          FORWD=EMT+24
   129          104025                  12700          READC=EMT+25
```

```
                                                13000           .SBTTL  TRAP CATCHER & STARTING ADDRESSES
                                                13100   ;
                                                13200           .ENABL  ABS,AMA
                                                13300
                                                13400
                        000000                  13500           .=0
                                                13600
000000  000002                                  13700           .+2                     ;UNASSIGNED TRAP
000002  000000                                  13800           HALT
000004  000006                                  13900   MACHER: .+2                     ;SP OVERFLOW, BUS ERROR TRAP
000006  000000                                  14000           HALT
000010  000012                                  14100           .+2                     ;RESERVED INSTRUCTION TRAP
000012  000000                                  14200           HALT
000014  000016                                  14300           .+2                     ;TRACE TRAP
000016  000000                                  14400           HALT
000020  000022                                  14500           .+2                     ;TRAP TO CALL IOX
000022  000000                                  14600           HALT
000024  000026                                  14700           .+2                     ;POWER FAIL TRAP
000026  000000                                  14800           HALT
000030  002722                                  14900           EMTINT                  ;EMT TRAP
000032  000340                                  15000           PRTY7
                                                15700
                        000042                  15800           .=42
                                                15900
000042  000000                                  16000           0
                                                16100
                        000046                  16200           .=46
                                                16300
000046  011522                                  16400           LOGICAL
                                                16500
                        000052                  16600           .=52
                                                16700
000052  010000                                  16800           010000
                                                16900
                        000174                  17000           .=174
                                                17100
000174  000000                                  17200   DISPREG:        .WORD   0       ;SOFTWARE DISPLAY
000176  000000                                  17300   SWREG:          .WORD   0       ;SOFTWARE SWITCH REGISTER
                                                17400
000200  000137  001010                          17500           JMP     START           ;START UP WITH I/O TEST RUNNING
000204  000137  000736                          17600           JMP     START1          ;START UP, SKIP ALL I/O TEST
000210  000137  000754                          17700           JMP     START2          ;START UP TERMINAL CONTROL WITH I/O TEST
000214  000137  000772                          17800           JMP     START3          ;START UP TERM CNTL WITHOUT I/O TESTS
                                                17900
                                                18000
                        000600                  18100           .=600
                                                18200
000600  000000                                  18300   SPBOT:  0                       ;BOTTOM OF STACK
```

```
                                18500           .SBTTL  SYMBOL DEFINITIONS
                                18600   ;
                                18700   ;
                                18800   ;
000602  177560                  18900   CONADD: 177560          ;ADDR OF CONSOLE RECEIVER STATUS REG
000604  000060                  19000   CONVEC: 60              ;CONSOLE TERMINAL INTERRUPT VECTOR
000606  176500                  19100   DLADR:  176500          ;ADDRESS OF FIRST DL11, DEFAULT TO DL11-A,B
                                19200                           ;IF DL11-C,D,E,, THEN
                                19300                           ;SET TO 175610 FOR FIRST 16 (OF 31) OR
                                19400                           ;SET TO 176000 FOR LAST 16 (OF 31)
                                19500                           ;OR SET OTHER DESIRED START ADDRESS
000610  000020                  19600   DLNR:   16.             ;# OF DL11'S TO BE INITIALLY ASSUMED
000612  177560                  19700   TKS:    177560          ;CONSOLE RECEIVER STATUS REG
000614  177562                  19800   TKB:    177562          ;CONSOLE RECEIVER BUFFER
000616  177564                  19900   TPS:    177564          ;CONSOLE TRANSMITTER STATUS REG
000620  177566                  20000   TPB:    177566          ;CONSOLE TRANSMITTER BUFFER
000622  000060                  20100   TKVTR:  60              ;C.T. RECEIVER INTERRUPT VECTOR
000624  000200                  20200   TKLVL:  PRTY4           ;C.T. RECEIVER PRIORITY LEVEL
000626  000064                  20300   TPVTR:  64              ;C.T. TRANSMITTER INTERRUPT VECTOR
000630  000200                  20400   TPLVL:  PRTY4           ;C.T. TRANSMITTER PRIORITY LEVEL
000632  000000                  20500   FSTDL:  OPEN            ;ADDRESS OF FIRST ACTIVE DL11
000634  000000                  20600   CNTLSW: OPEN            ;CONSOLE TERMINAL CONTROL SWITCH
000636  000000                  20700   RTNNO:  OPEN            ;CONTAINS CURRENT TEST NUMBER
000640  000000                  20800   NXTST:  OPEN            ;CONTAINS ADDRESS OF NEXT TEST
000642  000000                  20900   SCOPTR: OPEN            ;CONTAINS ADDRESS OF TEST SCOPE ENTRY
000644  000000                  21000   PRGID:  OPEN            ;CONTAINS TEST PROGRAM INDICATORS
000646  000000                  21100   CRBUF:  OPEN
000650  000000                  21200   CTRA:   OPEN
000652  000000                  21300   WIDTH:  OPEN            ;CURRENT PAPER WIDTH, BINARY
000654  000000                  21400   LEVEL:  OPEN            ;LEVEL OF EXECUTION
000656  000000                  21500   DLCNT:  OPEN            ;# OF MULTIPLE DL11S
000660  000000                  21600   ICTR:   OPEN            ;I/O TEST ITERATION COUNT
000662  000000                  21700   REPT:   OPEN            ;TEMP STORAGE FOR TESTS E021 & E022
000664  000000                  21800   BRCTR:  OPEN            ;COUNTER FOR ROUTINE "AREAD"
000666  000000                  21900   COUNT3: OPEN            ;COUNTER FOR ROUTINE "PRINTC"
000670  000000                  22000   XCSR:   OPEN            ;ADDRESS OF MULTIPLE DL11 STATUS
000672  000251                  22100   TIMER:  251             ;1 MSEC COUNTER FOR ROUTINE "DELAY"
000674  000000                  22200   SPCNT:  OPEN            ;COUNTER FOR TEST ROUTINE "PT3"
000676  000000                  22300   CURTST: OPEN            ;ADDRESS OF CURRENT TEST
000700  000000                  22400   TEMPCH: OPEN            ;TEMP STOR FOR ECHO TESTS
000702  000000                  22500   PARITY: OPEN            ;PARITY FLAG FOR RECEIVED CHAR
000704  000000                  22600   PCHAR:  OPEN            ;CHAR CODE WITH PARITY BIT
000706  000000                  22700   LFCNT:  OPEN            ;COUNTER FOR TEST ROUTINE "PT4"
000710  000000                  22800   INCHK:  OPEN            ;CHECK FOR INPUT FLAG
000712  000000                  22900   TEMP:   OPEN            ;TEMPORARY WORKING STORAGE
000714  177570                  23000   SR:     177570          ;SW REG ADDRESS
000716  000000                  23100   CNTR:   OPEN            ;TIME COUNTER FOR LSI11 TESTS
```

234

```
                                        00100          .SBTTL  PROGRAM INITIALIZATION & CONTROL
                                        00200
                                        00300  ;********
                                        00400  ;COMMON HALT---WHEN IN SWITCH REGISTER CONTROL THE CPU
                                        00500  ;        WILL BE ADVANCED TO THIS COMMON HALT WHERE
                                        00600  ;        A NEW TEST WILL BE EXPECTED TO BE STARTED
                                        00700  ;********
                                        00800
000720  005737  000654                  00900  CHLT:    TST     LEVEL           ;TEST CURRENT LEVEL
000724  001403                          01000           BEQ     SELHLT          ;BRANCH IF 0, DO NOT HALT
000726  011600                          01100           MOV     @SP,R0          ;PUT ADDRESS OF CALLER INTO R0
000730  005740                          01200           TST     -(R0)
000732  000000                          01300           HALT
000734  000002                          01400  SELHLT:  RTI                     ;RETURN FROM INTERRUPT
000736  012737  177777  005300          01500  START1:  MOV     #177777,@TOX    ;FORCE END OF I/O TESTS
000744  012737  104006  001700          01600           MOV     #CHALT,WAITF    ;FORCE SR CONTROL
000752  000424                          01700           BR      STARTX
000754  012737  104011  001700          01800  START2:  MOV     #TTYCTL,WAITF   ;FORCE TERMINAL CONTROL
000762  012737  005330  005300          01900           MOV     #AT1,@TOX       ;FORCE ALL I/O TESTS
000770  000415                          02000           BR      STARTX
000772  012737  104011  001700          02100  START3:  MOV     #TTYCTL,WAITF   ;FORCE TERMINAL CONTROL
001000  012737  177777  005300          02200           MOV     #177777,@TOX    ;FORCE END OF I/O TESTS
001006  000406                          02300           BR      STARTX
001010  012737  005330  005300          02400  START:   MOV     #AT1,@TOX       ;FORCE ALL I/O/TESTS
001016  012737  104006  001700          02500           MOV     #CHALT,WAITF    ;FORCE SR CONTROL
001024  012706  000600                  02600  STARTX:  MOV     #SPBOT,SP       ;SET STACK POINTER
001030  013746  000006                  02700           MOV     6,-(SP)         ;SAVE CURRENT VECTOR
001034  013746  000004                  02800           MOV     4,-(SP)
001040  012737  001054  000004          02900           MOV     #10$,4          ;SET UP TIMEOUT VECTOR
001046  005777  177642                  03000           TST     @SR             ;TRY TO REFERENCE HARDWARE SW REG
001052  000411                          03100           BR      11$             ;BRANCH IF NO TIMEOUT TRAP OCCURS
001054  012737  000176  000714          03200  10$:     MOV     #SWREG,SR       ;POINT TO SOFTWARE SWITCH REGISTER
001062  022626                          03300           CMP     (SP)+,(SP)+     ;RESTORE STACK
001064  104000                          03400           TYPE                    ;TELL OPERATOR TO USE SOFTWARE
001066  014407                          03500           NOSWR                   ;SWITCH REG AT LOC 176
001070  012737  000202  000672          03600           MOV     #202,TIMER      ;ADJUST TIMER FOR LSI-11
001076  012637  000004                  03700  11$:     MOV     (SP)+,4         ;RESTORE TIMEOUT VECTOR
001102  012637  000006                  03800           MOV     (SP)+,6
001106  005037  000710                  03900           CLR     INCHK           ;ALLOW INPUT CHECKING
001112  012737  000006  000004          04000           MOV     #6,@ACHER       ;CLEAN UP
001120  005037  000644                  04100           CLR     PRGID           ;INITIALIZE PROGRAM FLAGS
001124  005037  000634                  04200           CLR     CNTLSW          ;INITIALIZE TERMINAL CONTROL SWITCH
001130  005037  000654                  04300           CLR     LEVEL           ;INITIALIZE LEVEL
001134  012737  003450  000024          04400           MOV     #PFAIL,24       ;SET ADDP POWER FAIL ROUTINE
001142  004737  003700                  04500           JSR     PC,CONIT        ;SET UP CONSOLE TERMINAL ADDRESS
```

```
                                        04700   ;**********
                                        04800   ;READ THE PAPER WIDTH, NUMBER OF COLUMNS,
                                        04900   ;FROM SWITCH REGISTER POSITIONS 0-7. SAVE AND
                                        05000   ;CONVERT TO 3 ASCII CHARACTERS. A WIDTH GT132
                                        05100   ;OR LT30 COLUMNS (DECIMAL) WILL BE ABORTED TO 132.
                                        05200   ;THE SWITCHES MAY BE CHANGED ONCE THE PROGRAM TITLE OR THE DL11 COUNT
                                        05300   ;MESSAGE HAS STARTED TO PRINT.
                                        05400   ;**********
                                        05500
001146  017701  177542                  05600           MOV     @SP,R1          ;PUT (SR) INTO R1
001152  042701  177400                  05700           BIC     #177400,R1      ;SAVE ONLY BITS 0-7
001156  020127  000204                  05800           CMP     R1,#204         ;TEST NO. COLUMN GT132
001162  003003                          05900           BGT     2$              ;COLUMNS GT132, DEFAULT TO 132
001164  020127  000035                  06000   1$:     CMP     R1,#35          ;CHECK IF NO. COLUMNS LT 30
001170  101002                          06100           BHI     3$              ;NOT LT 30 NOR GT 132
001172  012701  000204                  06200   2$:     MOV     #204,R1         ;COLUMNS LT 30 OR GT 132, DEFAULT
001176  010137  000652                  06300   3$:     MOV     R1,WIDTH        ;SAVE NO. COLUMNS IN WIDTH
001202  012700  014112                  06400           MOV     #HDR0,R0        ;ADDR TO STORE ASC11 COLUMN VALUE
001206  012702  000003                  06500           MOV     #3,R2           ;DO A 3 CHAR. CONVERSION
001212  104023                          06600           BTOASC                  ;CONVERT NO. COLUMNS TO ASCII
001214  000401                          06700   4$:     BR      5$
001216  000410                          06800           BR      6$
001220  012700  000000                  06900   5$:     MOV     #0,R0           ;TRANSMIT A
001224  104015                          07000           PRINTC                  ;NUL CODE
001226  104007                          07100           TYPEM                   ;TYPE PROGRAM TITLE FIRST TIME RUN
001230  013676                          07200           STARTM
001232  012737  000240  001214          07300           MOV     #NOP,4$
```

```
                                        07500   ;**********
                                        07600   ;THIS NEXT PART CHECKS THE PRESENCE OF DL11-A OR DL11-C
                                        07700   ;STARTING AT 776500. A MESSAGE WILL BE PRINTED INDICATING THE NUMBER
                                        07800   ;PRESENT. THE PRINTER DIAGNOSTIC WILL ADDRESS EACH OF
                                        07900   ;THE MULTIPLE DL11S IN THE SYSTEM IF SWITCH 13 IS DOWN (0)
                                        08000   ;**********
                                        08100
001240  012737  001320  000004          08200   6$:     MOV     #END2A,MACHER   ;INITIALIZE TIME OUT TRAP
001246  013700  000606                  08300           MOV     DLADR,R0        ;ADDRESS OF FIRST DL11 TO R0
001252  013701  000610                  08400           MOV     DLNR,R1         ;SET DL CHECK COUNT
001256  005037  000656                  08500           CLR     DLCNT           ;INITIALIZE DLCNT
001262  005710                          08600   END3:   TST     (R0)            ;IS DL PRESENT?
001264  012737  001332  000004          08700           MOV     #END2,MACHER    ;YES, RESET TIME OUT TRAP
001272  010037  000632                  08800           MOV     R0,FSTDL        ;STORE ADDRESS OF FIRST DL11
001276  000401                          08900           BR      2$              ;CONTINUE
001300  005710                          09000   1$:     TST     (R0)            ;IS DL11 PRESENT
001302  062700  000010                  09100   2$:     ADD     #10,R0          ;POINTER AND DL11 ADDRESS
001306  005237  000656                  09200           INC     DLCNT           ;INCREMENT COUNT OF DL11'S
001312  005301                          09300           DEC     R1              ;DECREMENT DL CHECK COUNT, DONE?
001314  001407                          09400           BEQ     END4            ;BRANCH IF DONE
001316  000770                          09500           BR      1$              ;CHECK PRESENCE OF NEXT DL11
001320  005301                          09600   END2A:  DEC     R1              ;DONE DL CHECK?
001322  001404                          09700           BEQ     END4            ;YES, EXIT
001324  062700  000010                  09800           ADD     #10,R0          ;NO, CHECK NEXT DL
001330  000754                          09900           BR      END3            ;CONTINUE
001332  022626                          10000   END2:   POPSP2                  ;DL11 NOT PRESENT
001334  013701  000656                  10100   END4:   MOV     DLCNT,R1        ;GET # DL11'S
001340  012700  014047                  10200           MOV     #DL11S1,R0      ;ADR OF ASCII CHAR STORAGE
001344  012702  000002                  10300           MOV     #2,R2           ;# OF ASCII CHARS
001350  104023                          10400           BTOASC                  ;CONVERT NUMBER
001352  104007                          10500           TYPEM                   ;TYPE MESSAGE
001354  014034                          10600           DL11S
                                        10700
                                        10800   ;**********
                                        10900   ;EXECUTE THE STRING OF CONSOLE TERMINAL I/O TESTS
                                        11000   ;THEN EITHER HALT AT LOCATION SELHLT OR CONTINUE WITH
                                        11100   ;PRINTER TESTS AS A FUNCTION OF SR BIT 8.
                                        11200   ;**********
                                        11300
001356  005037  000636                  11400           CLR     RTNNO           ;SET ROUTINE NO = 0
001362  005037  000654                  11500           CLR     LEVEL           ;SET LEVEL = 0
001366  023727  005300  177777          11600           CMP     ATOX,#177777    ;SEE IF I/O IS TO BE SKIPPED
001374  001515                          11700           BEQ     SKIP
001376  012737  005276  000640          11800           MOV     #ATO,NXTST      ;ADDRESS OF FIRST I/O TEST
001404  104024                          11900           FORWD                   ;SET UP TEST PARAMETERS
001406  000177  177264                  12000           JMP     @CURTST         ;GO TO I/O TEST ROUTINE
```

```
                                        12200   ;**********
                                        12300   ;CHAINN--   THIS PORTION IS THE COMMON RETURN
                                        12400   ;           FOR ALL THREE CLASSES OF TESTS.
                                        12500   ;
                                        12600   ;       1--IF AN ERROR OCCURED DURING AN I/O TEST THE
                                        12700   ;          OPERATOR CAN CAUSE THAT TEST TO BE LOOPED
                                        12800   ;          WITHOUT ANY FURTHER ERROR HALTS BY
                                        12900   ;          SETTING THE "SCOPE" BIT (#14) ON THE SR=1.
                                        13000   ;          RESETTING SR BIT 14 TO 0 WILL ALLOW THE
                                        13100   ;          ERROR HALT TO OCCUR AGAIN IF IT STILL EXISTS.
                                        13200   ;
                                        13300   ;       2--IF THE OPERATOR IS IN THE MAINTENANCE
                                        13400   ;          MODE (BIT 8 SET = 1 AT START UP TIME), THE
                                        13500   ;          SELECTED PROGRAM WILL LOOP CONTINUOUSLY
                                        13600   ;          IF SR BIT 11 IS SET=1.  IF BIT 11 IS = 0
                                        13700   ;          THEN THE PROGRAM WILL BE ADVANCED TO
                                        13800   ;          THE NEXT TEST IN IT'S CLASS IF BIT 8=0.
                                        13900   ;          AS LONG AS BIT 11 AND
                                        14000   ;          BIT 8 ARE 0, THE CLASS OF TESTS SELECTED
                                        14100   ;          WILL BE CONTINUOUSLY SEQUENCED THROUGH.
                                        14200   ;          IF BIT 11 IS 0 AND BIT 8=1, THEN THE CPU
                                        14300   ;          WILL HALT AT LOCATION SELHLT AND WAIT FOR THE
                                        14400   ;          NEXT TEST NUMBER TO BE SET IN THE
                                        14500   ;          SWITCH REGISTER.
                                        14600   ;**********
                                        14700
001412  032737  000001  000634          14800   CHAINN: BIT     #1,CNTLSW       ;CKECK IF TERMINAL CONTROL
001420  001401                          14900           BEQ     1$              ;BRANCH IF NOT
001422  104011                          15000           TTYCTL                  ;GO TO TERMINAL CONTROL
001424  005737  000644                  15100   1$:     TST     PRGID           ;TEST ERROR BIT IN PRGID
001430  100016                          15200           BPL     3$              ;BRANCH IF ERROR BIT NOT SET
001432  032777  040000  177254          15300           BIT     #SCOPSW,@SR     ;ERR, CHECK IF SCOPE OPTION ON
001440  001407                          15400           BEQ     2$              ;BRANCH IF NO SCOPING
001442  022737  177777  000642          15500           CMP     #-1,SCOPTR      ;YES, CHECK IF OK TO SCOPE THIS TEST
001450  001403                          15600           BEQ     2$              ;BRANCH IF NOT OK
001452  017716  177164                  15700           MOV     @SCOPTR,@SP     ;PUT ADDR OF SCOPE ENTRY INTO STACK
001456  000002                          15800           RTI                     ;GO TO SCOPE ENTRY IN TEST
001460  042737  100000  000644          15900   2$:     BIC     #BIT15,PRGID    ;CLEAR ERROR IND.  IN PRGID
001466  005737  000654                  16000   3$:     TST     LEVEL           ;CHECK LEVEL
001472  001405                          16100           BEQ     4$              ;BRANCH IF LEVEL=0
001474  032777  004000  177212          16200           BIT     #NITRSW,@SR     ;TEST LOOP SWITCH ON (=1)
001502  001405                          16300           BEQ     5$              ;BRANCH IF NO LOOP TEST
001504  000002                          16400           RTI                     ;GO BACK TO TEST
001506  005337  000660                  16500   4$:     DEC     ICTR            ;DECREMENT TEST ITERATION COUNT
001512  001407                          16600           BEQ     6$              ;BRANCH IF COUNT=0
001514  000002                          16700           RTI                     ;NOT ZERO, REPEAT TEST
001516  032777  000400  177170          16800   5$:     BIT     #BIT8,@SR       ;TEST IF SEQUENCE TEST (BIT8)
001524  001402                          16900           BEQ     6$              ;BRANCH TO NEXT TEST IF BIT8=0
001526  000137  001700                  17000           JMP     WAITF           ;GO WAIT FOR MORE INPUT
001532  022626                          17100   6$:     POPSP2                  ;POP 2 OFF STACK
001534  000240                          17200   CHAINY: NOP                     ;THIS FORMERLY WAS RESET
001536  005777  177152                  17300           TST     @SR             ;CHECK SR
001542  100003                          17400           BPL     1$              ;BRANCH IF NO HALT WANTED
001544  113700  000636                  17500           MOVB    RTNNO,R0        ;CURRENT TEST NUMBER TO R0
001550  000000                          17600           HALT                    ;HALT (NOT FOR TEST SELECTION)
001552  005737  000654                  17700   1$:     TST     LEVEL           ;TEST THE CURRENT LEVEL
```

```
  412  001556  001420                    17800            BEQ     3$              ;BRANCH IF 0
  413  001560  012737  000006  000004    17900            MOV     #6,MACHER       ;CLEAN UP
  414  001566  012706  000600            18000            MOV     #SPBOT,SP       ;SET UP STACK POINTER
  415  001572  104024                    18100            FORWD                   ;SET UP VALUES FOR NEXT TEST
  416  001574  022737  177777  000640    18200            CMP     #-1,NXTST       ;END OF I/O TESTS (=-1)
  417  001602  001004                    18300            BNE     2$              ;BRANCH IF NOT END
  418  001604  012737  005276  000640    18400            MOV     #AT0,NXTST      ;RESET NXTST TO FIRST I/O TEST
  419  001612  104024                    18500            FORWD                   ;SET UP VALUES FOR NEXT TEST
  420  001614  000177  177056            18600    2$:     JMP     @CURTST         ;GO TO TEST
  421  001620  022737  177777  000640    18700    3$:     CMP     #-1,NXTST       ;END OF I/O TESTS (=-1)
  422  001626  001012                    18800            BNE     NEXT            ;BRANCH IF NOT
  423  001630  032777  000400  177056    18900    SKIP:   BIT     #BIT8,@SR       ;TEST IF WANT TEST SELECTION RIGHT AWAY
  424  001636  001016                    19000            BNE     NEXT1           ;BRANCH IF NOT
  425  001640  052737  000200  000644    19100            BIS     #BIT7,PRGID     ;BYPASS SCOPING
  426  001646  012737  007402  000640    19200            MOV     #PT0,NXTST      ;PROD TESTING, GO TO PRINTER TESTS
  427  001654  012737  000006  000004    19300    NEXT:   MOV     #6,MACHER       ;CLEAN UP
  428  001662  012706  000600            19400            MOV     #SPBOT,SP       ;SET UP STACK POINTER
  429  001666  104024                    19500            FORWD                   ;SET UP NEXT TEST PARAMETERS
  430  001670  000177  177002            19600            JMP     @CURTST         ;GO TO ROUTINE
  431  001674  005237  000654            19700    NEXT1:  INC     LEVEL
  432                                     19800
  433                                     19900    ;**********
  434                                     20000    ;WAIT FOR FURTHER INSTRUCTIONS:
  435                                     20100    ;       -LOAD PROGRAM NUMBER INTO BITS 0-5 OF THE SR
  436                                     20200    ;       -SET SR BIT 11=1 TO LOOP ON SELECTED TEST
  437                                     20300    ;       -SET SR BIT 11=0 AND BIT 8=0 TO LOOP THROUGH
  438                                     20400    ;        SEQUENCE OF SELECTED TESTS.
  439                                     20500    ;       -SET SR BIT 11=0 AND BIT 8=1 TO HALT AGAIN AFTER
  440                                     20600    ;        EXECUTING TEST ONCE
  441                                     20700    ;**********
  442                                     20800
  443  001700  104006                    20900    WAITF:  CHALT                   ;OR TTYCTL IF START WAS AT 206
  444  001702  012737  000006  000004    21000            MOV     #6,MACHER       ;CLEAN UP
  445  001710  012706  000600            21100            MOV     #SPBOT,SP       ;SET UP STACK POINTER
  446  001714  017700  176774            21200            MOV     @SR,R0          ;GET CURRENT SW REG
  447  001720  042700  177700            21300            BIC     #177700,R0
  448  001724  020027  000037            21400            CMP     R0,#37          ;TEST IF PROG NO. IS I/O TEST
  449  001730  101403                    21500            BLOS    1$              ;BRANCH IF EQ OR LT 37. AN ECHO OR PRINTER
  450  001732  005037  000644            21600            CLR     PRGID           ;I/O TEST, CLEAR PRGID
  451  001736  000403                    21700            BR      2$
  452  001740  052737  000200  000644    21800    1$:     BIS     #BIT7,PRGID     ;BYPASS SCOPING
  453  001746  000241                    21900    2$:     CLC                     ;CLEAR C BIT
  454  001750  006100                    22000            ROL     R0              ;GET PROGRAM ADDRESS OUT OF
  455  001752  016037  002522  000640    22100            MOV     PRGTAB(R0),NXTST;PROGRAM ADDRESS TABLE
  456  001760  023727  000640  001700    22200            CMP     NXTST,#WAITF    ;TEST IF LEGAL TEST NO.
  457  001766  001744                    22300            BEQ     WAITF           ;BRANCH IF ILLEGAL
  458  001770  104024                    22400            FORWD                   ;SET UP TEST PARAMETERS
  459  001772  000177  176700            22500            JMP     @CURTST         ;GO TO TEST
```

```
                                        22700   ;**********
                                        22800   ;TTY1-- THIS SECTION IS USED WHEN THE DIAGNOSTIC IS BEING CONTROLLED BY
                                        22900   ;         THE CONSOLE TERMINAL.  IT IS EFFECTIVE ONLY WHEN THE DIAGNOSTIC
                                        23000   ;        STARTING ADDRESS IS 210 AND SR BIT 8 WAS SET AT START TIME.
                                        23100   ;        THE RESPONSE TO THE MESSAGE "SELECT TEST NO." MUST BE THE 2
                                        23200   ;        DIGIT OCTAL TEST NUMBER FOLLOWED BY ;
                                        23300   ;                "L" TO LOOP ON TEST
                                        23400   ;                "S" TO LOOP ON SEQUENCE
                                        23500   ;                "." TO EXECUTE TEST ONCE
                                        23600   ;        ALL SPACES WILL BE IGNORED. AN ILLEGAL INPUT WILL BE FLAGED BY A "?"
                                        23700   ;        AND THE RETYPING OF THE ABOVE MESSAGE.
                                        23800   ;**********
                                        23900
001776  022626                          24000   TTY1:   POPSP2                  ;POP 2 FROM STACK
002000  105777  176606                  24100           TSTB    @TKS            ;TEST IF ANY INPUT
002004  100013                          24200           BPL     1$              ;BRANCH IF NOT
002006  017705  176602                  24300           MOV     @TKB,R5         ;GET CHAR
002012  042705  177600                  24400           BIC     #177600,R5      ;MASK BITS
002016  020527  000177                  24500           CMP     R5,#177         ;CHECK IF RUBOUT
002022  001004                          24600           BNE     1$              ;BRANCH IF NOT
002024  042737  004400  000634          24700           BIC     #4400,CNTLSW    ;CLEAR LOOP BITS
002032  000413                          24800           BR      TTY1B
002034  032737  004000  000634          24900   1$:     BIT     #NITRSW,CNTLSW  ;CHECK IF LOOP ON TEST
002042  001401                          25000           BEQ     2$              ;BRANCH IF NO LOOP ON TEST
002044  000002                          25100           RTI                     ;LOOP ON TEST
002046  032737  000400  000634          25200   2$:     BIT     #BIT8,CNTLSW    ;TEST IF LOOP ON SEQUENCE
002054  001402                          25300           BEQ     TTY1B           ;BRANCH IF NO LOOP ON SEQUENCE
002056  000137  001534                  25400           JMP     CHAINY          ;CHAIN TO NEXT TEST
002062  012737  177777  000710          25500   TTY1B:  MOV     #-1,INCHK       ;STOP INPUT CHECKING
002070  012700  000036                  25600           MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
002074  104010                          25700           DELAY
002076  104007                          25800           TYPEM
002100  014324                          25900           MESG3                   ;TYPE MESSAGE
002102  005037  000710                  26000           CLR     INCHK           ;ALLOW INPUT CHECKING AGAIN
002106  104020                          26100   1$:     READ                    ;WAIT FOR INPUT
002110  023727  000700  000040          26200           CMP     TEMPCH,#40      ;TEST IF CHAR IS A SPACE
002116  001773                          26300           BEQ     1$              ;BRANCH IF YES
002120  012700  000036                  26400           MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
002124  104010                          26500           DELAY
002126  104017                          26600           PRNT                    ;READY?
002130  117777  176460  176462          26700           MOVB    @TKB,@TPB       ;ECHO CHAR
002136  004737  002460                  26800           JSR     PC,TESTC        ;CHECK IF CHAR IS OK
002142  000541                          26900           BR      8$              ;NO, ERROR
002144  010005                          27000           MOV     R0,R5           ;OK, PUT CHAR INTO R5
002146  006305                          27100           ASL     R5              ;SHIFT INTO POSITION 5-3
002150  006305                          27200           ASL     R5
002152  006305                          27300           ASL     R5
002154  104020                          27400   2$:     READ                    ;WAIT FOR NEXT CHAR
002156  023727  000700  000040          27500           CMP     TEMPCH,#40      ;CHECK IF A SPACE
002164  001773                          27600           BEQ     2$              ;BRANCH IF SPACE
002166  012700  000036                  27700           MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
002172  104010                          27800           DELAY
002174  104017                          27900           PRNT                    ;READY?
002176  117777  176412  176414          28000           MOVB    @TKB,@TPB       ;ECHO CHAR
002204  004737  002460                  28100           JSR     PC,TESTC        ;CHECK IF CHAR IS OK
002210  000516                          28200           BR      8$              ;ERROR IN CHAR
```

```
   517  002212  060005                 28300          ADD     R0,R5           ;OK,R5 NOW = OCTAL TEST NO.
   518  002214  104020                 28400    3$:   READ                    ;WAIT FOR TERMINATION CHARACTER
   519  002216  023727  000700  000040 28500          CMP     TEMPCH,#40      ;CHECK IF SPACE
   520  002224  001773                 28600          BEQ     3$              ;BRANCH IF SPACE
   521  002226  012700  000036         28700          MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
   522  002232  104010                 28800          DELAY
   523  002234  104017                 28900          PRNT                    ;READY?
   524  002236  117777  176352  176354 29000          MOVB    @TKB,@TPB       ;ECHO CHAR
   525  002244  012737  004001  000634 29100          MOV     #4001,CNTLSW    ;SET BITS 11 & 0
   526  002252  023727  000700  000114 29200          CMP     TEMPCH,#114     ;NO, IS IT AN "L" ?
   527  002260  001427                 29300          BEQ     5$              ;BRANCH IF YES
   528  002262  023727  000700  000154 29400          CMP     TEMPCH,#154     ;CHECK LOWER CASE
   529  002270  001423                 29500          BEQ     5$
   530  002272  023727  000700  000123 29600          CMP     TEMPCH,#123     ;NO, IS IT AN "S"
   531  002300  001414                 29700          BEQ     4$              ;BRANCH IF YES
   532  002302  023727  000700  000163 29800          CMP     TEMPCH,#163     ;CHECK LOWER CASE
   533  002310  001410                 29900          BEQ     4$
   534  002312  023727  000700  000056 30000          CMP     TEMPCH,#56      ;NO, IS IT A "." ?
   535  002320  001052                 30100          BNE     8$              ;NO, ERROR
   536  002322  012737  000001  000634 30200          MOV     #1,CNTLSW       ;YES SET ONLY BIT 0 IN CONTROL WD
   537  002330  000403                 30300          BR      5$
   538  002332  012737  000401  000634 30400    4$:   MOV     #401,CNTLSW     ;SET BITS 8 & 0
   539  002340  012737  000006  000004 30500    5$:   MOV     #6,MACHER       ;CLEAN UP
   540  002346  012706  000600         30600          MOV     #SPBOT,SP       ;INIT SP
   541  002352  020527  000040         30700          CMP     R5,#40          ;IS THIS AN I/O TEST
   542  002356  103033                 30800          BHIS    8$              ;BRANCH IF YES
   543  002360  020527  000030         30900          CMP     R5,#30          ;IS THIS AN OPTION TEST?
   544  002364  103007                 31000          BHIS    6$              ;SKIP IF YES
   545  002366  020527  000020         31100          CMP     R5,#20          ;IS THIS AN ECHO TEST
   546  002372  103404                 31200          BLO     6$              ;BRANCH IF NOT
   547  002374  012737  000001  000634 31300          MOV     #1,CNTLSW       ;FORCE ECHO TEST TO A SINGLE RUN
   548  002402  000402                 31400          BR      7$              ;LEAVE THIS TERMINAL AS CONSOLE
   549  002404  004737  003700         31500    6$:   JSR     PC,CONIT        ;RESET CONSOLE TERMINAL ADDRESS
   550  002410  052737  000200  000644 31600    7$:   BIS     #BIT7,PRGID     ;BYPASS SCOPING
   551  002416  000241                 31700          CLC                     ;CLEAR C BIT
   552  002420  006105                 31800          ROL     R5
   553  002422  016537  002522  000640 31900          MOV     PRGTAB(R5),NXTST;ADDR OF TEST TO NXTST
   554  002430  023727  000640  001700 32000          CMP     NXTST,#WAITF    ;CHECK IF TEST EXISTS
   555  002436  001403                 32100          BEQ     8$              ;BRANCH IF NOT
   556  002440  104024                 32200          FORWD                   ;SET UP TEST PARAMETERS
   557  002442  000177  176230         32300          JMP     @CURTST         ;GO TO TEST
   558  002446  104017                 32400    8$:   PRNT                    ;CHECK IF PRINTER IS READY
   559  002450  112777  000077  176142 32500          MOVB    #77,@TPB        ;SEND A "?"
   560  002456  000601                 32600          BR      TTY1B           ;TRY AGAIN
```

```
                                                32800   ;TESTC--CHECKS THAT THE INPUTTED CHARACTER IS BETWEEN 0 AND 7 INCLUSIVE
                                                32900
002460  023727  000700  000060                33000   TESTC:  CMP     TEMPCH,#60      ;CHECK IF NUMERIC AND EQ OR GT 0
002466  103001                                  33100           BHIS    1$              ;BRANCH IF OK
002470  000207                                  33200           RTS     PC              ;ERROR RETURN
002472  023727  000700  000067                33300   1$:     CMP     TEMPCH,#67      ;CHECK IF EQ OR LT 7
002500  101401                                  33400           BLOS    2$              ;BRANCH IF OK
002502  000207                                  33500           RTS     PC              ;ERROR RETURN
002504  062716  000002                          33600   2$:     ADD     #2,@SP          ;SET UP RETURN ADDRESS
002510  013700  000700                          33700           MOV     TEMPCH,R0       ;GET CHAR
002514  042700  177770                          33800           BIC     #177770,R0      ;SAVE ONLY THE DIGIT
002520  000207                                  33900           RTS     PC              ;NORMAL RETURN
```

```
  575  002522  007402                34100  PRGTAB: PT0                    ;DATA PATH TEST
  576  002524  007456                34200          PT1                    ;PRINTER CHARACTER TEST
  577  002526  007600                34300          PT2                    ;NON-PRINTING CHARACTER TEST
  578  002530  010164                34400          PT3                    ;CARRIAGE RETURN TEST
  579  002532  010304                34500          PT4                    ;MULTIPLE LINE FEED TEST
  580  002534  010462                34600          PT5                    ;SINGLE LINE FEED TEST
  581  002536  010666                34700          PT6                    ;BACKSPACE TEST
  582  002540  011054                34800          PT7                    ;OVERPRINT TEST
  583  002542  011266                34900          PT10                   ;PRINTING FREQUENCY SWEEP TEST
  584  002544  011424                35000          PT11                   ;RIBBON FEED TEST
  585  002546  011456                35100          PT12                   ;PRINTER BELL TEST
  586  002550  001700                35200          WAITF                  ;SPARE
  587  002552  001700                35300          WAITF                  ;SPARE
  588  002554  001700                35400          WAITF                  ;SPARE
  589  002556  001700                35500          WAITF                  ;SPARE
  590  002560  011546                35600          PT17                   ;LIFE TEST
  591  002562  012116                35700          E020                   ;CHARACTER ECHO TEST
  592  002564  012166                35800          E021                   ;LINE ECHO TEST, FAST RATE
  593  002566  012224                35900          E022                   ;LINE ECHO TEST, SLOW RATE
  594  002570  012476                36000          E023                   ;CHARACTER/CODE ECHO TEST
  595  002572  013020                36100          E024                   ;SELECTIVE PATTERN ECHO TEST
  596  002574  013566                36200          E025                   ;BELL ECHO TEST
  597  002576  001700                36300          WAITF                  ;SPARE
  598  002600  001700                36400          WAITF                  ;SPARE
  599  002602  001700                36500          WAITF          ;SPARE
  600  002604  001700                36600          WAITF          ;SPARE
  601  002606  001700                36700          WAITF          ;SPARE
  602  002610  001700                36800          WAITF          ;PRARE
  603  002612  001700                36900          WAITF          ;SPARE
  604  002614  001700                37000          WAITF          ;SPARE
  605  002616  001700                37100          WAITF                  ;SPARE
  606  002620  001700                37200          WAITF                  ;SPARE
  607  002622  005276                37300          AT0                    ;I/O TEST NO.  40
  608  002624  005330                37400          AT1                    ;I/O TEST NO.  41
  609  002626  005362                37500          AT2                    ;I/O TEST NO.  42
  610  002630  005414                37600          AT3                    ;I/O TEST NO.  43
  611  002632  005446                37700          AT4                    ;I/O TEST NO.  44
  612  002634  005536                37800          AT5                    ;I/O TEST NO.  45
  613  002636  005614                37900          AT6                    ;I/O TEST NO.  46
  614  002640  005704                38000          AT7                    ;I/O TEST NO.  47
  615  002642  005754                38100          AT10                   ;I/O TEST NO.  50
  616  002644  006012                38200          AT11                   ;I/O TEST NO.  51
  617  002646  006052                38300          AT12                   ;I/O TEST NO.  52
  618  002650  006126                38400          AT13                   ;I/O TEST NO.  53
  619  002652  006206                38500          AT14                   ;I/O TEST NO.  54
  620  002654  006272                38600          AT15                   ;I/O TEST NO.  55
  621  002656  006372                38700          AT16                   ;I/O TEST NO.  56
  622  002660  006440                38800          AT17                   ;I/O TEST NO.  57
  623  002662  006510                38900          AT20                   ;I/O TEST NO.  60
  624  002664  006602                39000          AT21                   ;I/O TEST NO.  61
  625  002666  006702                39100          AT22                   ;I/O TEST NO.  62
  626  002670  007010                39200          AT23                   ;I/O TEST NO.  63
  627  002672  007122                39300          AT24                   ;LSI TEST NO.  64
  628  002674  007222                39400          AT25                   ;LSI TEST NO.  65
  629  002676  007300                39500          AT26                   ;LSI TEST NO.  66
  630  002700  001700                39600          WAITF                  ;SPARE
```

```
   631  002702  001700                39700            WAITF                   ;SPARE
   632  002704  001700                39800            WAITF                   ;SPARE
   633  002706  001700                39900            WAITF                   ;SPARE
   634  002710  001700                40000            WAITF                   ;SPARE
   635  002712  001700                40100            WAITF                   ;SPARE
   636  002714  001700                40200            WAITF                   ;SPARE
   637  002716  001700                40300            WAITF                   ;SPARE
   638  002720  001700                40400            WAITF                   ;SPARE
   639                                 40500
   640                                 40600    ;**********
   641                                 40700    ;EMTINT ----------SERVICE ROUTINE FOR TRAPS THROUGH
   642                                 40800    ;                  LOCATION 30.
   643                                 40900    ;**********
   644                                 41000
   645  002722  011646                41100    EMTINT: MOV     @SP,-(SP)       ;PUSH STACKED PC TO GET A WORK COPY.    (Q)
   646  002724  162716  000002        41200            SUB     #2,@SP          ;SUB 2 TO POINT TO CALLING TRAP INSTR.
   647  002730  017616  000000        41300            MOV     @(SP),@SP       ;PLACE TRAP INSTR INTO THIS STACK WORK AREA.
   648  002734  121627  000035        41400            CMPB    @SP,#35         ;EXAMINE ITS RIGHT SIDE.   (Q)
   649  002740  101402                41500            BLOS    2$              ;BRANCH IF WITHIN RANGE OF ESTABLISHED TABLE.
   650  002742  000000                41600    1$:     HALT                    ;ELSE HALT.
   651  002744  000776                41700            BR      1$
   652  002746  006116                41800    2$:     ROL     @SP             ;MULT INSTR BY 2 TO GET WORD DISPLACEMENT.
   653  002750  042716  177001        41900            BIC     #177001,@SP     ;STRIP OFF OP CODE AND LS BIT.
   654  002754  062716  002776        42000            ADD     #EMTTAB,@SP     ;ADD IN STARTING ADDRESS OF TABLE.
   655  002760  017616  000000        42100            MOV     @(SP),@SP       ;FROM TABLE GET OUT DESIRED POINTER.
   656  002764  005046                42200            CLR     -(SP)           ;PUSH A ZERO PSW.
   657  002766  012746  002774        42300            MOV     #3$,-(SP)       ;PUSH A PC = TO #3$ OF THIS ROUTINE.
   658  002772  000002                42400            RTI                     ;DO RTI (POP-POP) TO ESTABLISH THE ZERO PSW.
   659  002774  000136                42500    3$:     JMP     @(SP)+          ;JMP TO ROUTINE LEAVING STACK AS FOUND.
   660                                 42600
   661  002776  003076                42700    EMTTAB: TYP                     ;MESSAGE OUTPUT ROUTINE
   662  003000  003310                42800            ERR                     ;I/O TEST ERROR ROUTINE
   663  003002  003336                42900            EHLT                    ;UNCONDITIONAL HALT
   664  003004  003346                43000            STLSRV                  ;KEYBOARD VECTOR/PRIORITY SETUP
   665  003006  003376                43100            STLSPV                  ;PRINTER VECTOR/PRIORITY SETUP
   666  003010  001412                43200            CHAINN                  ;COMMON TEST EXIT
   667  003012  000720                43300            CHLT                    ;SR BIT 15 HALT
   668  003014  003154                43400            TYPM                    ;MESSAGE OUTPUT ROUTINE, MULTI DEVICES
   669  003016  003426                43500            DLY                     ,DELAY ROUTINE
   670  003020  001776                43600            TTY1                    ;CONSOLE TERMINAL CONTROL
   671  003022  003204                43700            $CRLF                   ;CARRIAGE RETURN-LINE FEED TO ALL DL11'S
   672  003024  003132                43800            $SCRLF                  ;CARRIAGE RETURN-LINE FEED TO CONSOLE
   673  003026  003206                43900            $LF                     ;LINE FEED ONLY (TO ALL)
   674  003030  004314                44000            $PRTC                   ;PRINT CHAR
   675  003032  003226                44100            $PRHDR                  ;PRINT TEST HEADER
   676  003034  004304                44200            $PRNT                   ;PRINTER READY
   677  003036  004102                44300            $READ                   ;READ CHAR
   678  003040  003630                44400            $AREAD                  ;I/O TEST READ ROUTINE
   679  003042  003216                44500            $CR                     ;CARRIAGE RETURN ONLY (TO ALL)
   680  003044  003776                44600            $BTASC                  ;BINARY TO ASCII CONVERSION
   681  003046  003552                44700            $FORWD                  ;FORWARD ROUTINE ( BETWEEN TESTS )
   682  003050  004174                44800            $READC                  ;READ CONSOLE KYBD ONLY
   683  003052  003072                44900            SPARET                  ;SPARE EMT
   684  003054  003072                45000            SPARET                  ;SPARE EMT
   685  003056  003072                45100            SPARET                  ;SPARE EMT
   686  003060  003072                45200            SPARET                  ;SPARE EMT
```

```
 687  003062  003072                45300           SPARET                  ;SPARE EMT
 688  003064  003072                45400           SPARET                  ;SPARE EMT
 689  003066  003072                45500           SPARET                  ;SPARE EMT
 690  003070  003072                45600           SPARET                  ;SPARE EMT
 691  003072  000000                45700   SPARET: HALT                    ;HALT IF TRAP TO UNDEFINED
 692  003074  000776                45800           BR      SPARET          ;EMT IS ATTEMPTED.
 693                                45900
 694                                46000
 695                                46100
 696                                46200
 697                                46300           .SBTTL  COMMON ROUTINES USED BY LA36 TESTS
 698                                46400
 699                                46500   ;**********
 700                                46600   ;
 701                                46700   ;THIS SECTION CONTAINS MOST ROUTINES CALLED BY
 702                                46800   ;THE VARIOUS TESTS EITHER BY TRAPPING THROUGH LOCATION
 703                                46900   ;30 OR BY SUBROUTINE CALLS (JSR PC,***)
 704                                47000   ;
 705                                47100   ;**********
 706                                47200   ;
 707                                47300   ;TYPE-- A COMMON ROUTINE USED TO TYPE MESSAGES ON THE
 708                                47400   ;       CONSOLE TERMINAL ONLY.  THE NULL CHARACTER TERMINATES
 709                                47500   ;       THE MESSAGE.  CALLED THROUGH AN EMT TRAP.
 710                                47600   ;       CALLING SEQUENCE
 711                                47700   ;       TYPE
 712                                47800   ;       MESG    ;ADDRESS OF MESSAGE
 713                                47900   ;
 714                                48000   ;**********
 715                                48100
 716  003076  011601                48200   TYP:    MOV     (SP),R1         ;GET POINTER TO ADDR.  OF MESG.
 717  003100  062716  000002        48300           ADD     #2,@SP
 718  003104  011101                48400           MOV     (R1),R1         ;ADDR.  OF MESG TO R1
 719  003106  112100                48500   1$:     MOVB    (R1)+,R0        ;GET CHAR
 720  003110  100402                48600           BMI     2$              ;BRANCH IF WANT AUTO CR-LF
 721  003112  001003                48700           BNE     3$              ;PRINT CHAR IF NOT NULL
 722  003114  000002                48800           RTI                     ;EXIT IF NULL CHAR
 723  003116  104013                48900   2$:     $CRLF                   ;YES, SEND CR-LF
 724  003120  000772                49000           BR      1$              ;GET NEXT CHAR
 725  003122  104017                49100   3$:     PRNT                    ;PRINTER READY?
 726  003124  110077  175470        49200           MOVB    R0,@TPB         ;LOAD PRINTER BUFFER WITH CHAR
 727  003130  000766                49300           BR      1$              ;GO GET NEXT CHAR
 728                                49400
 729  003132  104017                49500   $CRLF:  PRNT                    ;PRINTER READY?
 730  003134  112777  000015  175456 49600           MOVB    #15,@TPB        ;SEND CR
 731  003142  104017                49700           PRNT                    ;PRINTER READY?
 732  003144  112777  000012  175446 49800           MOVB    #12,@TPB        ;SEND LF
 733  003152  000002                49900           RTI                     ;RETURN TO CALLER
```

```
                                    50100   ;XXXXXXXXXX
                                    50200   ;
                                    50300   ;TYPM---MULTI TYPE-A COMMON ROUTINE TO OUTPUT
                                    50400   ;       A MESSAGE ON ALL DL11S IF THE MULTI TEST
                                    50500   ;       SWITCH (BIT 13) IS SET. THIS ROUTINE IS USED BY
                                    50600   ;       THE PRINTER TESTS TO TYPE HEADINGS.  IF A UNIT
                                    50700   ;       IS NOT READY, THE CHARACTER WILL NOT BE TYPED.
                                    50800   ;
                                    50900   ;XXXXXXXXXX
                                    51000
003154  011601                      51100   TYPM:   MOV     (SP),R1         ;GET POINTER TO ADDR OF MESG
003156  062716  000002              51200           ADD     #2,@SP
003162  011101                      51300           MOV     (R1),R1         ;ADDR OF MESG TO R1
003164  112100                      51400   1$:     MOVB    (R1)+,R0        ;GET CHAR
003166  100402                      51500           BMI     2$              ;BRANCH IF WANT AUTO CR-LF
003170  001003                      51600           BNE     3$              ;CONTINUE IF NOT NULL
003172  000002                      51700           RTI                     ;RETURN
003174  104012                      51800   2$:     CRLF                    ;YES, SEND CR-LF
003176  000772                      51900           BR      1$              ;NEXT CHAR
003200  104015                      52000   3$:     PRINTC                  ;PRINT CHAR
003202  000770                      52100           BR      1$              ;GO GET NEXT CHAR.
                                    52200
003204  104022                      52300   $CRLF:  CR                      ;SEND CR
003206  012700  000012              52400   $LF:    MOV     #12,R0          ;SET LF CHAR
003212  104015                      52500           PRINTC                  ;SEND IT
003214  000002                      52600           RTI                     ;RETURN TO CALLER
                                    52700
003216  012700  000015              52800   $CR:    MOV     #15,R0          ;SET CR CHAR
003222  104015                      52900           PRINTC                  ;SEND IT
003224  000002                      53000           RTI                     ;RETURN
                                    53100
                                    53200   ;**********
                                    53300   ;
                                    53400   ;ROUTINE TO PRINT TEST HEADER
                                    53500   ;
                                    53600   ;**********
                                    53700
003226  012700  000000              53800   $PRHDR: MOV     #0,R0           ;TRANSMIT
003232  104015                      53900           PRINTC                  ,NUL CODE.
003234  104007                      54000           TYPEM                   ;PRINT MESSAGE
003236  014076                      54100           HDRMSG
003240  013700  000636              54200           MOV     RTNNO,R0        ;GET TEST NUMBER
003244  006200                      54300           ASR     R0              ;GET FIRST DIGIT
003246  006200                      54400           ASR     R0
003250  006200                      54500           ASR     R0
003252  042700  177770              54600           BIC     #177770,R0      ;MASK FIRST DIGIT
003256  062700  000060              54700           ADD     #60,R0          ;MAKE ASCII
003262  104015                      54800           PRINTC                  ;PRINT DIGIT
003264  013700  000636              54900           MOV     RTNNO,R0        ;GET TEST NUMBER AGAIN
003270  042700  177770              55000           BIC     #177770,R0      ;MASK LAST DIGIT
003274  062700  000060              55100           ADD     #60,R0          ;MAKE ASCII
003300  104015                      55200           PRINTC                  ;PRINT DIGIT
003302  104012                      55300           CRLF                    ;CR-LF
003304  104014                      55400           LF                      ;BLANK LINE
003306  000002                      55500           RTI                     ;RETURN
```

```
                                                55700   ;**********
                                                55800   ;
                                                55900   ;ERRA-- COMMON ERROR RETURN FROM I/O TESTS.  HALTS
                                                56000   ;       WITH ADDRESS OF ERROR IN R0.  TO CONTINUE
                                                56100   ;       ON SAME TEST BUT NOT HALTING ON ERROR,
                                                56200   ;       SET THE SCOPE BIT (14) = 1 AND PRESS CONTINUE
                                                56300   ;
                                                56400   ;**********
                                                56500
003310  032777  040000  175376                  56600   ERR:    BIT     #SCOPSW,@SR     ;CHECK SCOPE SWITCH
003316  001404                                  56700           BEQ     1$              ;BRANCH IF NO SCOPE
003320  005737  000644                          56800           TST     PRGID           ;SCOPING WANTED, FIRST ERROR?
003324  100001                                  56900           BPL     1$              ;BRANCH AND HALT ON FIRST ERROR
003326  000002                                  57000           RTI                     ;SCOPE EXIT
003330  052737  100000  000644                  57100   1$:     BIS     #BIT15,PRGID    ;SET ERROR INDICATOR
003336  011600                                  57200   EHLT:   MOV     @SP,R0
003340  005740                                  57300           TST     -(R0)           ;ADDRESS OF CALL INTO R0
003342  000000                                  57400           HALT
003344  000002                                  57500   ERRHLT: RTI                     ;RETURN TO TEST FOLLOWING CALL
                                                57600
                                                57700   ;**********
                                                57800   ;
                                                57900   ;STLSRV--- THIS ROUTINE SETS UP KEYBOARD INTERRUPT
                                                58000   ;         VECTOR AND PRIOITY.  CALLING SEQUENCE
                                                58100   ;
                                                58200   ;       STRDRV
                                                58300   ;       AT20C   ;LOCATION OF NEW INTERRUPT VECTOR
                                                58400   ;
                                                58500   ;**********
                                                58600
003346  017637  000000  003366                  58700   STLSRV: MOV     @(SP),STPRA+2   ;SET RETURN ADR AND VECTOR
003354  062716  000002                          58800           ADD     #2,@SP
003360  013701  000622                          58900           MOV     TKVTR,R1
003364  012721  000000                          59000   STPRA:  MOV     #0,(R1)+
003370  013721  000624                          59100           MOV     TKLVL,(R1)+
003374  000002                                  59200           RTI
                                                59300
                                                59400   ;**********
                                                59500   ;
                                                59600   ;STLSPV-- THIS ROUTINE SETS UP PRINTER INTERRUPT
                                                59700   ;
                                                59800   ;       VECTOR AND PRIORITY CALLING SEQUENCE
                                                59900   ;       STPCHV
                                                60000   ;       AT35E   ;LOCATION OF NEW INTERRUPT VECTOR
                                                60100   ;
                                                60200   ;**********
                                                60300
003376  017637  000000  003416                  60400   STLSPV: MOV     @(SP),STPPA+2   ;SET RETURN ADR AND VECTOR
003404  062716  000002                          60500           ADD     #2,@SP
003410  013701  000626                          60600           MOV     TPVTR,R1
003414  012721  000000                          60700   STPPA:  MOV     #0,(R1)+
003420  013721  000630                          60800           MOV     TPLVL,(R1)+
003424  000002                                  60900           RTI                     ;RETURN TO CALLER
```

```
                                61100   ;**********
                                61200   ;
                                61300   ;DELAY--A COMMON ROUTINE TO DELAY PROCESSING
                                61400   ;       A GIVEN NUMBER OF MSEC.
                                61500   ;       CALLING SEQUENCE;
                                61600   ;       MOV   #5,R0   ;R0 CONTAINS THE NUMBER OF MSEC DELAY DESIRED
                                61700   ;       DELAY
                                61800   ;
                                61900   ;       THE DELAY IS EFFECTED BY THE EXECUTION OF THE LOOP;
                                62000   ;          1$:  DEC  R1
                                62100   ;               BNE  1$
                                62200   ;
                                62300   ;       SINCE THE EXECUTION TIMES OF THE PDP11 LINE DOES VARY FROM
                                62400   ;       MACHINE TO MACHINE, THE VALUE AT SYMBOLIC LOCATION
                                62500   ;       "TIMER" MUST BE CHANGED TO THE APPROPRIATE VALUE AS SHOWN BELOW
                                62600   ;       BEFORE STARTING THE DIAGNOSTIC. "TIMER" IS INITIALIZED
                                62700   ;       FOR AN 11/05,11/10(=251).
                                62800   ;
                                62900   ;MACHINE        05&10   35&40   15&20   LSI&03          11/45 & 11/70
                                63000   ;                                                BIPOLAR       MOS       CORE
                                63100   ;
                                63200   ;LOOP:  DEC R1   3.4     .99     2.3                     .30          .51     .90
                                63300   ;       BNE LOOP 2.5     1.76    2.6                     .60          .98    1.13
                                63400   ;  TIME=         5.9USEC 2.75    4.9     7.7     .90USEC          1.49USEC
                                63500   ; SET TIMER      251     554     314     202     2127             1237    755
                                63600   ;
                                63700   ;XXXXXXXXXX
                                63800
000672                          63900   DLY:    MOV     R1,-(SP)        ;SAVE R1
003426  010146                  64000   1$:     MOV     TIMER,R1        ;MOV 1 MSEC LOOP CNT TO R1
003430  013701  000672          64100   2$:     DEC     R1              ;DECREMENT COUNT
003434  005301                  64200           BNE     2$              ;BRANCH IF NOT ZERO
003436  001376                  64300           DEC     R0              ;DEC NO. OF MSEC DELAY
003440  005300                  64400           BNE     1$              ;DELAY AGAIN IF NOT ZERO
003442  001372                  64500           MOV     (SP)+,R1        ;ALL DONE RESTORE R1
003444  012601                  64600           RTI
003446  000002
```

```
                                        64800   ;**********
                                        64900   ;
                                        65000   ;PFAIL--POWER FAIL ROUTINE
                                        65100   ;       SAVE ALL REGISTERS AND SET RESTART ADDRESS
                                        65200   ;       INTO LOCATION 24
                                        65300   ;
                                        65400   ;RESTART--POWER FAIL RECOVERY
                                        65500   ;        RESTORE ALL REGISTERS AND GO TO START
                                        65600   ;
                                        65700   ;**********
                                        65800
003450  010046                          65900   PFAIL:  MOV     R0,-(SP)
003452  010146                          66000           MOV     R1,-(SP)
003454  010246                          66100           MOV     R2,-(SP)
003456  010346                          66200           MOV     R3,-(SP)
003460  010446                          66300           MOV     R4,-(SP)
003462  010546                          66400           MOV     R5,-(SP)
003464  013746  000024                  66500           MOV     24,-(SP)
003470  010637  003504                  66600           MOV     SP,SAVR6        ;SAVE STACK POSITION
003474  012737  003506  000024          66700           MOV     #RESTRT,24      ;STORE RESTART ADDRESS
003502  000000                          66800           HALT
003504  000000                          66900   SAVR6:  .WORD   0
003506  104007                          67000   RESTRT: TYPEM
003510  003542                          67100           1$
003512  013706  003504                  67200           MOV     SAVR6,SP        ;RESTORE STACK POINTER
003516  012637  000024                  67300           MOV     (SP)+,24        ;RESTORE PFAIL ADDRESS
003522  012605                          67400           MOV     (SP)+,R5
003524  012604                          67500           MOV     (SP)+,R4
003526  012603                          67600           MOV     (SP)+,R3
003530  012602                          67700           MOV     (SP)+,R2
003532  012601                          67800           MOV     (SP)+,R1
003534  012600                          67900           MOV     (SP)+,R0
003536  000137  001010                  68000           JMP     START
                                        68100
003542  050200  053517  051105         68200   1$:     .ASCIZ  <ACRLF>/POWER/<ACRLF>
003550  000200
                                        68300           EVEN
```

```
                                                68500   ;**********
                                                68600   ;
                                                68700   ;FORWARD--THIS ROUTINE TRANSFERS THE 2 OR 4 ARGUMENTS
                                                68800   ;         FROM THE TEST ROUTINE.  THEY ARE;
                                                68900   ;
                                                69000   ;         1- ROUTINE NUMBER
                                                69100   ;         2- ADDRESS OF NEXT TEST
                                                69200   ;         3- ITERATION COUNT (I/O TESTS ONLY)
                                                69300   ;         4- SCOPE ENTRY ADDRESS (I/O TESTS ONLY)
                                                69400   ;
                                                69500   ;**********
                                                69600
003552  013705  000640          000640  69700  $FORWD: MOV     NXTST,R5        ;ADDR OF NEXT TEST TO R5
003556  012537  000636          000636  69800          MOV     (R5)+,RTNNO     ;GET NUMBER OF NEXT TEST
003562  012537  000640          000640  69900          MOV     (R5)+,NXTST     ;GET ADDR OF FOLLOWING TEST
003566  105737  000644          000644  70000          TSTB    PRGID           ;CHECK IF I/O TEST
003572  100407                  000576  70100          BMI     FORWDB          ;SKIP THE FETCH OF ITER CNT AND SCOPE
003574  012537  000660          000660  70200          MOV     (R5)+,ICTR      ;GET ITERATION COUNT
003600  012537  000642          000642  70300          MOV     (R5)+,SCOPTR    ;GET SCOPE ENTRY POINT
003604  010537  000676          000676  70400  FORWDA: MOV     R5,CURTST       ;ENTRY POINT TO TEST IN CUR TST
003610  000002                  000002  70500          RTI                     ;EXIT
003612  012737  177777  000642  177777  70600  FORWDB: MOV     #-1,SCOPTR      ;FORCE NO SCOPE
003620  012737  000001  000660  000001  70700          MOV     #1,ICTR         ;FORCE INTERATION COUNT OF 1
003626  000766                  003604  70800          BR      FORWDA
```

```
                                                71000   ;**********
                                                71100   ;
                                                71200   ;AREAD--A ROUTINE WHICH, THROUGH THE FACILITY OF
                                                71300   ;       THE MAINTENANCE BIT, OUTPUTS TO THE
                                                71400   ;       PRINTER BUFFER AND READS THE KEYBOARD
                                                71500   ;       STATUS DONE.  IF THE DONE IS NOT SET
                                                71600   ;       WITHIN 200 MSEC, THE CPU WILL HALT WITH
                                                71700   ;       THE LOCATION OF THE ERROR IN R0.  PRESS
                                                71800   ;       CONTINUE TO CONTINUE WITH TESTS.
                                                71900   ;
                                                72000   ;**********
                                                72100
003630  012737  000200  000664                 72200   $AREAD: MOV     #200,BRCTR      ;SET UP 200 MSEC DELAY
003636  052777  000004  174752                 72300           BIS     #4,@TPS         ;SET MAINTENANCE BIT
003644  005077  174750                         72400           CLR     @TPB            ;LOAD PRINTER BUFFER
003650  105777  174736                         72500   1$:     TSTB    @TKS            ;CHECK DONE BIT
003654  100410                                 72600           BMI     2$              ;BRANCH IF DONE
003656  012700  000001                         72700           MOV     #1,R0           ;ONE TO R0
003662  104010                                 72800           DELAY                   ;DELAY 1 MSEC.
003664  005337  000664                         72900           DEC     BRCTR           ;200 MSEC OVER
003670  001367                                 73000           BNE     1$              ;BRANCH IF NO
003672  104002                                 73100           EHALT
003674  000755                                 73200           BR      $AREAD          ;TRY AGAIN
003676  000002                                 73300   2$:     RTI                     ;RETURN TO TEST
                                               73400
                                               73500   ;**********
                                               73600   ;
                                               73700   ;CONIT--THIS ROUTINE SETS UP THE DEVICE ADDRESSES
                                               73800   ;       AND INTERRUPT VECTORS FOR THE CONSOLE
                                               73900   ;       TERMINAL.
                                               74000   ;
                                               74100   ;**********
                                               74200
003700  013700  000602                         74300   CONIT:  MOV     CONADD,R0       ;CONSOLE KEYBOARD STATUS ADDR TO R0
003704  010037  000612                         74400   CONSET: MOV     R0,TKS          ;KEYBOARD STATUS ADDRESS (777560) TO TKS
003710  005720                                 74500           TST     (R0)+           ;INCREMENT R0 BY TWO
003712  010037  000614                         74600           MOV     R0,TKB          ;KEYBOARD DATA ADDR (777562) TO TKB
003716  005720                                 74700           TST     (R0)+           ;INCREMENT R0 BY TWO
003720  013737  000616  003772                 74800           MOV     TPS,TPSS        ;SAVE TPS OF LAST TERMINAL
003726  010037  000616                         74900           MOV     R0,TPS          ;PRINTER STATUS ADDR(777564) TO TPS
003732  005720                                 75000           TST     (R0)+           ;INCREMENT R0 BY TWO
003734  013737  000620  003774                 75100           MOV     TPB,TPBS        ;SAVE TPB OF LAST TERMINAL
003742  010037  000620                         75200           MOV     R0,TPB          ;PRINTER DATA ADDR (777566) TO TPB
003746  013737  000604  000622                 75300           MOV     CONVEC,TKVTR    ;KEYBOARD INTERRUPT VECTOR (60) TO TKVTR
003754  013737  000604  000626                 75400           MOV     CONVEC,TPVTR
003762  062737  000004  000626                 75500           ADD     #4,TPVTR        ;PRINTER INTERRUPT VECTOR (64) TO TPVTR
003770  000207                                 75600           RTS     PC
                                               75700
003772  000000                                 75800   TPSS:   .WORD   0               ;LAST TERM STATUS REG ADR
003774  000000                                 75900   TPBS:   .WORD   0               ;LAST TERM BUFFER REG ADR
```

```
                                                76100   ;**********
                                                76200   ;
                                                76300   ;BINARY TO ASC11 CONVERSION (1 TO 5 ASC11 CHARACTERS)
                                                76400   ;CALLING SEQUENCE
                                                76500   ;       MOV     ADDRESS OF LOC TO STORE FIRST ASC11 CHAR.  INTO R0
                                                76600   ;       MOV     BINARY NUMBER TO BE CONVERTED INTO R1
                                                76700   ;       MOV     NUMBER TO BE CONVERTED AS A POWER OF TEN INTO R2
                                                76800   ;       BTOASC
                                                76900   ;
                                                77000   ;**********
                                                77100
1006  003776  010237  004062                   77200   $BTASC: MOV     R2,CNVCTR       ;SAVE TEN POWER
1007  004002  006302                           77300           ASL     R2              ;R2*2
1008  004004  062702  004070                   77400           ADD     #ADTENP,R2      ;CALCULATE ADDRESS OF
1009                                            77500                                   ;STARTING TEN POWER
1010  004010  014237  004066                   77600   1$:     MOV     -(R2),TENPWR    ;POWER OF TEN VALUE TO TEN PWR
1011  004014  005037  004064                   77700           CLR     DIGIT           ;CLEAR CURRENT DIGIT
1012  004020  163701  004066                   77800   2$:     SUB     TENPWR,R1       ;SUBTRACT TEN POWER FROM BINARY VALUE
1013  004024  103403                           77900           BCS     3$              ;BRANCH IF END
1014  004026  005237  004064                   78000           INC     DIGIT
1015  004032  000772                           78100           BR      2$
1016  004034  063701  004066                   78200   3$:     ADD     TENPWR,R1       ;RESTORE SUBTRACTED VALUE
1017  004040  062737  000060  004064           78300           ADD     #60,DIGIT       ;CONVERT (DIGIT) TO ASCII
1018  004046  113720  004064                   78400           MOVB    DIGIT,(R0)+     ;PUT ASCII CHAR INTO USER BUFFER
1019  004052  005337  004062                   78500           DEC     CNVCTR          ;FINISHED ALL CHARS. CALLED FOR
1020  004056  001354                           78600           BNE     1$              ;BRANCH IF NOT FINISHED
1021  004060  000002                           78700           RTI                     ;YES, EXIT
1022  004062  000000                           78800   CNVCTR: .WORD   0               ;CONVERSION CHARACTER COUNT
1023  004064  000000                           78900   DIGIT:  .WORD   0               ;CONVERTED CHARACTER
1024  004066  000000                           79000   TENPWR: .WORD   0               ;CURRENT TEN POWER
1025  004070  000001  000012  000144           79100   ADTENP: .WORD   1.,10.,100.,1000.,10000.
      004076  001750  023420
```

```
1027                                    79300   ;XXXXXXXXXX
1028                                    79400   ;
1029                                    79500   ;READ-- A COMMON ROUTINE WHICH CHECKS THE KEYBOARD
1030                                    79600   ;       DONE FLAG & SETS A FLAG INDICATING CHAR PARITY
1031                                    79700   ;
1032                                    79800   ;XXXXXXXXXX
1033                                    79900
1034  004102  004737  003700              80000   $READ:  JSR     PC,CONIT        ;RESET CONSOLE ADR AND VECTORS
1035  004106  005737  000656              80100           TST     DLCNT           ;CHECK IF MULTI DL11'S AVAILABLE
1036  004112  001430                      80200           BEQ     $READC          ;NONE, WAIT FOR CONSOLE INPUT
1037  004114  013737  000656  000666      80300   1$:     MOV     DLCNT,COUNT3    ;SET DL11 COUNT
1038  004122  013737  000632  000670      80400           MOV     FSTDL,XCSR      ;ADDRESS OF FIRST DL11 INTO XCSR
1039  004130  105777  174534              80500   2$:     TSTB    @XCSR           ;TEST IF ANY INPUT
1040  004134  100005                      80600           BPL     3$              ;CONTINUE IF NO INPUT
1041  004136  013700  000670              80700           MOV     XCSR,R0         ;SET THIS DL11 AS CONSOLE
1042  004142  004737  003704              80800           JSR     PC,CONSET
1043  004146  000415                      80900           BR      READ1           ;READ CHAR AND RETURN
1044  004150  005337  000666              81000   3$:     DEC     COUNT3          ;DECREMENT DL11 COUNT
1045  004154  001404                      81100           BEQ     4$              ;TEST CONSOLE WHEN DONE DL11'S
1046  004156  062737  000010  000670      81200           ADD     #10,XCSR        ;NEXT DL11 ADDRESS
1047  004164  000761                      81300           BR      2$              ;CONTINUE
1048  004166  105777  174420              81400   4$:     TSTB    @TKS            ;CHECK CONSOLE
1049  004172  100350                      81500           BPL     1$              ;WAIT, NO INPUT
1050  004174  105777  174412              81600   $READC: TSTB    @TKS            ;CHECK KEYBOARD DONE FLAG
1051  004200  100375                      81700           BPL     $READC          ;BRANCH IF NOT SET
1052  004202  117737  174406  000700      81800   READ1:  MOVB    @TKB,TEMPCH     ;SAVE CHARACTER
1053  004210  113737  000700  000704      81900           MOVB    TEMPCH,PCHAR    ;SAVE CODE WITH PARITY BIT
1054  004216  042737  177400  000704      82000           BIC     #177400,PCHAR   ;MASK UNWANTED BITS
1055  004224  113737  000700  000703      82100           MOVB    TEMPCH,PARITY+1 ;SAVE CHAR WITH PARITY BIT
1056  004232  042737  177600  000700      82200           BIC     #177600,TEMPCH  ;MAKE IT 7 BIT ASCII
1057  004240  023727  000700  000004      82300           CMP     TEMPCH,#4       ;DISREGARD EOT
1058  004246  001715                      82400           BEQ     $READ
1059  004250  012700  000011              82500           MOV     #11,R0          ;SET SHIFT COUNT
1060  004254  042737  000377  000702      82600           BIC     #377,PARITY     ;CLEAR PARITY FLAG
1061  004262  005300                      82700   1$:     DEC     R0              ;DECREMENT SHIFT COUNT
1062  004264  001406                      82800           BEQ     2$              ;EXIT IF DONE
1063  004266  106337  000703              82900           ASLB    PARITY+1        ;SHIFT CODE
1064  004272  103373                      83000           BCC     1$              ;CONTINUE IF BIT WAS ZERO
1065  004274  105137  000702              83100           COMB    PARITY          ;CHANGE PARITY FLAG IF BIT WAS ONE
1066  004300  000770                      83200           BR      1$              ;CONTINUE
1067  004302  000002                      83300   2$:     RTI                     ;SET, RET. TO CALLER
1068                                    83400
1069                                    83500   ;XXXXXXXXXX
1070                                    83600   ;
1071                                    83700   ;PRINT-- A COMMON ROUTINE TO CHECK THE PRINTER READY FLAG
1072                                    83800   ;
1073                                    83900   ;XXXXXXXXXX
1074                                    84000
1075  004304  105777  174306              84100   $PRNT:  TSTB    @TPS            ;CHECK PRINTER READY FLAG
1076  004310  100375                      84200           BPL     $PRNT           ;BRANCH IF NOT SET
1077  004312  000002                      84300           RTI                     ;SET, RETURN
```

```
1079                                    84500   ;**********
1080                                    84600   ;
1081                                    84700   ;PRINTC--SENDS A CHARACTER AT A TIME FIRST TO THE
1082                                    84800   ;       CONSOLE  DL11 THEN TO ALL MULTIPLE DL11S IF
1083                                    84900   ;       SR BIT 13 IS = 0.  IF THE REFERENCED PRINTER
1084                                    85000   ;       READY BIT IS NOT SET, THE CHARACTER WILL NOT BE
1085                                    85100   ;       SENT TO THAT PRINTER.  ENTER WITH CHARACTER IN R0
1086                                    85200   ;       CALL:   PRINTC
1087                                    85300   ;
1088                                    85400   ;********
1089                                    85500
1090  004314  013737  000602  000712    85600   $PRTC:  MOV     CONADD,TEMP     ;SET CONSOLE ADR
1091  004322  062737  000004  000712    85700           ADD     #4,TEMP
1092  004330  105777  174356            85800   1$:     TSTB    @TEMP
1093  004334  100375                    85900           BPL     1$              ;WAIT FOR CONSOLE READY
1094  004336  062737  000002  000712    86000           ADD     #2,TEMP         ;SET ADR
1095  004344  010077  174342            86100           MOV     R0,@TEMP        ;LOAD CONSOLE PRINT BUFFER
1096  004350  032777  020000  174336    86200           BIT     #BIT13,@SR      ;CHECK SW 13
1097  004356  001003                    86300           BNE     2$              ;SEND ALL TERMS IF SW13 DOWN
1098  004360  005737  000656            86400           TST     DLCNT           ;CHECK IF MULTIPLE DL11'S
1099  004364  001002                    86500           BNE     3$              ;CHECK FOR INPUT IF THERE
1100  004366  000137  005034            86600   2$:     JMP     18$
1101  004372  013737  000656  000666    86700   3$:     MOV     DLCNT,COUNT3    ;PUT NO. DL11'S INTO COUNT3
1102  004400  013737  000632  000670    86800           MOV     FSTDL,XCSR      ;ADDR OF FIRST DL INTO XCSR
1103  004406  023727  000636  000032    86900   4$:     CMP     RTNNO,#32       ;CHECK IF TEST 32
1104  004414  001543                    87000           BEQ     13$             ;DON'T CHECK FOR INPUT IF TEST 32
1105  004416  005737  000710            87100           TST     INCHK           ;CHECK FOR INPUT?
1106  004422  001140                    87200           BNE     13$
1107  004424  023727  000636  000020    87300           CMP     RTNNO,#20       ;PRINTING TEST?
1108  004432  002004                    87400           BGE     5$              ;BRANCH IF NOT
1109  004434  022737  104011  001700    87500           CMP     #TTYCTL,WAITF   ;KEYBOARD CONTROL?
1110  004442  001130                    87600           BNE     13$             ;SKIP INPUT CHECK IF NOT
1111  004444  105777  174220            87700   5$:     TSTB    @XCSR           ;TEST IF ANY INPUT
1112  004450  100125                    87800           BPL     13$             ;CONTINUE IF NO INPUT
1113  004452  062737  000002  000670    87900           ADD     #2,XCSR         ;SET BUFFER ADDRESS
1114  004460  017737  174204  000700    88000           MOV     @XCSR,TEMPCH
1115  004466  042737  177600  000700    88100           BIC     #177600,TEMPCH
1116  004474  023727  000700  000003    88200           CMP     TEMPCH,#3       ;CHECK IF CONTROL-C
1117  004502  001006                    88300           BNE     6$              ;CONTINUE IF NOT
1118  004504  023727  000636  000024    88400           CMP     RTNNO,#24       ;CHECK IF TEST 24
1119  004512  001002                    88500           BNE     6$              ;CONTINUE IF NOT CONTROL-C
1120  004514  000137  005150            88600           JMP     20$
1121  004520  023727  000700  000177    88700   6$:     CMP     TEMPCH,#177     ;CHECK IF RUBOUT
1122  004526  001427                    88800           BEQ     9$              ;YES, CHECK TEST NUMBER
1123  004530  023727  000636  000017    88900           CMP     RTNNO,#17       ;TEST 17?
1124  004536  001003                    89000           BNE     7$              ;BRANCH IF NOT
1125  004540  013703  000700            89100           MOV     TEMPCH,R3       ;SAVE CHAR
1126  004544  000461                    89200           BR      12$             ;CONTINUE
1127  004546  023727  000636  000021    89300   7$:     CMP     RTNNO,#21       ;TEST 21?
1128  004554  001004                    89400           BNE     8$              ;BRANCH IF NOT
1129  004556  013737  000700  000662    89500           MOV     TEMPCH,REPT     ;SAVE CHAR
1130  004564  000451                    89600           BR      12$             ;CONTINUE
1131  004566  023727  000636  000022    89700   8$:     CMP     RTNNO,#22       ;TEST 22?
1132  004574  001056                    89800           BNE     14$             ;CONTINUE IF NOT
1133  004576  013737  000700  000662    89900           MOV     TEMPCH,REPT     ;SAVE CHAR
1134  004604  000441                    90000           BR      12$             ;CONTINUE
```

```
1135  004606  023727  000636  000021  90100  9$:    CMP     RTNNO,#21       ;CHECK IF TEST 21
1136  004614  001011                  90200         BNE     10$             ;NO, CHECK IF TEST 22
1137  004616  022626                  90300         POPSP2                  ;ASJUST STACK
1138  004620  012700  000036          90400         MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
1139  004624  104010                  90500         DELAY
1140  004626  104007                  90600         TYPEM                   ;YES, TEST 21
1141  004630  014255                  90700         ECOEND                  ;PRINT TERMINATION MESSAGE
1142  004632  104005                  90800         CHAIN                   ;CHAIN TO NEXT TEST
1143  004634  000137  012174          90900         JMP     E021A           ;REPEAT TEST IF LOOP ON TEST SW SET
1144  004640  023727  000636  000022  91000  10$:    CMP     RTNNO,#22       ;CHECK IF TEST 22
1145  004646  001011                  91100         BNE     11$             ;NO, CHECK IF TEST 24
1146  004650  022626                  91200         POPSP2                  ;ADJUST STACK
1147  004652  012700  000036          91300         MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
1148  004656  104010                  91400         DELAY
1149  004660  104007                  91500         TYPEM                   ;YES, PRINT TERMINATION MESSAGE
1150  004662  014255                  91600         ECOEND
1151  004664  104005                  91700         CHAIN                   ;CHAIN TO NEXT TEST
1152  004666  000137  012232          91800         JMP     E022A           ;REPEAT TEST IF LOOP ON TEST SW SET
1153  004672  023727  000636  000024  91900  11$:    CMP     RTNNO,#24       ;TEST 24?
1154  004700  001137                  92000         BNE     22$             ;WAIT FOR NEXT TEST IF NOT TEST 24
1155  004702  022626                  92100         POPSP2                  ;RESET STACK
1156  004704  000137  013154          92200         JMP     TERM            ;TERMINATE TEST
1157  004710  012700  000036          92300  12$:    MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
1158  004714  104010                  92400         DELAY
1159  004716  013700  000700          92500         MOV     TEMPCH,R0       ;SET NEW CHAR
1160  004722  000403                  92600         BR      14$             ;CONTINUE
1161  004724  062737  000002  000670  92700  13$:    ADD     #2,XCSR         ;SET STATUS ADDRESS IN XCSR
1162  004732  062737  000002  000670  92800  14$:    ADD     #2,XCSR
1163  004740  013737  000602  000712  92900         MOV     CONADD,TEMP     ;CHECK IF CONSOLE TERM
1164  004746  062737  000004  000712  93000         ADD     #4,TEMP         ;IS THIS DL
1165  004754  023737  000712  000670  93100         CMP     TEMP,XCSR
1166  004762  001420                  93200         BEQ     17$
1167  004764  105777  173700          93300  15$:    TSTB    @XCSR           ;TEST PRINTER READY
1168  004770  100375                  93400         BPL     15$             ;WAIT FOR READY
1169  004772  062737  000002  000670  93500         ADD     #2,XCSR         ;SET XCSR TO PRINTER BUFFER
1170  005000  010077  173664          93600         MOV     R0,@XCSR        ;LOAD CHARACTER INTO BUFFER
1171  005004  005337  000666          93700  16$:    DEC     COUNT3          ;DECREASE COUNT OF DL11'S
1172  005010  001411                  93800         BEQ     18$             ;ALL DONE,EXIT
1173  005012  062737  000002  000670  93900         ADD     #2,XCSR         ;SET XCSR TO NEXT DL11 PRINTER STATUS
1174  005020  000137  004406          94000         JMP     4$              ;GO TEST NEXT DL11 READY FLAG
1175  005024  062737  000002  000670  94100  17$:    ADD     #2,XCSR         ;SET XCSR TO PRINTER BUFFER
1176  005032  000764                  94200         BR      16$             ;DO NOT LOAD BUFFER
1177  005034  023727  000636  000032  94300  18$:    CMP     RTNNO,#32       ;CHECK IF TEST 32
1178  005042  001514                  94400         BEQ     26$             ;DON'T CHECK FOR INPUT IF TEST 32
1179  005044  005737  000710          94500         TST     INCHK           ;WANT INPUT CHECK?
1180  005050  001111                  94600         BNE     26$             ;NO, BRANCH
1181  005052  023727  000636  000020  94700         CMP     RTNNO,#20       ;PRINTING TEST?
1182  005060  002004                  94800         BGE     19$             ;BRANCH IF NOT
1183  005062  022737  104011  001700  94900         CMP     #TTYCTL,WAITF   ;KEYBOARD CONTROL?
1184  005070  001101                  95000         BNE     26$             ;SKIP INPUT CHECK IF NOT
1185  005072  105777  173504          95100  19$:    TSTB    @CONADD         ;TEST IF ANY INPUT
1186  005076  100076                  95200         BPL     26$             ;BRANCH IF NONE
1187  005100  013737  000602  000712  95300         MOV     CONADD,TEMP     ;SET ADR
1188  005106  062737  000002  000712  95400         ADD     #2,TEMP
1189  005114  117737  173572  000700  95500         MOVB    @TEMP,TEMPCH
1190  005122  042737  177600  000700  95600         BIC     #177600,TEMPCH  ;MASK UNWANTED BITS
```

```
 1191  005130  023727  000700  000003  95700          CMP     TEMPCH,#3       ;CHAR = CONTROL-C?
 1192  005136  001013                  95800          BNE     21$             ;CONTINUE IF NOT
 1193  005140  023727  000636  000024  95900          CMP     RTNNO,#24       ;TEST 24?
 1194  005146  001007                  96000          BNE     21$             ;CONTINUE IF NOT
 1195  005150  012700  000036          96100  20$:    MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
 1196  005154  104010                  96200          DELAY
 1197  005156  104012                  96300          CRLF                    ;SEND CR-LF
 1198  005160  022626                  96400          POPSP2                  ;RESET STACK
 1199  005162  000137  013026          96500          JMP     E024B           ;RETURN TO TEST
 1200  005166  023727  000700  000177  96600  21$:    CMP     TEMPCH,#177     ;CHECK IF RUBOUT
 1201  005174  001006                  96700          BNE     23$             ;BRANCH IF NO
 1202  005176  000603                  96800          BR      9$
 1203  005200  012737  000001  000634  96900  22$:    MOV     #1,CNTLSW       ;CLEAR LOOP AND SEQUENCE BITS
 1204  005206  000137  002062          97000          JMP     TTY1B           ;GO WAIT FOR NEXT TEST
 1205  005212  010046                  97100  23$:    MOV     R0,-(SP)        ;SAVE R0
 1206  005214  012700  000036          97200          MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
 1207  005220  104010                  97300          DELAY
 1208  005222  012600                  97400          MOV     (SP)+,R0        ;RESTORE R0
 1209  005224  023727  000636  000017  97500          CMP     RTNNO,#17       ;CHECK IF TEST 17
 1210  005232  001002                  97600          BNE     24$             ;BRANCH IF NOT TEST 17
 1211  005234  013703  000700          97700          MOV     TEMPCH,R3       ;STORE INPUTTED CHAR
 1212  005240  023727  000636  000021  97800  24$:    CMP     RTNNO,#21       ;CHECK IF TEST 21
 1213  005246  001003                  97900          BNE     25$             ;BRANCH IF NOT TEST 21
 1214  005250  013737  000700  000662  98000          MOV     TEMPCH,REPT     ;STORE INPUTTED CHAR
 1215  005256  023727  000636  000022  98100  25$:    CMP     RTNNO,#22       ;CHECK IF TEST 22
 1216  005264  001003                  98200          BNE     26$             ;BRANCH IF NOT TEST 22
 1217  005266  013737  000700  000662  98300          MOV     TEMPCH,REPT     ;STORE INPUTTED CHAR
 1218  005274  000002                  98400  26$:    RTI                     ;RETURN TO TEST
 1219                                  98500
```

```
1221
```

```
1222                                    00100          .SBTTL  I/O LOGIC TESTS
1223                                    00200  ;
1224                                    00300  ;**********
1225                                    00400  ;
1226                                    00500  ;ONLY THE CONSOLE TERMINAL IS TESTED.
1227                                    00600  ;UPON COMPLETION, THE CPU WILL EITHER HALT IF SR
1228                                    00700  ;BIT8 IS = 1 AND AWAIT FUTHER INSTRUCTIONS OR CONTINUE
1229                                    00800  ;AND EXECUTE THE PRINTER TESTS CONTINOUSLY
1230                                    00900  ;IF AN I/O TEST FAILS, THE CPU WILL HALT AT ERRHLT
1231                                    01000  ;WITH THE ADDRESS OF THE ERROR IN R0 (LOC 777700).  PRESSING
1232                                    01100  ;THE CONTINUE SWITCH WILL CAUSE THE I/O TEST TO
1233                                    01200  ;CONTINUE WITH THE NEXT TEST.  HOWEVER IF SWITCH 14
1234                                    01300  ;WERE SET, OR IS SET BEFORE THE CONTINUE SWITCH IS
1235                                    01400  ;PRESSED, THE FAILED TEST WILL LOOP ON ITSELF
1236                                    01500  ;WITHOUT FUTHER HALTS
1237                                    01600  ;
1238                                    01700  ;**********
1239                                    01800  ;AT0--  TEST #40--TESTS THE ABILITY TO REFERENCE THE
1240                                    01900  ;       RECEIVER STATUS WORD (TKS) WITHOUT TRAPPING.
1241                                    02000  ;**********
1242                                    02100
1243  005276  000040                    02200  AT0:    40                      ;TEST NUMBER
1244  005300  005330                    02300  AT0X:   AT1                     ;NEXT TEST
1245  005302  000012                    02400          10                      ;ITERATION COUNT
1246  005304  005314                    02500          1$                      ;SCOPE ENTRY
1247  005306  012737  005324  000004    02600          MOV     #3$,@MACHER     ;SET UP MACHINE ERROR TRAP
1248  005314  005777  173272            02700  1$:     TST     @TKS            ;REFERENCE RECEIVER STATUS WORD
1249  005320  104005                    02800  2$:     CHAIN                   ;CHAIN TO NEXT TEST
1250  005322  000774                    02900          BR      1$              ;REPEAT TEST
1251  005324  104001                    03000  3$:     ERROR                   ;ERROR TRAPPED WHEN REFERENCING
1252  005326  000774                    03100          BR      2$              ;RECEIVER STATUS WORD (TKS)
1253                                    03200
1254                                    03300  ;**********
1255                                    03400  ;AT1--TEST #41--TESTS THE ABILITY TO  REFERENCE THE
1256                                    03500  ;       RECEIVER BUFFER (TKB) WITHOUT TRAPPING.
1257                                    03600  ;**********
1258                                    03700
1259  005330  000041                    03800  AT1:    41                      ;TEST NUMBER
1260  005332  005362                    03900          AT2                     ;NEXT TEST
1261  005334  000012                    04000          10.                     ;ITERATION COUNT
1262  005336  005346                    04100          1$                      ;SCOPE ENTRY
1263  005340  012737  005356  000004    04200          MOV     #3$,@MACHER     ;SET UP MACHINE ERROR TRAP
1264  005346  005777  173242            04300  1$:     TST     @TKB            ;REFERENCE RECEIVER BUFFER
1265  005352  104005                    04400  2$:     CHAIN                   ;CHAIN TO NEXT TEST
1266  005354  000774                    04500          BR      1$              ;REPEAT TEST
1267  005356  104001                    04600  3$:     ERROR                   ;TRAPPED WHEN REFERENCING
1268  005360  000774                    04700          BR      2$              ;RECEIVER BUFFER (TKB)
```

```
1270                                    04900   ;**********
1271                                    05000   ;AT2--TEST #42--TESTS THE ABILITY TO REFERENCE THE
1272                                    05100   ;       TRANSMITTER STATUS WORD (TPS) WITHOUT TRAPPING.
1273                                    05200   ;**********
1274                                    05300
1275  005362  000042                    05400   AT2:    42                      ;TEST NUMBER
1276  005364  005414                    05500           AT3                     ;NEXT TEST
1277  005366  000012                    05600           10.                     ;ITERATION COUNT
1278  005370  005400                    05700           1$                      ;SCOPE ENTRY
1279  005372  012737  005410  000004    05800           MOV     #3$,MACHER      ;SET UP MACHINE ERROR TRAP
1280  005400  005777  173212            05900   1$:     TST     @TPS            ;REFERENCE TRANSMITTER STATUS
1281  005404  104005                    06000   2$:     CHAIN                   ;CHAIN TO NEXT TEST
1282  005406  000774                    06100           BR      1$              ;REPEAT TEST
1283  005410  104001                    06200   3$:     ERROR                   ;TRAPPED WHEN REFERENCING
1284  005412  000774                    06300           BR      2$              ;TRANSMITTER STATUS WORD
1285                                    06400
1286                                    06500   ;**********
1287                                    06600   ;AT3--  TEST #43--TESTS THE ABILITY TO REFERENCE THE
1288                                    06700   ;       TRANSMITTER BUFFER (TPB) WITHOUT TRAPPING.
1289                                    06800   ;**********
1290                                    06900
1291  005414  000043                    07000   AT3:    43                      ;TEST NUMBER
1292  005416  005446                    07100           AT4                     ;NEXT TEST
1293  005420  000012                    07200           10                      ;ITERATION COUNT
1294  005422  005432                    07300           1$                      ;SCOPE ENTRY
1295  005424  012737  005442  000004    07400           MOV     #3$,MACHER      ;SET UP ERROR TRAP
1296  005432  005777  173162            07500   1$:     TST     @TPB            ;REFERENCE TRANSMITTER BUFFER
1297  005436  104005                    07600   2$:     CHAIN                   ;CHAIN TO NEXT TEST
1298  005440  000774                    07700           BR      1$              ;REPEAT TEST
1299  005442  104001                    07800   3$:     ERROR                   ;TRAPPED WHEN REFERENCING
1300  005444  000774                    07900           BR      2$              ;TRANSMITTER BUFFER.
```

```
1302                                    08100   ;**********
1303                                    08200   ;AT4--  TEST #44--TESTS THE ABILITY TO SET AND CLEAR THE
1304                                    08300   ;       RECEIVER INTERRUPT ENABLE BIT.
1305                                    08400   ;**********
1306                                    08500
1307  005446  000044                    08600   AT4:    44                      ;TEST NUMBER
1308  005450  005536                    08700           AT5                     ;NEXT TEST
1309  005452  000012                    08800           10.                     ;ITERATION COUNT
1310  005454  005470                    08900           1$                      ;SCOPE ENTRY
1311  005456  012746  000340            09000           MOV     #PRTY7,-(SP)    ;SET PRIORITY 7
1312  005462  012746  005470            09100           MOV     #1$,-(SP)
1313  005466  000002                    09200           RTI
1314  005470  052777  000100  173114    09300   1$:     BIS     #BIT6,@TKS      ;SET INTERRUPT ENABLE BIT
1315  005476  032777  000100  173106    09400           BIT     #BIT6,@TKS      ;CHECK IF BIT IS SET
1316  005504  001002                    09500           BNE     3$              ;BRANCH IF SET
1317  005506  104001                    09600   2$:     ERROR                   ;NOT SET, ERROR
1318  005510  000410                    09700           BR      5$              ;TRY AGAIN
1319  005512  042777  000100  173072    09800   3$:     BIC     #BIT6,@TKS      ;CLEAR INTERRUPT ENABLE BIT
1320  005520  032777  000100  173064    09900           BIT     #BIT6,@TKS      ;CHECK IF BIT IS CLEARED
1321  005526  001401                    10000           BEQ     5$              ;BRANCH IF CLEARED
1322  005530  104001                    10100   4$:     ERROR                   ;NOT CLEARED, ERROR
1323  005532  104005                    10200   5$:     CHAIN                   ;CHAIN TO NEXT TEST
1324  005534  000755                    10300           BR      1$              ;DO TEST AGAIN
1325                                    10400
1326                                    10500   ;**********
1327                                    10600   ;AT5--  TEST #45--CHECKS THAT THE RECEIVER INTERRUPT
1328                                    10700   ;       ENABLE BIT CAN BE CLEARED WITH RESET INSTRUCTION.
1329                                    10800   ;**********
1330                                    10900   ;
1331  005536  000045                    11000   AT5:    45                      ;TEST NUMBER
1332  005540  005614                    11100           AT6                     ;NEXT TEST
1333  005542  000012                    11200           10.                     ;ITERATION COUNT
1334  005544  005560                    11300           1$                      ;SCOPE ENTRY
1335  005546  012746  000340            11400           MOV     #PRTY7,-(SP)    ;SET PRIORITY TO 7
1336  005552  012746  005560            11500           MOV     #1$,-(SP)
1337  005556  000002                    11600           RTI
1338  005560  052777  000100  173024    11700   1$:     BIS     #BIT6,@TKS      ;SET INTERRUPT ENABLE BIT
1339  005566  105777  173024            11800   3$:     TSTB    @TPS            ;BE SURE PRINTER IS DONE WITH DL11S1 MESSAGE
1340  005572  001775                    11900           BEQ     3$              ;BEFORE ALLOWING FOLLOWING RESET.
1341  005574  000005                    12000           RESET                   ;RESET
1342  005576  032777  000100  173006    12100           BIT     #BIT6,@TKS      ;TEST INTERRUPT ENABLE BIT
1343  005604  001401                    12200           BEQ     2$              ;BRANCH IF CLEARED
1344  005606  104001                    12300           ERROR                   ;STILL SET,ERROR
1345  005610  104005                    12400   2$:     CHAIN                   ;CHAIN TO NEXT ROUTINE
1346  005612  000762                    12500           BR      1$              ;REPEAT TEST
```

```
1348                                    12700   ;**********
1349                                    12800   ;AT6--  TEST#46--TESTS THE ABILITY TO SET AND CLEAR
1350                                    12900   ;       TRANSMITTER INTERRUPT ENABLE BIT.
1351                                    13000   ;**********
1352                                    13100
1353  005614  000046                    13200   AT6:    46                      ;TEST NUMBER
1354  005616  005704                    13300           AT7                     ;NEXT TEST
1355  005620  000012                    13400           10.                     ;ITERATION COUNT
1356  005622  005636                    13500           1$                      ;SCOPE ENTRY
1357  005624  012746  000340            13600           MOV     #PPTY7,-(SP)    ;SET PRIORITY TO 7
1358  005630  012746  005636            13700           MOV     #1$,-(SP)
1359  005634  000002                    13800           RTI
1360  005636  052777  000100  172752    13900   1$:     BIS     #BIT6,@TPS      ;SET INTERRUPT ENABLE BIT
1361  005644  032777  000100  172744    14000           BIT     #BIT6,@TPS      ;CHECK THAT BIT IS SET
1362  005652  001002                    14100           BNE     2$              ;BRANCH IF SET
1363  005654  104001                    14200           ERROR                   ;NOT SET, ERROR
1364  005656  000410                    14300           BR      3$              ;TRY AGAIN
1365  005660  042777  000100  172730    14400   2$:     BIC     #BIT6,@TPS      ;CLEAR INTERRUPT ENABLE BIT
1366  005666  032777  000100  172722    14500           BIT     #BIT6,@TPS      ;CHECK IF BIT IS CLEARED
1367  005674  001401                    14600           BEQ     3$              ;BRANCH IF CLEARED
1368  005676  104001                    14700           ERROR                   ;NOT CLEARED, ERROR
1369  005700  104005                    14800   3$:     CHAIN                   ;CHAIN TO NEXT TEST
1370  005702  000755                    14900           BR      1$              ;DO AGAIN
1371                                    15000
1372                                    15100   ;**********
1373                                    15200   ;AT7--  TEST #47--TESTS THE ABILITY TO CLEAR TRANSMITTER
1374                                    15300   ;       INTERRUPT ENABLE BIT WITH RESET INSTRUCTION.
1375                                    15400   ;**********
1376                                    15500
1377  005704  000047                    15600   AT7:    47                      ;TEST NUMBER
1378  005706  005754                    15700           AT10                    ;NEXT TEST
1379  005710  000012                    15800           10.                     ;ITERATION COUNT
1380  005712  005726                    15900           1$                      ;SCOPE ENTRY
1381  005714  012746  000340            16000           MOV     #PRTY7,-(SP)    ;SET PRIORITY TO 7
1382  005720  012746  005726            16100           MOV     #1$,-(SP)
1383  005724  000002                    16200           RTI
1384  005726  052777  000100  172662    16300   1$:     BIS     #BIT6,@TPS      ;SET INTERRUPT BIT
1385  005734  000005                    16400           RESET                   ;RESET
1386  005736  032777  000100  172652    16500           BIT     #BIT6,@TPS      ;CHECK IF BIT IS CLEARED
1387  005744  001401                    16600           BEQ     2$              ;BRANCH IF CLEARED
1388  005746  104001                    16700           ERROR                   ;ERROR, RESET DID NOT CLEAR BIT
1389  005750  104005                    16800   2$:     CHAIN                   ;CHAIN TO NEXT ROUTINE
1390  005752  000765                    16900           BR      1$              ;REPEAT TEST
```

```
1392                                    17100   ;**********
1393                                    17200   ;AT10-- TEST #50--CHECKS THAT RESET SETS THE TRANSMITTER
1394                                    17300   ;       READY BIT AND THAT THE READY BIT CAN BE READ RELIABLY.
1395                                    17400   ;**********
1396                                    17500
1397  005754  000050                    17600   AT10:   50                      ;TEST NUMBER
1398  005756  006012                    17700           AT11                    ;NEXT TEST
1399  005760  000012                    17800           10.                     ;ITERATION COUNT
1400  005762  005764                    17900           1$                      ;SCOPE ENTRY
1401  005764  032777  001000  172722    17920   1$:     BIT     #LSI11,@SR      ;SKIP TEST IF AN LSI-11
1402  005772  001005                    17940           BNE     2$
1403  005774  000005                    18000           RESET                   ;RESET
1404  005776  105777  172614            18100           TSTB    @TPS            ;CHECK TRANSMIT READY BIT
1405  006002  100401                    18200           BMI     2$              ;BRANCH IF SET
1406  006004  104001                    18300           ERROR                   ;ERROR, RESET DID NOT SET READY BIT
1407  006006  104005                    18400   2$:     CHAIN                   ;CHAIN TO NEXT TEST
1408  006010  000765                    18500           BR      1$              ;DO AGAIN
1409                                    18600
1410                                    18700   ;**********
1411                                    18800   ;AT11-- TEST #51--TESTS THAT THE TRANSMITTER READY RESETS
1412                                    18900   ;       BY LOADING THE TRANSMITTER BUFFER.
1413                                    19000   ;**********
1414                                    19100
1415  006012  000051                    19200   AT11:   51                      ;TEST NUMBER
1416  006014  006052                    19300           AT12                    ;NEXT TEST
1417  006016  000012                    19400           10.                     ;ITERATION COUNT
1418  006020  006022                    19500           1$                      ;SCOPE ENTRY
1419  006022  012700  000226            19600   1$:     MOV     #226,R0
1420  006026  104010                    19700           DELAY                   ;DELAY 150 MSEC.
1421  006030  000005                    19800           RESET                   ;RESET
1422  006032  005077  172562            19900           CLR     @TPB            ;LOAD TRANSMITTER BUFFER
1423  006036  105777  172554            20000           TSTB    @TPS            ;CHECK TRANSMIT READY BIT
1424  006042  100001                    20100           BPL     2$              ;BRANCH IF CLEARED
1425  006044  104001                    20200           ERROR                   ;NOT CLEARED, ERROR
1426  006046  104005                    20300   2$:     CHAIN                   ;CHAIN TO NEXT TEST
1427  006050  000764                    20400           BR      1$              ;REPEAT TEST
```

```
1429                                            20600   ;**********
1430                                            20700   ;AT12-- TEST #52--CHECKS THAT THE TRANSMIT READY BIT CAN
1431                                            20800   ;       CAUSE AN INTERRUPT
1432                                            20900   ;**********
1433                                            21000
1434  006052  000052                            21100   AT12:   52                      ;TEST NUMBER
1435  006054  006126                            21200           AT13                    ;NEXT TEST
1436  006056  000012                            21300           10.                     ;ITERATION COUNT
1437  006060  006066                            21400           1$                      ;SCOPE ENTRY
1438  006062  104004                            21500           STPCHV                  ;SET UP TRANSMITTER INTERRUPT VECTOR
1439  006064  006122                            21600           4$                      ;TO 3$
1440  006066  000005                            21700   1$:     RESET                   ;SEE CHAINY COMMENT
1441  006070  005077  172522                    21800           CLR     @TPS            ;DISABLE TRANSMIT INTERRUPT
1442  006074  005046                            21900           CLR     -(SP)           ;SET PRIORITY TO ZERO
1443  006076  012746  006104                    22000           MOV     #2$,-(SP)
1444  006102  000002                            22100           RTI
1445  006104  052777  000100  172504            22200   2$:     BIS     #BIT6,@TPS      ;ENABLE TRANSMIT INTERRUPT
1446  006112  000240                            22300           NOP
1447  006114  104001                            22400           ERROR                   ;TRANSMIT READY DID NOT CAUSE INTERRUPT
1448  006116  104005                            22500   3$:     CHAIN                   ;CHAN TO NEXT TEST
1449  006120  000762                            22600           BR      1$              ;REPEAT TEST
1450  006122  022626                            22700   4$:     POPSP2                  ;INTERRUPT OCCURED, CLEAN STACK
1451  006124  000774                            22800           BR      3$              ;CHAIN TO NEXT TEST
1452                                            22900
1453                                            23000   ;**********
1454                                            23100   ;AT13-- TEST#53--TESTS THAT THE TRANSMIT READY DOES NOT CAUSE AN
1455                                            23200   ;       INTERRUPT WHEN THE PROCESSOR IS AT THE SAME LEVEL
1456                                            23300   ;**********
1457                                            23400
1458  006126  000035                            23500   AT13:   35                      ;TEST NUMBER
1459  006130  006206                            23600           AT14                    ;NEXT TEST
1460  006132  000012                            23700           10.                     ;ITERATION COUNT
1461  006134  006142                            23800           1$                      ;SCOPE ENTRY
1462  006136  104004                            23900           STPCHV                  ;SET UP TRANSMIT INTERRUPT
1463  006140  006200                            24000           4$                      ;VECTOR TO 3$
1464  006142  013746  000630                    24100   1$:     MOV     TPLVL,-(SP)     ;SET PROCESSOR TO SAME LEVEL AS XMITTER
1465  006146  012746  006154                    24200           MOV     #2$,-(SP)
1466  006152  000002                            24300           RTI
1467  006154  005077  172436                    24400   2$:     CLR     @TPS            ;DISABLE TRANSMITTER INTERRUPTS
1468  006160  052777  000100  172430            24500           BIS     #BIT6,@TPS      ;ENABLE TRANSMITTER INTERRUPTS
1469  006166  000240                            24600           NOP
1470  006170  005077  172422                    24700   3$:     CLR     @TPS            ;OK, NO INTERRUPT OCCURED
1471  006174  104005                            24800           CHAIN                   ;CHAIN TO NEXT TEST
1472  006176  000761                            24900           BR      1$              ;REPEAT TEST
1473  006200  022626                            25000   4$:     POPSP2                  ;INTERRUPT OCURRED,ERROR,CLEAN
1474  006202  104001                            25100           ERROR                   ;UP STACK
1475  006204  000771                            25200           BR      3$              ;CHAIN TO NEXT TEST
```

```
1477                                    25400   ;**********
1478                                    25500   ;AT14-- TEST#54--TESTS THAT THE TRANSMIT READY DOES CAUSE AN
1479                                    25600   ;       INTERRUPT WHEN THE PROCESSOR IS AT A PRIORITY LEVEL
1480                                    25700   ;       ONE LOWER THAN THE TRANSMIT INTERRUPT REQUEST LEVEL
1481                                    25800   ;**********
1482                                    25900
1483  006206  000054                    26000   AT14:   54                      ;TEST NUMBER
1484  006210  006272                    26100           AT15                    ;NEXT TEST
1485  006212  000012                    26200           10.                     ;ITERATION COUNT
1486  006214  006222                    26300           1$                      ;SCOPE ENTRY
1487  006216  104004                    26400           STPCHV                  ;SET UP TRANSMIT INTERRUPT
1488  006220  006260                    26500           3$                      ;VECTOR TO 2$
1489  006222  005077  172370            26600   1$:     CLR     @TPS            ;DISABLE TRANSMIT INTERRUPTS
1490  006226  013746  000630            26700           MOV     TPLVL,-(SP)     ;SET PROCESSOR PRIORITY ONE
1491  006232  162716  000040            26800           SUB     #40,(SP)        ;LEVEL LOWER THAN TRANSMITTER
1492  006236  012746  006244            26900           MOV     #2$,-(SP)
1493  006242  000002                    27000           RTI
1494  006244  052777  000100  172344    27100   2$:     BIS     #BIT6,@TPS      ;ENABLE TRANSMITTER INTERRUPTS
1495  006252  000240                    27200           NOP
1496  006254  104001                    27300           ERROR                   ;NO INTERRUPT, ERROR
1497  006256  000401                    27400           BR      4$              ;CHAIN TO NEXT TEST
1498  006260  022626                    27500   3$:     POPSP2                  ;INTERRUPT OCCURED, OK, CLEAN STACK
1499  006262  005077  172330            27600   4$:     CLR     @TPS            ;DISABLE TRANSMITTER INTERRUPTS
1500  006266  104005                    27700           CHAIN                   ;CHAIN TO NEXT TEST
1501  006270  000754                    27800           BR      1$              ;REPEAT TEST
```

```
1503                                    28000   ;**********
1504                                    28100   ;AT15-- TEST#55--TESTS THAT THE TRANSMIT READY DOES NOT
1505                                    28200   ;       REINTERRUPT AFTER AN RTI WHEN THE READY BIT HAS
1506                                    28300   ;       NOT BEEN RESET.
1507                                    28400   ;**********
1508                                    28500
1509  006272  000055                    28600   AT15:   55                      ;TEST NUMBER
1510  006274  006372                    28700           AT16                    ;NEXT TEST
1511  006276  000012                    28800           10.                     ;ITERATION COUNT
1512  006300  006302                    28900           1$                      ;SCOPE ENTRY
1513  006302  104004                    29000   1$:     STPCHV                  ;SET TRANSMIT INTERRUPT VECTOR
1514  006304  006344                    29100           4$                      ;TO 3$
1515  006306  005077  172304            29200           CLR     @TPS            ;DISABLE TRANSMITTER INTERRUPTS
1516  006312  005046                    29300           CLR     -(SP)           ;SET PROCESSOR PRIORITY TO ZERO
1517  006314  012746  006322            29400           MOV     #2$,-(SP)
1518  006320  000002                    29500           RTI
1519  006322  052777  000100  172266    29600   2$:     BIS     #BIT6,@TPS      ;ENABLE TRANSMITTER INTERRUPTS
1520  006330  000240                    29700           NOP
1521  006332  104001                    29800           ERROR                   ;ERROR1, TRANSMITTER FAILED TO INTERRUPT
1522  006334  005077  172256            29900   3$:     CLR     @TPS            ;DISABLE TRANSMITTER INTERRUPTS
1523  006340  104005                    30000           CHAIN                   ;CHAIN TO NEXT TEST
1524  006342  000757                    30100           BR      1$              ;REPEAT TEST
1525  006344  012777  006364  172254    30200   4$:     MOV     #6$,@TPVTR      ;INTERRUPT OCCURED, CHANGE INTERRUPT
1526  006352  012716  006360            30300           MOV     #5$,@SP         ;VECTOR TO 5$ AND RETURN TO 4$
1527  006356  000002                    30400           RTI                     ;RETURN FROM INTERRUPT
1528  006360  000240                    30500   5$:     NOP
1529  006362  000764                    30600           BR      3$              ;CHAIN TO NEXT TEST
1530  006364  022626                    30700   6$:     POPSP2                  ;ERROR2, TRANSMITTER REINTERRUPTED
1531  006366  104001                    30800           ERROR                   ;AFTER RTI WITH READY BIT LEFT ON.
1532  006370  000761                    30900           BR      3$              ;CLEAN STACK, CHAIN TO NEXT TEST.
1533                                    31000
1534                                    31100   ;**********
1535                                    31200   ;AT16--TEST#56--CHECKS THAT RESET CLEARS THE RECEIVER DONE BIT
1536                                    31300   ;**********
1537                                    31400
1538  006372  000056                    31500   AT16:   56                      ;TEST NUMBER
1539  006374  006440                    31600           AT17                    ;NEXT TEST
1540  006376  000012                    31700           10.                     ;ITERATION COUNT
1541  006400  006402                    31800           1$                      ;SCOPE ENTRY
1542  006402  032777  001000  172304    31900   1$:     BIT     #LSI11,@SR      ;SKIP TEST IF LSI-11
1543  006410  001011                    32000           BNE     3$
1544  006412  012700  000226            32100           MOV     #226,R0
1545  006416  104010                    32200           DELAY                   ;DELAY 150 MSEC.
1546  006420  104021                    32300   2$:     AREAD                   ;ENABLE RECEIVER
1547  006422  000005                    32400           RESET                   ;RESET
1548  006424  105777  172162            32500           TSTB    @TKS            ;TEST DONE BIT
1549  006430  100001                    32600           BPL     3$              ;BRANCH IF DONE IS CLEARED
1550  006432  104001                    32700           EPROR                   ;NOT CLEARED, ERROR
1551  006434  104005                    32800   3$:     CHAIN                   ;CHAIN TO NEXT TEST
1552  006436  000770                    32900           BR      2$              ;REPEAT TEST
```

```
1554                                    33100   ;**********
1555                                    33200   ;AT17-- TEST#57--CHECKS THAT REFERENCING THE RECEIVER BUFFER
1556                                    33300   ;       CLEARS THE DONE BIT.
1557                                    33400   ;**********
1558                                    33500
1559  006440  000057                    33600   AT17:   57                      ;TEST NUMBER
1560  006442  006510                    33700           AT20                    ;NEXT TEST
1561  006444  000012                    33800           10.                     ;ITERATION COUNT
1562  006446  006450                    33900           1$                      ;SCOPE ENTRY
1563  006450  032777  001000  172236    34000   1$:     BIT     #LSI11,@SR      ;CHECK FOR LSI-11
1564  006456  001012                    34100           BNE     3$              ;SKIP TEST IF SET
1565  006460  012700  000226            34200           MOV     #226,R0
1566  006464  104010                    34300           DELAY                   ;DELAY 150 MSEC.
1567  006466  104021                    34400   2$:     AREAD                   ;ENABLE RECEIVER
1568  006470  105777  172120            34500           TSTB    @TKB            ;REFERENCE RECEIVER BUFFER
1569  006474  105777  172112            34600           TSTB    @TKS            ;TEST DONE BIT
1570  006500  100001                    34700           BPL     3$              ;BRANCH IF NOT SET
1571  006502  104001                    34800           ERROR                   ;DONE BIT IS SET, ERROR
1572  006504  104005                    34900   3$:     CHAIN                   ;CHAIN TO NEXT TEST
1573  006506  000767                    35000           BR      2$              ;REPEAT TEST
1574                                    35100
1575                                    35200   ;**********
1576                                    35300   ;AT20-- TEST#60--CHECK THAT THE RECEIVER DONE BIT IS ABLE TO
1577                                    35400   ;       CAUSE AN INTERRUPT.
1578                                    35500   ;**********
1579                                    35600
1580  006510  000060                    35700   AT20:   60                      ;TEST NUMBER
1581  006512  006602                    35800           AT21                    ;NEXT TEST
1582  006514  000012                    35900           10.                     ;ITERATION COUNT
1583  006516  006544                    36000           2$                      ;SCOPE ENTRY
1584  006520  104003                    36100           STRDRV                  ;SET UP RECEIVER INTERRUPT
1585  006522  006574                    36200           4$                      ;VECTOR TO 3$
1586  006524  032777  001000  172162    36300   1$:     BIT     #LSI11,@SR      ;CHECK FOR LSI-11
1587  006532  001021                    36400           BNE     5$              ;SKIP TEST IF SET
1588  006534  012700  000226            36500           MOV     #226,R0
1589  006540  104010                    36600           DELAY                   ;DELAY 150 MSEC
1590  006542  104021                    36700           AREAD                   ;ENABLE RECEIVER
1591  006544  005077  172042            36800   2$:     CLR     @TKS            ;DISABLE RECEIVER INTERRUPTS
1592  006550  005046                    36900           CLR     -(SP)           ;SET PROCESS STATUS TO ZERO
1593  006552  012746  006560            37000           MOV     #3$,-(SP)
1594  006556  000002                    37100           RTI
1595  006560  052777  000100  172024    37200   3$:     BIS     #BIT6,@TKS      ;ENABLE RECEIVER INTERRUPT
1596  006566  000240                    37300           NOP
1597  006570  104001                    37400           ERROR                   ;ERR,RECEIVER FAILED TO INTERRUPT
1598  006572  000401                    37500           BR      5$              ;CHAIN TO NEXT TEST
1599  006574  022626                    37600   4$:     POPSP2                  ;OK, CLEAN STACK
1600  006576  104005                    37700   5$:     CHAIN                   ;CHAIN TO NEXT TEST
1601  006600  000761                    37800           BR      2$              ;REPEAT TEST
```

```
1603                                    38000   ;**********
1604                                    38100   ;AT21-- TEST#61--TESTS THAT THE RECEIVER DONE DOES NOT CAUSE AN
1605                                    38200   ;       INTERRUPT WHEN THE PROCESSOR IS AT THE SAME LEVEL AS
1606                                    38300   ;       THE RECEIVER'S INTERRUPT REQUEST LEVEL.
1607                                    38400   ;**********
1608                                    38500
1609  006602  000061                    38600   AT21:   61                      ;TEST NUMBER
1610  006604  006702                    38700           AT22                    ;NEXT TEST
1611  006606  000012                    38800           10.                     ;ITERATION COUNT
1612  006610  006636                    38900           2$                      ;SCOPE ENTRY
1613  006612  104003                    39000           STRDRV                  ;SET RECEIVER VECTOR TO 4$
1614  006614  006674                    39100           5$
1615  006616  032777  001000  172070    39200   1$:     BIT     #LSI11,@SR      ;CHECK FOR LSI-11
1616  006624  001017                    39300           BNE     4$              ;SKIP TEST IF SET
1617  006626  012700  000226            39400           MOV     #226,R0
1618  006632  104010                    39500           DELAY                   ;DELAY 150 MSEC
1619  006634  104021                    39600           AREAD                   ;ENABLE RECEIVER
1620  006636  005077  171750            39700   2$:     CLR     @TKS            ;DISABLE RECEIVER INTERRUPTS
1621  006642  013746  000624            39800           MOV     TKLVL,-(SP)     ;SET PROCESSOR TO SAME LEVEL AS RECEIVER
1622  006646  012746  006654            39900           MOV     #3$,-(SP)
1623  006652  000002                    40000           RTI
1624  006654  052777  000100  171730    40100   3$:     BIS     #BIT6,@TKS      ;ENABLE RECEIVER INTERRUPTS
1625  006662  000240                    40200           NOP
1626  006664  005077  171722            40300   4$:     CLR     @TKS            ;OK, NO INTERRUPT OCCURED
1627  006670  104005                    40400           CHAIN                   ;CHAIN TO NEXT TEST
1628  006672  000770                    40500           BR      3$              ;REPEAT TEST
1629  006674  022626                    40600   5$:     POPSP2                  ;ERROR, RECEIVER INTERRUPTED, CLEAN STACK
1630  006676  104001                    40700           ERROR
1631  006700  000771                    40800           BR      4$              ;BRANCH 3$
```

```
                                    41000   ;**********
                                    41100   ;AT22-- TEST#62--TESTS THAT THE RECEIVER DONE DOES CAUSE AN
                                    41200   ;       INTERRUPT WHEN THE PROCESSOR IS AT A PRIORITY ONE
                                    41300   ;       LEVEL LOWER THAN THE RECEIVER'S INTERRUPT
                                    41400   ;       REQUEST LEVEL
                                    41500   ;**********
                                    41600
006702  000062                      41700   AT22:   62                      ;TEST NUMBER
006704  007010                      41800           AT23                    ;NEXT TEST
006706  000012                      41900           10.                     ;ITERATION COUNT
006710  006736                      42000           1$                      ;SCOPE ENTRY
006712  104003                      42100           STRDRV                  ;SET RECEIVER INTERRUPT
006714  006776                      42200           3$                      ;VECTOR TO 2$
006716  032777  001000  171770      42300           BIT     #LSI11,@SR      ;CHECK FOR LSI-11
006724  001025                      42400           BNE     4$              ;SKIP TEST IF SET
006726  012700  000226              42500           MOV     #226,R0
006732  104010                      42600           DELAY                   ;DELAY 150 MSEC
006734  104021                      42700           AREAD                   ;ENABLE RECEIVER
006736  005077  171650              42800   1$:     CLR     @TKS            ;DISABLE READER INTERRUPTS
006742  013746  000624              42900           MOV     TKLVL,-(SP)     ;SET PROCESSOR ONE LEVEL
006746  012746  006754              43000           MOV     #2$,-(SP)
006752  000002                      43100           RTI
006754  162737  000040  177776      43200   2$:     SUB     #40,PSW         ;LOWER THAN READERS
006762  052777  000100  171622      43300           BIS     #BIT6,@TKS      ;ENABLE INTERRUPTS
006770  000240                      43400           NOP
006772  104001                      43500           ERROR                   ;FAILED TO INTERRUPT
006774  000401                      43600           BR      4$              ;CHAIN TO NEXT TEST
006776  022626                      43700   3$:     POPSP2                  ;OK, CLEAN STACK
007000  005077  171606              43800   4$:     CLR     @TKS            ;DISABLE RECEIVER INTERRUPTS
007004  104005                      43900           CHAIN                   ;CHAIN TO NEXT TEST
007006  000753                      44000           BR      1$              ;REPEAT TEST
```

```
                                        44200   ;**********
                                        44300   ;AT23-- TEST#63--CHECKS THAT THE RECEIVER DONE DOES NOT
                                        44400   ;       REINTERRUPT AFTER RTI INSTRUCTION WHEN DONE
                                        44500   ;       BIT IS LEFT SET.
                                        44600   ;**********
                                        44700
007010  000063                          44800   AT23:   63                      ;TEST NUMBER
007012  007122                          44900           AT24                    ;NEXT TEST
007014  000012                          45000           10.                     ;ITERATION COUNT
007016  007030                          45100           1$                      ;SCOPE ENTRY
007020  032777  001000  171666          45200           BIT     #LSI11,@SR      ;CHECK FOR LSI-11
007026  001015                          45300           BNE     2$              ;SKIP TEST IF SET
007030  012700  000226                  45400   1$:     MOV     #226,R0
007034  104010                          45500           DELAY                   ;DELAY 150 MSEC
007036  104021                          45600           AREAD                   ;ENABLE RECEIVER
007040  104003                          45700           STRDRV                  ;SET RECEIVER INTERRUPT
007042  007074                          45800           3$                      ;VECTOR TO 3$
007044  005077  171542                  45900           CLR     @TKS            ;DISABLE RECEIVER INTERRUPTS
007050  052777  000100  171534          46000           BIS     #BIT6,@TKS      ;ENABLE RECEIVER INTERRUPT
007056  000240                          46100           NOP                     ;
007060  104001                          46200           ERROR                   ;NO INTERRUPT, ERROR
007062  005077  171524                  46300   2$:     CLR     @TKS            ;DISABLE RECEIVER INTERRUPTS
007066  000005                          46400           RESET                   ;RESET AFTER LAST INTF TEST.
007070  104005                          46500           CHAIN                   ;CHAIN TO NEXT TEST
007072  000756                          46600           BR      1$              ;REPEAT TEST
007074  012777  007114  171520          46700   3$:     MOV     #5$,@TKVTR      ;INTERRUPT, OK, CHANGE VECTOR TO 5$
007102  012716  007110                  46800           MOV     #4$,@SP         ;CHANGE RET ADDR. TO 4$
007106  000002                          46900           RTI                     ;RETURN
007110  000240                          47000   4$:     NOP
007112  000763                          47100           BR      2$              ;OK, NO ADDITIONAL INTERRUPTS
007114  022626                          47200   5$:     POPSP2                  ;ERROR, ADDITIONAL INTERRUPT
007116  104001                          47300           ERROR
007120  000760                          47400           BR      2$              ;CHAIN TO NEXT TEST
```

```
1699                                    47600   ;**********
1700                                    47700   ;AT24--TEST#64--HAVE OPERATOR TYPE A CHARACTER ON THE.
1701                                    47800   ;               KEYBOARD, THEN CKECK FOR RECIEVER DONE
1702                                    47900   ;
1703                                    48000   ;               ALLOW 12 SECONDS FOR OPERATOR RESPONSE.
1704                                    48100   ;**********
1705                                    48200
1706  007122  000064                    48300   AT24:   64                      ;TEST NUMBER
1707  007124  007222                    48400           AT25                    ;NEXT TEST
1708  007126  000001                    48500           1                       ;ITERATION COUNT
1709  007130  007152                    48600           2$                      ;SCOPE ENTRY
1710  007132  032777  001000  171554    48700   1$:     BIT     #LSI11,@SR      ;SKIP TEST IF NOT AN LSI-11
1711  007140  001426                    48800           BEQ     5$
1712  007142  005777  171444            48900           TST     @TKS            ;SHOULD BE CLEAR
1713  007146  001401                    49000           BEQ     2$
1714  007150  104001                    49100           ERROR                   ;RECIEVER STATUS NOT =0
1715  007152  012700  000600            49200   2$:     MOV     #600,R0         ;1/2 SEC DELAY
1716  007156  012737  000030  000716    49300           MOV     #30,CNTR        ;SET UP FOR 12 SEC WAIT
1717  007164  104000                    49400           TYPE
1718  007166  014364                    49500           OPMSG                   ;MESSAGE TO TYPE A CHAR
1719  007170  104010                    49600   3$:     DELAY                   ;1/2 SECOND
1720  007172  105777  171414            49700           TSTB    @TKS            ;CHECK DONE BIT
1721  007176  100407                    49800           BMI     5$              ;SET - EXIT LOOP
1722  007200  005337  000716            49900           DEC     CNTR
1723  007204  001403                    50000           BEQ     4$              ;TIME HAS RUN OUT...
1724  007206  012700  177777            50100           MOV     #-1,R0          ;ANOTHER 1/2 SEC
1725  007212  000766                    50200           BR      3$              ;CONTINUE WAIT
1726  007214  104001                    50300   4$:     ERROR                   ;NO RECIEVER DONE, OR
1727                                    50400                                   ;OPERATOR IS OUT TO LUNCH.
1728  007216  104005                    50500   5$:     CHAIN                   ;CHAIN TO NEXT TEST
1729  007220  000754                    50600           BR      2$
```

```
1731                                    50800   ;**********
1732                                    50900   ;AT25--TEST#65--CHECK THAT RECIEVER DONE CAUSES AN INTERRUPT
1733                                    51000   ;               WHEN BIT 6 (INTERRUPT ENABLE) IS SET.
1734                                    51100   ;**********
1735                                    51200
1736  007222  000065                    51300   AT25:   65                      ;TEST NUMBER
1737  007224  007300                    51400           AT26                    ;NEXT TEST
1738  007226  000001                    51500           1                       ;ITERATION COUNT
1739  007230  007242                    51600           2$                      ;SCOPE ENTRY
1740  007232  032777  001000  171454    51700   1$:     BIT     #LSI11,@SR      ;SKIP TEST IF NOT AN LSI-11
1741  007240  001414                    51800           BEQ     5$      ;
1742  007242  105777  171344            51900   2$:     TSTB    @TKS            ;DONE SHOULD BE SET
1743  007246  001001                    52000           BNE     3$
1744  007250  104001                    52100           ERROR                   ;RECIEVER DONE NOT SET
1745  007252  104003                    52200   3$:     STRDRV                  ;SET RECIEVER INTERRUPT
1746  007254  007272                    52300           5$                      ;VECTOR TO 5$
1747  007256  052777  000100  171326    52400           BIS     #BIT6,@TKS      ;ENABLE INTERRUPT
1748  007264  000240                    52500           NOP
1749  007266  000240                    52600           NOP
1750  007270  104001                    52700   4$:     ERROR                   ;RECIEVER DIDN'T INTERRUPT
1751  007272  022626                    52800   5$:     POPSP2                  ;CLEAN UP THE STACK
1752  007274  104005                    52900           CHAIN                   ;CHAIN TO NEXT TEST
1753  007276  000761                    53000           BR      2$
```

```
1755                                    53200   ;**********
1756                                    53300   ;AT26--TEST#66--CHECK THAT READING TKB CLEARS DONE BIT
1757                                    53400   ;               AND THAT DONE CLEARED CON'T CAUSE AN INTERRUPT
1758                                    53500   ;**********
1759                                    53600
1760  007300  000066                    53700   AT26:   66                      ;TEST NUMBER
1761  007302  177777                    53800           -1                      ;LAST TEST
1762  007304  000001                    53900           1                       ;ITERATION COUNT
1763  007306  007320                    54000           2$                      ;SCOPE ENTRY
1764  007310  032777  001000  171376    54100   1$:     BIT     #LSI11,@SR      ;SKIP TEST IF NOT AN LSI-11
1765  007316  001422                    54200           BEQ     5$
1766  007320  105777  171266            54300   2$:     TSTB    @TKS            ;MAKE SHURE DONE IS STILL SET
1767  007324  001001                    54400           BNE     3$
1768  007326  104001                    54500           ERROR                   ;RECIEVER DONE NOT SET
1769  007330  017737  171260  000716    54600   3$:     MOV     @TKB,CNTR       ;READ DATA BUFFER
1770  007336  105777  171250            54700           TSTB    @TKS            ;CHECK THE DONE BIT
1771  007342  100001                    54800           BPL     4$              ;OK
1772  007344  104001                    54900           ERROR                   ;DEADING DATA REG DIDN'T CLEAR DONE
1773  007346  104003                    55000   4$:     STRDRV                  ;SET RECIEVER INTERRUPT
1774  007350  007374                    55100           6$                      ;VECTOR TO 6$
1775  007352  052777  000100  171232    55200           BIS     #BIT6,@TKS      ;ENABLE INTERRUPT
1776  007360  000240                    55300           NOP
1777  007362  000240                    55400           NOP
1778  007364  005077  171222            55500   5$:     CLR     @TKS            ;OK- CLEAN UP
1779  007370  104005                    55600           CHAIN                   ;EXIT TESTS
1780  007372  000752                    55700           BR      2$
1781  007374  104001                    55800   6$:     ERROR                   ;DLV INTERRUPTED WITH DONE CLEAR
1782  007376  022626                    55900           POPSP2                  ;CLEAN UP THE STACK
1783  007400  000771                    56000           BR      5$
```

```
1785
```

```
1786                                    00100          .SBTTL  LA36 PRINTER TESTS
1787                                    00200  ;
1788                                    00300  ;THE LA36 PRINTER TESTS WILL BE EXECUTED IN A
1789                                    00400  ;CONTINOUS LOOP OUTPUTTING TO ALL MULTIPLE DL11'S
1790                                    00500  ;IF SR BIT 8 IS SET TO ZERO AT START UP TIME.  IF
1791                                    00600  ;BIT 8 IS SET TO 1 AT START UP THEY MAY BE EXECUTED
1792                                    00700  ;INDIVIDUALLY ONCE OR CONTINUALLY LOOPED, OR
1793                                    00800  ;BECOME THE FIRST OF THE ENTIRE SEQUENCE OF PRINTER
1794                                    00900  ;TESTS.  REFERENCE INTRUCTIONS IN THE INTRODUCTION
1795                                    01000  ;FOR PROPER MODE OF OPERATION.
1796                                    01100  ;
1797                                    01200  ;
1798                                    01300  ;XXXXXXXXXX
1799                                    01400  ;
1800                                    01500  ;PTO -- DATA PATH TEST---FOUR LINES OF ALTERNATING
1801                                    01600  ;       "*" AND "U" ARE PRINTED, OUT TO THE GIVEN PAPER
1802                                    01700  ;       WIDTH.  THE PATTERN WILL APPEAR AS FOLLOWS.
1803                                    01800  ;
1804                                    01900  ;       *U*U*U*U*U*U
1805                                    02000  ;       U*U*U*U*U*U*
1806                                    02100  ;       *U*U*U*U*U*U
1807                                    02200  ;       U*U*U*U*U*U*
1808                                    02300  ;
1809                                    02400  ;XXXXXXXXXX
1810                                    02500  ;
1811  007402  000000                    02600  PTO:    0                       ;TEST NUMBER
1812  007404  007456                    02700          PT1                     ;NEXT TEST
1813  007406  104016                    02800          PRTHDR                  ;
1814  007410  104007                    02900          TYPEM                   ;PRINT COLUMN # MESG
1815  007412  014112                    03000          HDR0
1816  007414  012703  025125            03100  1$:     MOV     #"U*,R3         ;SET FIRST CHAR PAIR
1817  007420  012702  000004            03200          MOV     #4,R2           ;SET LINE COUNT
1818  007424  010300                    03300  2$:     MOV     R3,R0           ;SET CHAR PAIR
1819  007426  013701  000652            03400          MOV     WIDTH,R1        ;SET COLUMN COUNT
1820  007432  104015                    03500  3$:     PRINTC                  ;PRINT CHAR
1821  007434  000300                    03600          SWAB    R0              ;SET NEXT CHAR
1822  007436  005301                    03700          DEC     R1              ;DEC COLUMN COUNT
1823  007440  001374                    03800          BNE     3$              ;FINISH LINE
1824  007442  000303                    03900          SWAB    R3              ;SET NEXT LINE START CHAR
1825  007444  104012                    04000          CRLF                    ;SEND CR-LF
1826  007446  005302                    04100          DEC     R2              ;DEC LINE COUNT
1827  007450  001365                    04200          BNE     2$              ;FINISH TEST
1828  007452  104005                    04300          CHAIN                   ;ALL DONE, EXIT
1829  007454  000757                    04400          BR      1$              ;REPEAT TEST
```

```
1831                                    04600   ;XXXXXXXXXX
1832                                    04700   ;
1833                                    04800   ;PT1 -- PRINTER CHARACTER TEST --- PRINTS ALL PRINTABLE CHARACTERS
1834                                    04900   ;
1835                                    05000   ;XXXXXXXXXX
1836                                    05100
1837  007456  000001                    05200   PT1:    1                       ;TEST NUMBER
1838  007460  007600                    05300           PT2                     ;NEXT TEST
1839  007462  104016                    05400           PRTHDR
1840  007464  012701  000040            05500   1$:     MOV     #40,R1          ;SPACE TO R1
1841  007470  012702  000100            05600           MOV     #100,R2         ;@ TO R2
1842  007474  012703  000140            05700           MOV     #140,R3         ;  TO R3
1843  007500  110100                    05800   2$:     MOVB    R1,R0           ;CHAR TO R0
1844  007502  004737  007550            05900           JSR     PC,SPSP         ;SEND TWO SPACES
1845  007506  110200                    06000           MOVB    R2,R0           ;NEXT CHAR TO R0
1846  007510  004737  007550            06100           JSR     PC,SPSP         ;SEND TWO SPACES
1847  007514  012704  000003            06200           MOV     #3,R4           ;PRINT COUNT TO R4
1848  007520  110300                    06300           MOVB    R3,R0           ;THIRD CHAR TO R0
1849  007522  104015                    06400   3$:     PRINTC                  ;PRINT THE CHAR
1850  007524  005304                    06500           DEC     R4              ;THREE TIMES ?
1851  007526  001375                    06600           BNE     3$              ;BRANCH IF NOT
1852  007530  104012                    06700           CRLF                    ;CARRIAGE RETURN LINE FEED
1853  007532  122122                    06800           CMPB    (R1)+,(R2)+     ;NEXT CHARACTERS
1854  007534  105723                    06900           TSTB    (R3)+
1855  007536  020327  000200            07000           CMP     R3,#200         ;CHECK IF ALL DONE
1856  007542  103756                    07100           BLO     2$              ;BRANCH IF NOT
1857  007544  104005                    07200           CHAIN                   ;EXIT TO NEXT TEST
1858  007546  000746                    07300           BR      1$              ;REPEAT TEST
1859  007550  012704  000003            07400   SPSP:   MOV     #3,R4           ;PRINT COUNT TO R4
1860  007554  104015                    07500   1$:     PRINTC                  ;PRINT CHAR
1861  007556  005304                    07600           DEC     R4              ;THREE TIMES?
1862  007560  001375                    07700           BNE     1$              ;BRANCH IF NOT
1863  007562  012700  000040            07800   SP2:    MOV     #40,R0          ;SPACE TO R0
1864  007566  104015                    07900           PRINTC                  ;SEND A SPACE
1865  007570  012700  000040            08000   SPC:    MOV     #40,R0          ;SPACE TO R0
1866  007574  104015                    08100           PRINTC                  ;SEND ANOTHER
1867  007576  000207                    08200           RTS     PC              ;RETURN
```

```
1869                                    08400   ;XXXXXXXXXX
1870                                    08500   ;
1871                                    08600   ;PT2 -- NON-PRINTING CHARACTER TEST.  THIS TEST
1872                                    08700   ;       PRINTS THE OCTAL CODE FOLLOWED BY THE MNEMONIC
1873                                    08800   ;       OF ALL NON-PRINTING CHARACTERS.  FOLLOWING EACH
1874                                    08900   ;       MNEMONIC, THE PRINTER IS DRIVEN BY THE NON-PRINTING
1875                                    09000   ;       CODE (000 THROUGH 037 PLUS 177)
1876                                    09100   ;     : ALL CONTROL CHARACTERS (INCLUDING THOSE FOR OPTIONS
1877                                    09200   ;       WILL BE SKIPPED, REFER TO THE DOCUMENT FOR A LIST OF THOSE
1878                                    09300   ;       TESTED.
1879                                    09400   ;
1880                                    09500   ;XXXXXXXXXX
1881                                    09600   ;
1882  007600  000002                   09700   PT2:    2                       ;TEST NUMBER
1883  007602  010164                   09800           PT3                     ;NEXT TEST
1884  007604  104016                   09900           PRTHDR                  ;PRINT TEST HEADER
1885  007606  012701  007706           10000   1$:     MOV     #IDEZ,R1        ;ADDR OF IDENT TO R1
1886  007612  012703  010137           10100           MOV     #NPCODE,R3      ;ADDR OF NON-PRINT-CODES TO R3
1887  007616  012702  000003           10200   2$:     MOV     #3,R2           ;NO. OF ID'S PER LINE TO R2
1888  007622  012704  000010           10300   3$:     MOV     #10,R4          ;NO. OF CHARS PER ID TO R4
1889  007626  121327  000055           10400   4$:     CMPB    (R3),#55        ;ZERO TERMINATOR IN NP TABLE??
1890  007632  001422                   10500           BEQ     7$              ;BRANCH IF YES
1891  007634  112100                   10600           MOVB    (R1)+,R0        ;GET ID CHARACTERS
1892  007636  104015                   10700           PRINTC                  ;AND PRINT A
1893  007640  005304                   10800           DEC     R4              ;GROUP OF
1894  007642  001371                   10900           BNE     4$              ;8 CHARACTERS
1895  007644  112300                   11000           MOVB    (R3)+,R0        ;GET NP CODE FROM TABLE
1896  007646  012704  000003           11100           MOV     #3,R4           ;AND
1897  007652  104015                   11200   5$:     PRINTC                  ;TRY TO PRINT IT
1898  007654  005304                   11300           DEC     R4              ;THREE
1899  007656  001375                   11400           BNE     5$              ;TIMES
1900  007660  005302                   11500           DEC     R2              ;MORE TO GO ON THIS LINE ?
1901  007662  001404                   11600           BEQ     6$              ;BRANCH IF NO
1902  007664  004737  007562           11700           JSR     PC,SP2          ;SEND 3 SPACES
1903  007670  104015                   11800           PRINTC
1904  007672  000753                   11900           BR      3$              ;BRANCH TO CONTINUE LINE
1905  007674  104012                   12000   6$:     CRLF
1906  007676  000747                   12100           BR      2$              ;GO DO NEXT LINE
1907  007700  104012                   12200   7$:     CRLF
1908  007702  104005                   12300           CHAIN                   ;CHAIN TO NEXT TEST
1909  007704  000740                   12400           BR      1$
1910                                    12500
1911                                    12600
1912                                    12700
1913  007706  030060  020060  047040   12800   IDEZ:   .ASCII  /000 NUL002 STX/
      007714  046125  030060  020062
      007722  051440  054124
1914  007726  030060  020066  040440   12900           .ASCII  /006 ACK020 DLE021 DC1/
      007734  045503  031060  020060
      007742  042040  042514  031060
      007750  020061  042040  030503
1915  007756  031060  020062  042040   13000           .ASCII  /022 DC2023 DC3024 DC4/
      007764  031103  031060  020063
      007772  042040  031503  031060
      010000  020064  042040  032103
1916  010006  031060  020065  047040   13100           .ASCII  /025 NAK026 SYN027 ETB/
```

```
                010014  045501  031060  020066
                010022  051440  047131  031060
                010030  020067  042440  041124
1917            010036  031460  020060  041440  13200           .ASCII  /030  CAN031  EM 032  SUB/
                010044  047101  031460  020061
                010052  042440  020115  031460
                010060  020062  051440  041125
1918            010066  031460  020064  043040  13300           .ASCII  /034  FS 035  GS 036  RS /
                010074  020123  031460  020065
                010102  043440  020123  031460
                010110  020066  051040  020123
1919            010116  031460  020067  052440  13400           .ASCII  /037  US 177  DEL /
                010124  020123  033461  020067
                010132  042040  046105     040
1920            010137     000     002     006  13500  NPCODE:  .BYTE   0,2,6,20,21,22,23,24
                010142     020     021     022
                010145     023     024
1921            010147     025     025     027  13600           .BYTE   25,25,27,30,31,32,34,35
                010152     030     031     032
                010155     034     035
1922            010157     036     037     177  13700           .BYTE   36,37,177,55
                010162     055
1923                    010164                   13800           .EVEN
```

```
1925                                    14000   ;XXXXXXXXXX
1926                                    14100   ;
1927                                    14200   ;PT3 -- CARRIAGE RETURN TEST
1928                                    14300   ;
1929                                    14400   ;       THE LINE CONSISTS OF A STRING OF 0'S AND
1930                                    14500   ;       X'S.  FIRST, THE 0'S ARE PRINTED OUT TO THE LAST
1931                                    14600   ;       COLUMN WITH A SPACE SEPARATING EACH.  THEN THE
1932                                    14700   ;       CARRIAGE IS SPACED TO THE FIRST BLANK SPACE, AN X
1933                                    14800   ;       IS PRINTED AND THEN RETURNED TO THE MARGIN.  THIS
1934                                    14900   ;       PROCESS IS CONTINUE UNTIL ALL SPACES BETWEEN
1935                                    15000   ;       THE ZEROES HAVE BEEN FILLED.
1936                                    15100   ;
1937                                    15200   ;XXXXXXXXXX
1938                                    15300
1939  010164  000003                   15400   PT3:    3                       ;TEST NUMBER
1940  010166  010304                   15500           PT4                     ;NEXT TEST
1941  010170  104016                   15600           PRTHDR                  ;TYPE HEADER
1942  010172  005037  000674           15700   1$:     CLR     SPCNT           ;CLEAR SPACE COUNTER
1943  010176  013701  000652           15800           MOV     WIDTH,R1        ;POSITION COUNTER TO R1
1944  010202  012700  000117           15900   2$:     MOV     #117,R0         ;"O" TO R0
1945  010206  104015                   16000           PRINTC                  ;PRINT THE "O"
1946  010210  005301                   16100           DEC     R1              ;DECREMENT POSITION COUNTER
1947  010212  001404                   16200           BEQ     3$              ;BRANCH IF 0
1948  010214  004737  007570           16300           JSR     PC,SPC          ;SEND SPACE
1949  010220  005301                   16400           DEC     R1              ;DECREMENT POSITION COUNTER
1950  010222  001367                   16500           BNE     2$              ;BRANCH IF NOT ZERO
1951  010224  104022                   16600   3$:     CR                      ;SEND A CR
1952  010226  012737  000001  000674   16700           MOV     #1,SPCNT        ;SPACE, COUNTER SET TO 1
1953  010234  013701  000674           16800   4$:     MOV     SPCNT,R1        ;NO. OF SPACES TO R1
1954  010240  004737  007570           16900   5$:     JSR     PC,SPC          ;SEND SPACE
1955  010244  005301                   17000           DEC     R1              ;DECREMENT SPACE COUNTER
1956  010246  001374                   17100           BNE     5$              ;BRANCH IF NOT ZERO
1957  010250  012700  000130           17200           MOV     #130,R0         ;"X" INTO R0
1958  010254  104015                   17300           PRINTC                  ;PRINT "X"
1959  010256  104022                   17400           CR                      ;PRINT CR
1960  010260  062737  000002  000674   17500           ADD     #2,SPCNT        ;INCREMENT SPACE COUNT BY 2
1961  010266  023737  000674  000652   17600           CMP     SPCNT,WIDTH     ;COMPARE POSITION COUNTER WITH COLM. COUNT
1962  010274  103757                   17700           BLO     4$              ;BRANCH IF LOWER
1963  010276  104014                   17800           LF                      ;SEND LF
1964  010300  104005                   17900           CHAIN                   ;CHAIN TO NEXT TEST
1965  010302  000733                   18000           BR      1$              ;REPEAT TEST
```

```
1967                                    18200   ;XXXXXXXXXX
1968                                    18300   ;
1969                                    18400   ;PT4 -- MULTIPLE LINE FEED TEST -- 63 LINE FEEDS ARE
1970                                    18500   ;       SENT WITH A REFERENCE LINE AT THE START AND END.,
1971                                    18600   ;       A NUMBER IS PRINTED WHICH INDICATES THE NUMBER OF LINE
1972                                    18700   ;       FEEDS THAT WILL BE ISSUED BEFORE THE NEXT
1973                                    18800   ;       NUMBER OR REFERENCE LINE IS PRINTED.
1974                                    18900   ;
1975                                    19000   ;XXXXXXXXXX
1976                                    19100
1977  010304  000004                    19200   PT4:    4                       ;TEST NUMBER
1978  010306  010462                    19300           PT5                     ;NEXT TEST
1979  010310  104016                    19400           PRTHDR                  ;TYPE HEADER
1980  010312  012737  000001  000706    19500   1$:     MOV     #1,LFCNT        ;LINE FEED COUNT TO 1
1981  010320  013701  000652            19600           MOV     WIDTH,R1        ;COLUMN COUNT TO R1
1982  010324  012702  010444            19700           MOV     #LINE3,R2       ;ADDR OF NUMBER FIELD TO R2
1983  010330  004737  010414            19800           JSR     PC,REF          ;PRINT REFERENCE LINE
1984  010334  013701  000706            19900   2$:     MOV     LFCNT,R1        ;LINE FEED COUNT TO R1
1985  010340  104014                    20000   3$:     LF                      ;SEND LF
1986  010342  005301                    20100           DEC     R1              ;DECREMENT COUNTER
1987  010344  001375                    20200           BNE     3$              ;BRANCH IF NOT YET 0
1988  010346  006337  000706            20300           ASL     LFCNT           ;DOUBLE LINE FEED COUNT
1989  010352  022737  000100  000706    20400           CMP     #BIT6,LFCNT     ;TEST IF COUNT IS 32
1990  010360  001406                    20500           BEQ     4$              ;BRANCH IF =32, END
1991  010362  112200                    20600           MOVB    (R2)+,R0        ;NUMBER TO R0
1992  010364  104015                    20700           PRINTC                  ;PRINT IT
1993  010366  112200                    20800           MOVB    (R2)+,R0        ;NUMBER TO R0
1994  010370  104015                    20900           PRINTC                  ;PRINT IT
1995  010372  104022                    21000           CR                      ;PRINT CR
1996  010374  000757                    21100           BR      2$              ;DRIVE THE LINEFEEDS
1997  010376  013701  000652            21200   4$:     MOV     WIDTH,R1        ;COLUMN COUNT TO R1
1998  010402  004737  010414            21300           JSR     PC,REF          ;SEND END REFERENCE LINE
1999  010406  104014                    21400           LF                      ;ADVANCE PAPER
2000  010410  104005                    21500           CHAIN
2001  010412  000737                    21600           BR      1$              ;REPEAT TEST
2002  010414  112200                    21700   REF:    MOVB    (R2)+,R0        ;NUMBER TO R0
2003  010416  104015                    21800           PRINTC                  ;PRINT IT
2004  010420  112200                    21900           MOVB    (R2)+,R0        ;NUMBER TO R0
2005  010422  104015                    22000           PRINTC                  ;PRINT IT
2006  010424  005741                    22100           TST     -(R1)           ;DECREASE COUNTER BY 2
2007  010426  012700  000137            22200           MOV     #137,R0         ;DASH (-) TO R0
2008  010432  104015                    22300   1$:     PRINTC                  ;PRINT IT
2009  010434  005301                    22400           DEC     R1              ;DECREMENT COLUMN COUNTER
2010  010436  001375                    22500           BNE     1$              ;BRANCH IF NO ZERO
2011  010440  104022                    22600           CR                      ;PRINT CR
2012  010442  000207                    22700           RTS     PC              ;RETURN
2013                                    22800
2014  010444  030460  031060  032060    22900   LINE3:  .ASCII  /01020408163200/
      010452  034060  033061  031063
      010460  030060
```

```
2016                                    23100   ;XXXXXXXXXX
2017                                    23200   ;PT5--  SINGLE LINE FEED TEST --  TESTS THE LINE FEED
2018                                    23300   ;       CAPABILITY FROM ALL COLUMNS.
2019                                    23400   ;XXXXXXXXXX
2020                                    23500
2021  010462  000005                    23600   PT5:    5                       ;TEST NUMBER
2022  010464  010666                    23700           PT6                     ;NEXT TEST
2023  010466  104016                    23800           PRTHDR                  ;TYPE HEADER
2024  010470  013701  000652            23900   1$:     MOV     WIDTH,R1        ;COLUMN COUNT TO R1
2025  010474  005741                    24000           TST     -(R1)           ;DECREASE BY 2
2026  010476  012700  000060            24100           MOV     #60,R0          ;'0' TO R0
2027  010502  104015                    24200   2$:     PRINTC                  ;SEND 0
2028  010504  005301                    24300           DEC     R1              ;DECREMENT COLUMN COUNTER
2029  010506  001375                    24400           BNE     2$              ;BRANCH IF NOT ZERO
2030  010510  012700  000062            24500           MOV     #62,R0          ;SEND A 2
2031  010514  104015                    24600           PRINTC
2032  010516  104015                    24700           PRINTC                  ;SEND A SECOND TWO
2033  010520  023727  000652  000204    24800           CMP     WIDTH,#132.     ;COMPARE COLUMN COUNT
2034  010526  001404                    24900           BEQ     3$              ;BRANCH IF EQ 132
2035  010530  012700  003410            25000           MOV     #3410,R0        ;DELAY 1.8 SEC
2036  010534  104010                    25100           DELAY
2037  010536  000407                    25200           BR      5$
2038  010540  012700  000063            25300   3$:     MOV     #63,R0          ;3'S TO R0
2039  010544  012701  000100            25400           MOV     #100,R1         ;64 TO COUNTER
2040  010550  104015                    25500   4$:     PRINTC                  ;SEND CHARACTER
2041  010552  005301                    25600           DEC     R1              ;DECREMENT COUNT
2042  010554  001375                    25700           BNE     4$              ;BRANCH IF NOT ZERO
2043  010556  104012                    25800   5$:     CRLF                    ;SEND A CR,LF
2044  010560  013701  000652            25900           MOV     WIDTH,R1        ;NO.  COLUMNS TO R1
2045  010564  012700  000134            26000   6$:     MOV     #134,R0         ;BACKSLASH TO R0
2046  010570  104015                    26100           PRINTC                  ;SEND IT
2047  010572  104014                    26200           LF                      ;PRINT LF
2048  010574  005301                    26300           DEC     R1              ;DECREMENT COUNTER
2049  010576  001372                    26400           BNE     6$              ;BRANCH IF NOT ZERO.
2050  010600  104022                    26500           CR                      ;SEND CR
2051  010602  004737  010630            26600           JSR     PC,PT5AL        ;SEND REF LINE #1
2052  010606  104012                    26700           CRLF                    ;SEND A CR,LF
2053  010610  012700  001750            26800           MOV     #1750,R0        ;DELAY 1 SEC
2054  010614  104010                    26900           DELAY
2055  010616  004737  010630            27000           JSR     PC,PT5AL        ;SEND A SECOND REF.  LINE
2056  010622  104012                    27100           CRLF                    ;SEND A CR,LF
2057  010624  104005                    27200           CHAIN                   ;CHAIN TO NEXT TEST
2058  010626  000720                    27300           BR      1$              ;REPEAT TEST
2059  010630  013701  000652            27400   PT5AL:  MOV     WIDTH,R1        ;COLUMN COUNT TO R1
2060  010634  012700  000061            27500           MOV     #61,R0          ;"1" TO R0
2061  010640  104015                    27600   1$:     PRINTC                  ;PRINT R0
2062  010642  005301                    27700           DEC     R1              ;DECREMENT COUNTER
2063  010644  001407                    27800           BEQ     2$              ;BRANCH IF=0
2064  010646  005200                    27900           INC     R0              ;INCREMENT CHARACTER
2065  010650  020027  000071            28000           CMP     R0,#71          ;COMP CHAR TO "9"
2066  010654  101771                    28100           BLOS    1$              ;BRANCH IF LOWER OR SAME
2067  010656  012700  000060            28200           MOV     #60,R0          ;RESET CHAR TO "0"
2068  010662  000766                    28300           BR      1$              ;CONTINUE
2069  010664  000207                    28400   2$:     RTS     PC              ;FINISHED, RETURN TO CALLER
```

```
2071                                    28600   ;XXXXXXXXXX
2072                                    28700   ;PT6--  BACKSPACE TEST -- A REFERENCE LINE SUCH AS IN
2073                                    28800   ;       TEST PT5 IS PRINTED.  THE SECOND LINE CONSISTS
2074                                    28900   ;       OF PRINTING A BACKSLASH, BACKSPACE AND FORWARD
2075                                    29000   ;       SLASH COMBINATION OUT TO THE GIVEN COLUMN WIDTH.
2076                                    29100   ;       THIS LINE IS THEN FOLLOWED BY THE SAME TWO REFERENCE
2077                                    29200   ;       LINES AS PRINTED IN TEST PT5.
2078                                    29300   ;XXXXXXXXXX
2079                                    29400
2080  010666  000006                    29500   PT6:    6                       ;TEST NUMBER
2081  010670  011054                    29600           PT7                     ;NEXT TEST
2082  010672  104016                    29700           PRTHDR                  ;PRINT HEADER
2083  010674  104007                    29800           TYPEM                   ;PRINT COLUMN # MESG
2084  010676  014112                    29900           HDR0
2085  010700  013701  000652            30000   1$:     MOV     WIDTH,R1        ;COLUMN COUNT TO R1
2086  010704  005741                    30100           TST     -(R1)           ;DECREMENT BY 2
2087  010706  012700  000060            30200           MOV     #60,R0          ;"0" TO R0
2088  010712  104015                    30300   2$:     PRINTC                  ;SEND 0
2089  010714  005301                    30400           DEC     R1              ;DECREMENT COLUMN COUNTER
2090  010716  001375                    30500           BNE     2$              ;BRANCH IF NOT ZERO
2091  010720  012700  000062            30600           MOV     #62,R0          ;"2" TO R0
2092  010724  104015                    30700           PRINTC                  ;SEND A "2"
2093  010726  104015                    30800           PRINTC                  ;SEND A SECOND "2"
2094  010730  023727  000652  000204    30900           CMP     WIDTH,#132.     ;COMPARE COLUMN COUNT
2095  010736  001404                    31000           BEQ     3$
2096  010740  012700  003410            31100           MOV     #3410,R0        ;DELAY 1.8 SEC
2097  010744  104010                    31200           DELAY
2098  010746  000407                    31300           BR      5$
2099  010750  012700  000063            31400   3$:     MOV     #63,R0          ;3'S TO R0
2100  010754  012701  000100            31500           MOV     #100,R1         ;64 TO CHAR COUNT
2101  010760  104015                    31600   4$:     PRINTC                  ;SEND CHAR
2102  010762  005301                    31700           DEC     R1              ;DECREMENT CHAR COUNT
2103  010764  001375                    31800           BNE     4$              ;CONTINUE IF NOT DONE
2104  010766  104012                    31900   5$:     CRLF                    ;SEND A CR,LF
2105  010770  013701  000652            32000           MOV     WIDTH,R1        ;COLUMN COUNT TO R1
2106  010774  012700  000134            32100   6$:     MOV     #134,R0         ;BACKSLASH TO R0
2107  011000  104015                    32200           PRINTC                  ;SEND IT
2108  011002  012700  000010            32300           MOV     #10,R0          ;BACKSPACE TO R0
2109  011006  104015                    32400           PRINTC                  ;SEND IT
2110  011010  012700  000057            32500           MOV     #57,R0          ;FORWORD SLASH TO R0
2111  011014  104015                    32600           PRINTC                  ;SEND IT
2112  011016  005301                    32700           DEC     R1              ;END OF PAPER
2113  011020  001365                    32800           BNE     6$              ;BRANCH IF NO
2114  011022  104014                    32900           LF                      ;SEND LF
2115  011024  104022                    33000           CR                      ;SEND CR
2116  011026  004737  010630            33100           JSR     PC,PT5AL        ;SEND REF LINE #1
2117  011032  104012                    33200           CRLF                    ;SEND A CR,LF
2118  011034  012700  001750            33300           MOV     #1750.R0        ;DELAY 1 SEC
2119  011040  104010                    33400           DELAY
2120  011042  004737  010630            33500           JSR     PC,PT5AL        ;SEND SECOND REF LINE
2121  011046  104012                    33600           CRLF                    ;SEND A CR,LF
2122  011050  104005                    33700           CHAIN                   ;CHAIN TO NEXT TEST
2123  011052  000712                    33800           BR      1$              ;REPEAT TEST
```

```
2125                                    34000   ;XXXXXXXXXX
2126                                    34100   ;
2127                                    34200   ;PT7--  OVERPRINT TEST-- A ROW OF ALTERNATING M'S AND
2128                                    34300   ;       SPACES ARE PRINTED, OUT TO THE LAST COLUMN AND OVERPRINTED TWICE.
2129                                    34400   ;       A SECOND LINE OF ALTERNATING SPACES AND "@'S" IS THEN
2130                                    34500   ;       SENT 3 TIMES AS THE FIRST LINE.  THIS IS FOLLOWED
2131                                    34600   ;       BY A THIRD AND FINAL LINE OF ALTERNATING '&'
2132                                    34700   ;       AND SPACES.
2133                                    34800   ;
2134                                    34900   ;XXXXXXXXXX
2135                                    35000
2136  011054  000007                    35100   PT7:    7                       ;TEST NUMBER
2137  011056  011266                    35200           PT10                    ;NEXT TEST
2138  011060  104016                    35300           PRTHDR                  ;PRINT MESSAGE
2139  011062  012703  000002            35400   1$:     MOV     #2,R3           ;2 COUNT TO R3
2140  011066  013701  000652            35500   2$:     MOV     WIDTH,R1        ;NO. OF COLUMNS TO R1
2141  011072  012700  000115            35600   3$:     MOV     #115,R0         ;"M" TO R0
2142  011076  104015                    35700           PRINTC                  ;SEND IT
2143  011100  005301                    35800           DEC     R1              ;END OF LINE
2144  011102  001404                    35900           BEQ     4$              ;BRANCH IF YES
2145  011104  004737  007570            36000           JSR     PC,SPC          ;SEND SPACE
2146  011110  005301                    36100           DEC     R1              ;END OF LINE?
2147  011112  001367                    36200           BNE     3$              ;BRANCH IF NO
2148  011114  022703  000002            36300   4$:     CMP     #2,R3           ;TEST R3
2149  011120  001003                    36400           BNE     6$              ;BRANCH IF NOT FIRST TIME
2150  011122  104022                    36500   5$:     CR                      ;SEND CR
2151  011124  005303                    36600           DEC     R3              ;DECREASE LINE COUNTER
2152  011126  000757                    36700           BR      2$              ;REPEAT LINE
2153  011130  005703                    36800   6$:     TST     R3              ;THIRD TIME?
2154  011132  001373                    36900           BNE     5$              ;BRANCH IF NOT
2155  011134  104012                    37000           CRLF                    ;NEXT LINE
2156  011136  005723                    37100           TST     (R3)+           ;REPEAT COUNTER TO R3
2157  011140  013701  000652            37200   7$:     MOV     WIDTH,R1        ;COLUMN COUNT TO R1
2158  011144  004737  007570            37300   8$:     JSR     PC,SPC          ;SEND SPACE
2159  011150  005301                    37400           DEC     R1              ;DECREASE COLUMN COUNT
2160  011152  001405                    37500           BEQ     9$              ;BRANCH IF 0, END OF LINE
2161  011154  012700  000100            37600           MOV     #100,R0         ;"@" TO R0
2162  011160  104015                    37700           PRINTC                  ;SEND IT
2163  011162  005301                    37800           DEC     R1              ;DECREASE COLUMN COUNT
2164  011164  001367                    37900           BNE     8$              ;BRANCH IF NOT 0 (NOT END)
2165  011166  022703  000002            38000   9$:     CMP     #2,R3           ;END OF LINE, FIRST TIME?
2166  011172  001003                    38100           BNE     11$             ;BRANCH IF NOT
2167  011174  104022                    38200   10$:    CR                      ;SEND CR
2168  011176  005303                    38300           DEC     R3              ;DECREASE LINE COUNTER
2169  011200  000757                    38400           BR      7$              ;REPEAT LINE
2170  011202  005703                    38500   11$:    TST     R3              ;TEST IF THIRD REPEAT
2171  011204  001373                    38600           BNE     10$             ;BRANCH IF NOT
2172  011206  104012                    38700           CRLF                    ;DO NEXT LINE
2173  011210  005723                    38800           TST     (R3)+           ;LINE REPEAT COUNTER TO R3
2174  011212  013701  000652            38900   12$:    MOV     WIDTH,R1        ;COLUMN COUNT TO R1
2175  011216  012700  000046            39000   13$:    MOV     #46,R0          ;"&" TO R0
2176  011222  104015                    39100           PRINTC                  ;SEND IT
2177  011224  005301                    39200           DEC     R1              ;DECREASE COLUMN COUNT
2178                                    39300
2179  011226  001404                    39400           BEQ     14$             ;BRANCH IF END
2180  011230  004737  007570            39500           JSR     PC,SPC          ;SEND SPACE
```

```
2181  011234  005301            39600          DEC     R1          ;DECREASE COLUMN COUNT
2182  011236  001367            39700          BNE     13$         ;BRANCH IF NOT END
2183  011240  022703  000002    39800  14$:    CMP     #2,R3       ;TEST IF FIRST TIME
2184  011244  001003            39900          BNE     16$         ;BRANCH IF =2, FIRST TIME
2185  011246  104022            40000  15$:    CR                  ;SENT CR
2186  011250  005303            40100          DEC     R3          ;DECREASE REPEAT COUNTER
2187  011252  000757            40200          BR      12$         ;PRINT LINE AGAIN
2188  011254  005703            40300  16$:    TST     R3          ;TEST IF END, R3=0
2189  011256  001373            40400          BNE     15$         ;BRANCH IF NOT END
2190  011260  104012            40500          CRLF                ;SEND CR,LF
2191  011262  104005            40600          CHAIN               ;CHAIN TO NEXT TEST
2192  011264  000676            40700          BR      1$          ;REPEAT TEST
```

```
2194                                    40900   ;XXXXXXXXXX
2195                                    41000   ;
2196                                    41100   ;PT10-- PRINTING FREQUENCY TEST-- 120 H'S ARE PRINTED ON 4 LINES
2197                                    41200   ;       30 PER LINE. THE TEST IS SUCH THAT BETWEEN THE FIRST AND SECOND
2198                                    41300   ;       "H" A 30 MSEC DELAY IS INTRODUCED. THIS DELAY IS THEN INCREASED
2199                                    41400   ;       BETWEEN CHARACTERS OUT TO 60 CHARACTERS IN AN EXPONENTIAL
2200                                    41500   ;       MANNER. THE DELAY IS THEN DECREASED IN THE SAME MANNER OUT TO THE
2201                                    41600   ;       120TH CHARACTER. THIS DELAY IS CALCULATED AS FOLLOWS:
2202                                    41700   ;
2203                                    41800   ;               NEW DELAY = OLD DELAY  + OR -  (OLD DELAY/16 + OLD DELAY/128 )
2204                                    41900   ;
2205                                    42000   ;XXXXXXXXXX
2206                                    42100
2207  011266  000010                    42200   PT10:   10                      ;TEST NUMBER
2208  011270  011424                    42300           PT11                    ;NEXT TEST
2209  011272  104016                    42400           PRTHDR                  ;TYPE MESSAGE
2210  011274  012701  000036            42500   1$:     MOV     #36,R1          ;SET R1=30
2211  011300  012702  000170            42600           MOV     #120.,R2        ;SET CHAR COUNT = 120
2212  011304  012737  000036  011322    42700           MOV     #30.,3$+2       ;SET UP DELAY VALUE
2213  011312  012700  000110            42800   2$:     MOV     #110,R0         ;"H" TO R0
2214  011316  104015                    42900           PRINTC                  ;SEND IT
2215  011320  012700  000036            43000   3$:     MOV     #30.,R0
2216  011324  104010                    43100           DELAY                   ;DELAY
2217  011326  005301                    43200           DEC     R1              ;DEC. COUNT OF CHARS PER LINE
2218  011330  001426                    43300           BEQ     6$              ;BRANCH IF 0, END OF LINE
2219  011332  005302                    43400   4$:     DEC     R2              ;DECREMENT CHAR COUNTER
2220  011334  001430                    43500           BEQ     7$              ;BRANCH IF END
2221  011336  013704  011322            43600           MOV     3$+2,R4         ;GET OLD DELAY
2222  011342  006204                    43700           ASR     R4              ;CAL 1/16 OF OLD DELAY
2223  011344  006204                    43800           ASR     R4
2224  011346  006204                    43900           ASR     R4
2225  011350  006204                    44000           ASR     R4
2226  011352  010405                    44100           MOV     R4,R5           ;SAVE 1/16 IN R5
2227  011354  006204                    44200           ASR     R4              ;CAL 1/128 OF OLD DELAY
2228  011356  006204                    44300           ASR     R4
2229  011360  006204                    44400           ASR     R4
2230  011362  060405                    44500           ADD     R4,R5           ;1/16 +1/128 TO R5
2231  011364  022702  000074            44600           CMP     #60.,R2         ;TEST WHICH HALF OF THE 120 CHARS.
2232  011370  003403                    44700           BLE     5$              ;BRANCH IF LT OR EQ 60
2233  011372  160537  011322            44800           SUB     R5,3$+2         ;GT 51, DECREASE DELAY BY 34 MEC.
2234  011376  000745                    44900           BR      2$              ;GO PRINT AGAIN
2235  011400  060537  011322            45000   5$:     ADD     R5,3$+2         ;LT HALF WAY, ADD DELAY OF 34 MEC.
2236  011404  000742                    45100           BR      2$              ;GO PRINT AGAIN
2237  011406  104012                    45200   6$:     CRLF                    ;SEND CRLF
2238  011410  012701  000036            45300           MOV     #36,R1          ;SET R1=30
2239  011414  000746                    45400           BR      4$
2240  011416  104012                    45500   7$:     CRLF                    ;SEND CR,LF
2241  011420  104005                    45600           CHAIN                   ;CHAIN TO NEXT TEST
2242  011422  000724                    45700           BR      1$              ;REPEAT TEST
```

```
2244                                    45900   ;XXXXXXXXXX
2245                                    46000   ;
2246                                    46100   ;PT11-- RIBBON FEED TEST-- THIS TEST PRINTS A SINGLE COLUMN OF X'S
2247                                    46200   ;       (24 LINES) DOWN THE LEFT MARGIN OF THE PAGE.
2248                                    46300   ;       VISUALLY CHECK THE RIBBON FEED MECHANISM FOR PROPER OPERATION.
2249                                    46400   ;
2250                                    46500   ;XXXXXXXXXX
2251                                    46600
2252                                    46700
2253  011424  000011                    46800   PT11:   11                      ;TEST NUMBER
2254  011426  011456                    46900           PT12                    ;NEXT TEST
2255  011430  104016                    47000           PRTHDR                  ;TYPE MESSAGE
2256  011432  012701  000030            47100   1$:     MOV     #30,R1          ;SET R1=24(10), LINE COUNT
2257  011436  012700  000130            47200   2$:     MOV     #130,R0         ;SET CHAR = X
2258  011442  104015                    47300           PRINTC                  ;PRINT X
2259  011444  104012                    47400           CRLF                    ;SEND CR-LF
2260  011446  005301                    47500           DEC     R1              ;DECREMENT LINE COUNT
2261  011450  001372                    47600           BNE     2$              ;CONTINUE IF NOT DONE TEST
2262  011452  104005                    47700           CHAIN                   ;CHAIN TO NEXT TEST
2263  011454  000766                    47800           BR      1$              ;REPEAT TEST
2264                                    47900
2265                                    48000
2266                                    48100
2267                                    48200   ;XXXXXXXXXX
2268                                    48300   ;
2269                                    48400   ;PT12-- PRINTER BELL TEST-- THE LAST TEST IN THE
2270                                    48500   ;       PRINTER TEST SEQUENCE.  THIS TEST OUTPUTS
2271                                    48600   ;       EIGHT BELL SIGNALS TO THE PRINTER
2272                                    48700   ;
2273                                    48800   ;
2274                                    48900   ;XXXXXXXXXX
2275                                    49000   ;
2276  011456  000012                    49100   PT12:   12                      ;THIS TEST
2277  011460  007402                    49200           PTO                     ;NEXT TEST
2278  011462  104016                    49300           PRTHDR                  ;TYPE HEADER
2279  011464  012701  000010            49400   PT12A:  MOV     #10,R1          ;COUNTER TO R1
2280  011470  012700  000007            49500           MOV     #7,R0           ;BELL TO R0
2281  011474  104015                    49600   1$:     PRINTC                  ;SEND IT
2282  011476  005301                    49700           DEC     R1              ;DECREMENT COUNT
2283  011500  001375                    49800           BNE     1$              ;BRANCH IF NOT ZERO
2284  011502  104014                    49900           LF
2285  011504  012700  003720            50000           MOV     #3720,R0        ;DELAY 2 SEC BEFORE RESTARTING
2286  011510  104010                    50100           DELAY
2287  011512  013700  000042            50200           MOV     @#42,R0         ;CHECK IF UNDER ACT11 OR XXDP
2288  011516  001405                    50300           BEQ     HERE            ;CONTINUE TEST SEQUENCE
2289  011520  000240                    50400           NOP                     ;A RESET WAS FORMERLY HERE
2290  011522  004710                    50500   LOGICAL: JSR    PC,(R0)
2291  011524  000240                    50600           NOP
2292  011526  000240                    50700           NOP
2293  011530  000240                    50800           NOP
2294  011532  104005                    50900   HERE:   CHAIN                   ;CHAIN TO NEXT TEST
2295  011534  000753                    51000           BR      PT12A           ;REPEAT TEST
```

```
 2297                                   51200   ;XXXXXXXXXX
 2298                                   51300   ;
 2299                                   51400   ;PT17-- LIFE TEST
 2300                                   51500   ;       THIS TEST PRINTS 2 FULL LINES OF EACH PRINTABLE
 2301                                   51600   ;       CHARACTER AND OVERPRINTS THE SECOND LINE 4 TIMES.
 2302                                   51700   ;       THIS TEST IS CONTINUOUS RUNNING ONCE INITIATED,
 2303                                   51800   ;       LOOPING AUTOMATICALLY ON ITSELF.
 2304                                   51900   ;       END OF PASS COUNT IS CLEARED WHENEVER TEST IS RESTARTED
 2305                                   52000   ;
 2306                                   52100   ;XXXXXXXXXX
 2307                                   52200
 2308  011536  000017                  52300   PT17B:  17                      ;TEST NUMBER
 2309  011540  011536                  52400           PT17B                   ;NEXT TEST
 2310  011542  000137  011576          52500           JMP     PT17D           ;CONTINUE
 2311  011546  000017                  52600   PT17:   17                      ;TEST NUMBER
 2312  011550  011536                  52700           PT17B                   ;NEXT TEST
 2313  011552  005037  012114          52800           CLR     PASCNT          ;CLEAR PASS COUNT
 2314  011556  013704  000652          52900           MOV     WIDTH,R4        ;INITIALIZE R4
 2315  011562  012737  000001  012112  53000           MOV     #1,DIRTN        ;AND DIRECTION OF PRECESS
 2316  011570  104016                  53100           PRTHDR
 2317  011572  104007                  53200           TYPEM                   ;PRINT COLUMN # MESG
 2318  011574  014112                  53300           HDR0
 2319  011576  012703  000041          53400   PT17D:  MOV     #41,R3          ;SET START CHAR
 2320  011602  005237  012114          53500           INC     PASCNT
 2321  011606  023727  012114  000031  53600           CMP     PASCNT,#31      ;DO 31 TIMES
 2322  011614  001003                  53700           BNE     20$             ;BRANCH IF NOT DONE
 2323  011616  012737  000001  012114  53800           MOV     #1,PASCNT       ;START OVER
 2324  011624  012700  014025          53900   20$:    MOV     #PASMES,R0      ;SET MESG ADDR
 2325  011630  013701  012114          54000           MOV     PASCNT,R1       ;# TO CONVERT
 2326  011634  012702  000002          54100           MOV     #2,R2           ;# DIGITS
 2327  011640  104023                  54200           BTOASC                  ;CONVERT PASCNT TO ASCII
 2328  011642  013701  000652          54300   1$:     MOV     WIDTH,R1        ;SET COLUMN COUNT
 2329  011646  010300                  54400   2$:     MOV     R3,R0           ;GET CHARACTER
 2330  011650  004737  011764          54500           JSR     PC,CKPOS        ;TIME TO INSERT PASS # ?
 2331  011654  104015                  54600           PRINTC                  ;SEND CHAR
 2332  011656  005301                  54700           DEC     R1              ;DECREMENT COUNT
 2333  011660  003372                  54800           BGT     2$              ;BRANCH IF NOT DONE
 2334  011662  004737  012032          54900           JSR     PC,ADJR4        ;ADJUST R4 POINTER
 2335  011666  104012                  55000           CRLF
 2336  011670  012702  000005          55100           MOV     #5,R2           ;SET OVERPRINT COUNT
 2337  011674  013701  000652          55200   3$:     MOV     WIDTH,R1        ;SET COLUMN COUNT
 2338  011700  010300                  55300   4$:     MOV     R3,R0           ;GET CHARACTER
 2339  011702  004737  011764          55400           JSR     PC,CKPOS        ;TIME TO INSERT PASS # ?
 2340  011706  104015                  55500           PRINTC                  ;SEND CHAR
 2341  011710  005301                  55600           DEC     R1              ;DECREMENT COUNT
 2342  011712  003372                  55700           BGT     4$              ;BRANCH IF NOT DONE
 2343  011714  104022                  55800           CR                      ;SEND CR
 2344  011716  005302                  55900           DEC     R2              ;DONE OVERPRINTS ?
 2345  011720  001365                  56000           BNE     3$              ;NO. CONTINUE
 2346  011722  004737  012032          56100           JSR     PC,ADJR4        ;ADJUST R4 POINTER
 2347  011726  104014                  56200           LF                      ;SEND LF
 2348  011730  005203                  56300           INC     R3              ;SET NEXT CHAR
 2349  011732  022703  000177          56400           CMP     #177,R3         ;DONE CHAR SET ?
 2350  011736  001341                  56500           BNE     1$              ;NO. CONTINUE
 2351  011740  004737  012032          56600           JSR     PC,ADJR4        ;OFFSET POINTER 3 PLACES
 2352  011744  004737  012032          56700           JSR     PC,ADJR4        ;TO RETAIN VISUAL ALIGNMENT
```

```
2353  011750  004737  012032          56800           JSR     PC,ADJR4        ;THROUGH END OF PASS
2354  011754  104007                  56900           TYPEM                   ;TYPE END OF PASS MESG
2355  011756  014006                  57000           ENDPAS
2356  011760  104005                  57100           CHAIN                   ;REPEAT TEST
2357  011762  000705                  57200           BR      PT17D
2358                                  57300
```

```
2360                                    57500
2361  011764  020401                    57600  CKPOS:  CMP     R4,R1           ;IS IT TIME TO INSERT PASS # ?
2362  011766  001020                    57700          BNE     1$              ;BRANCH IF NO
2363  011770  012700  000040            57800          MOV     #40,R0          ;PRINT A SPACE
2364  011774  104015                    57900          PRINTC
2365  011776  113700  014025            58000          MOVB    PASMES,R0       ;PRINT MSG OF PASS COUNT
2366  012002  104015                    58100          PRINTC
2367  012004  113700  014026            58200          MOVB    PASMES+1,R0
2368  012010  104015                    58300          PRINTC
2369  012012  012700  000040            58400          MOV     #40,R0          ;PRINT A SPACE
2370  012016  104015                    58500          PRINTC
2371  012020  162701  000003            58600          SUB     #3,R1           ;ADJUST R1 3 POSITIONS
2372  012024  062716  000002            58700          ADD     #2,(SP)         ;ADJUST RETURN PC OVER PRINTC
2373  012030  000207                    58800  1$:     RTS     PC
2374                                    58900
2375  012032  005737  012112            59000  ADJR4:  TST     DIRTN           ;TEST DIRECTION OF PRECESS
2376  012036  001013                    59100          BNE     1$              ;BR IF LEFT
2377  012040  005204                    59200          INC     R4              ;INCREASE POSITION CNTR
2378  012042  020437  000652            59300          CMP     R4,WIDTH        ;IS R4 > WIDTH ?
2379  012046  101420                    59400          BLOS    3$              ;BR IF NOT GREATER
2380  012050  013704  000652            59500          MOV     WIDTH,R4        ;CHANGE DIRECTION
2381  012054  005304                    59600          DEC     R4              ;       TO
2382  012056  012737  000001  012112    59700          MOV     #1,DIRTN        ;      LEFT.
2383  012064  000411                    59800          BR      3$
2384  012066  005304                    59900  1$:     DEC     R4              ;DECREASE POSITION CNTR
2385  012070  020427  000004            60000          CMP     R4,#4   ;LESS THAN 4 ?
2386  012074  002401                    60100          BLT     2$              ;BR IF YES
2387  012076  000404                    60200          BR      3$              ;ELSE EXIT
2388  012100  012704  000005            60300  2$:     MOV     #5,R4           ;SET R4 TO POS 5
2389  012104  005037  012112            60400          CLR     DIRTN           ;CHANGE DIRECTION TO RIGHT
2390  012110  000207                    60500  3$:     RTS     PC              ;EXIT
2391                                    60600
2392  012112  000000                    60700  DIRTN:  WORD    0               ;DIRECTION OF PRECESS (0=LEFT)
2393                                    60800
2394  012114  000000                    60900  PASCNT: WORD    0
```

2396

```
2397                                    00100          .SBTTL  LA36 ECHO TESTS
2398                                    00200  ;
2399                                    00300  ;XXXXXXXXXX
2400                                    00400  ;
2401                                    00500  ;E020-- CHARACTER ECHO TEST-- ALL PRINTABLE AND
2402                                    00600  ;        NON-PRINTING CHARACTERS TYPED ON THE KEYBOARD
2403                                    00700  ;        ARE USED TO DRIVE THE PRINTER, ONE CHARACTER AT
2404                                    00800  ;        A TIME.  A "RUBOUT" WILL CAUSE THE TEST TO BE
2405                                    00900  ;        TERMINATED.
2406                                    01000  ;
2407                                    01100  ;XXXXXXXXXX
2408                                    01200  ;
2409  012116  000020                    01300  E020:    20                      ;TEST NUMBER
2410  012120  012166                    01400          E021                    ;NEXT TEST
2411  012122  104016                    01500          PRTHDR                  ;TYPE HEADER
2412  012124  104020                    01600  1$:     READ                    ;GO WAIT FOR KEYBOARD INPUT
2413  012126  012700  000036            01700          MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
2414  012132  104010                    01800          DELAY
2415  012134  022737  000177  000700    01900          CMP     #177,TEMPCH     ;CHECK IF RUBOUT
2416  012142  001405                    02000          BEQ     2$              ;BRANCH IF YES
2417  012144  104017                    02100          PRNT                    ;NO, CHECK PRINTER READY
2418  012146  117777  166442  166444    02200          MOVB    @TKB,@TPB       ;READY, ECHO CHARACTER
2419  012154  000763                    02300          BR      1$
2420  012156  104007                    02400  2$:     TYPEM                   ;PRINT TERMINATION MESSAGE
2421  012160  014255                    02500          ECOEND
2422  012162  104005                    02600          CHAIN                   ;CHAIN TO NEXT TEST
2423  012164  000757                    02700          BR      1$              ;REPEAT TEST
2424                                    02800  ;
2425                                    02900  ;XXXXXXXXXX
2426                                    03000  ;
2427                                    03100  ;E021-- LINE ECHO TEST, FAST RATE-- THIS TEST WILL
2428                                    03200  ;        CAUSE THE CONTINUAL PRINTING OF "0" AT THE MAXIMUM
2429                                    03300  ;        RATE UNTIL EITHER ANOTHER CHARACTER IS SELECTED
2430                                    03400  ;        BY PRESSING A KEY ON THE KEYBOARD OR TERMINATION BY THE
2431                                    03500  ;        RUBOUT.
2432                                    03600  ;
2433                                    03700  ;XXXXXXXXXX
2434                                    03800  ;
2435  012166  000021                    03900  E021:    21                      ;TEST NUMBER
2436  012170  012224                    04000          E022                    ;NEXT TEST
2437  012172  104016                    04100          PRTHDR                  ;TYPE HEADER
2438  012174  012737  000060  000662    04200  E021A:  MOV     #60,REPT        ;CHARACTER TO BE REPEATED (0)
2439  012202  013702  000652            04300  1$:     MOV     WIDTH,R2        ;SET COLUMN COUNT
2440  012206  013700  000662            04400  2$:     MOV     REPT,R0         ;GET CHAR
2441  012212  104015                    04500          PRINTC                  ;PRINT CHAR
2442  012214  005302                    04600          DEC     R2              ;DEC COLUMN COUNT
2443  012216  003373                    04700          BGT     2$              ;FINISH LINE
2444  012220  104012                    04800          CRLF                    ;SEND A CR AND LF
2445  012222  000767                    04900          BR      1$
```

```
2447                                        05100   ;XXXXXXXXXX
2448                                        05200   ;
2449                                        05300   ;E022-- LINE ECHO TEST, SLOW RATE-- SAME AS E021 EXCEPT
2450                                        05400   ;       THAT A DELAY IS INTRODUCED BETWEEN CHARACTERS
2451                                        05500   ;       TO PRODUCE A LCV ACTION
2452                                        05600   ;
2453                                        05700   ;XXXXXXXXXX
2454                                        05800   ;
2455  012224  000022                        05900   E022:   22
2456  012226  012476                        06000           E023
2457  012230  104016                        06100           PRTHDR                  ;TYPE HEADER
2458  012232  012737  000060  000662        06200   E022A:  MOV     #60,REPT        ;LOAD 0 AS INITIAL CHARACTER
2459  012240  013702  000652                06300   1$:     MOV     WIDTH,R2        ;SET COLUMN COUNT
2460  012244  013700  000662                06400   2$:     MOV     REPT,R0         ;GET CHAR
2461  012250  104015                        06500           PRINTC                  ;PRINT CHAR
2462  012252  005302                        06600           DEC     R2              ;DEC COLUMN COUNT
2463  012254  001404                        06700           BEQ     3$              ;BRANCH IF DONE LINE
2464  012256  012700  003410                06800           MOV     #3410,R0
2465  012262  104010                        06900           DELAY                   ;DELAY 1.8 SEC.
2466  012264  000767                        07000           BR      2$              ;OUTPUT NEW CHAR.
2467  012266  104012                        07100   3$:     CRLF                    ;SEND A CR AND LF
2468  012270  000763                        07200           BR      1$
```

```
                                        07400   ;**********
                                        07500   ;
                                        07600   ;       THIS FOLLOWING TABLE IS USED BY TEST E023
                                        07700   ;
                                        07800   ;**********
                                        07900
2476  012272  052516  020114            08000   MONIC:  .ASCII  /NUL /
2477  012276  047523  020110            08100           .ASCII  /SOH /
2478  012302  052123  020130            08200           .ASCII  /STX /
2479  012306  052105  020130            08300           .ASCII  /ETX /
2480  012312  047505  020124            08400           .ASCII  /EOT /
2481  012316  047105  020121            08500           .ASCII  /ENQ /
2482  012322  041501  020113            08600           .ASCII  /ACK /
2483  012326  042502  020114            08700           .ASCII  /BEL /
2484  012332  051502  020040            08800           .ASCII  /BS  /
2485  012336  052110  020040            08900           .ASCII  /HT  /
2486  012342  043114  020040            09000           .ASCII  /LF  /
2487  012346  052126  020040            09100           .ASCII  /VT  /
2488  012352  043106  020040            09200           .ASCII  /FF  /
2489  012356  051103  020040            09300           .ASCII  /CR  /
2490  012362  047523  020040            09400           .ASCII  /SO  /
2491  012366  044523  020040            09500           .ASCII  /SI  /
2492  012372  046104  020105            09600           .ASCII  /DLE /
2493  012376  041504  020061            09700           .ASCII  /DC1 /
2494  012402  041504  020062            09800           .ASCII  /DC2 /
2495  012406  041504  020063            09900           .ASCII  /DC3 /
2496  012412  041504  020064            10000           .ASCII  /DC4 /
2497  012416  040516  020113            10100           .ASCII  /NAK /
2498  012422  054523  020116            10200           .ASCII  /SYN /
2499  012426  052105  020102            10300           .ASCII  /ETB /
2500  012432  040503  020116            10400           .ASCII  /CAN /
2501  012436  046505  020040            10500           .ASCII  /EM  /
2502  012442  052523  020102            10600           .ASCII  /SUB /
2503  012446  051505  020103            10700           .ASCII  /ESC /
2504  012452  051506  020040            10800           .ASCII  /FS  /
2505  012456  051507  020040            10900           .ASCII  /GS  /
2506  012462  051522  020040            11000           .ASCII  /RS  /
2507  012466  051525  020040            11100           .ASCII  /US  /
2508  012472  050123  020040            11200           .ASCII  /SP  /
2509                                    11300
2510                                    11400           .EVEN
```

```
2512                                    11600   ;XXXXXXXXXX
2513                                    11700   ;
2514                                    11800   ;E023-- CHARACTER CODE TEST-- ANY CHARACTER SELECTED
2515                                    11900   ;       WILL BE ECHOED ALONG WITH ITS OCTAL CODE.
2516                                    12000   ;       A MNEMONIC WILL BE PRINTED INSTEAD OF THE CHARACTER
2517                                    12100   ;       IF IT IS A NON-PRINTING CHARACTER.
2518                                    12200   ;       THE PARITY OF THE RECEIVED CODE WILL ALSO BE
2519                                    12300   ;       INDICATED AS EITHER EVEN OR ODD.
2520                                    12400   ;
2521                                    12500   ;XXXXXXXXXX
2522                                    12600   ;
2523  012476  000023                    12700   E023:   23                      ;TEST NUMBER
2524  012500  013020                    12800           E024                    ;NEXT TEST
2525  012502  104016                    12900           PRTHDR                  ;TYPE HEADER
2526  012504  104020                    13000   1$:     READ                    ;GO WAIT FOR CHARACTER
2527  012506  012700  000036            13100           MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
2528  012512  104010                    13200           DELAY
2529  012514  023727  000700  000041    13300           CMP     TEMPCH,#41      ;TEST IF CHAR IS PRINTABLE
2530  012522  103015                    13400           BHIS    3$              ;BRANCH IF IT IS
2531  012524  004737  012660            13500           JSR     PC,STRLN        ;STORE CODE INTO MESSAGE
2532  012530  113700  000700            13600           MOVB    TEMPCH,R0       ;GET CODE AGAIN
2533  012534  006300                    13700           ASL     R0              ;MULT BY 2
2534  012536  006300                    13800           ASL     R0              ;MULT BY 4
2535  012540  062700  012272            13900           ADD     #MONIC,R0       ;ADD ADDR OF MNEMONIC TABLE
2536  012544  004737  012736            14000           JSR     PC,MOVNUM       ;MOV MNEMONIC TO MESSAGE
2537  012550  104000                    14100   2$:     TYPE                    ;TYPE CODE AND MNEMONIC
2538  012552  014304                    14200           E023M                   ;ADDRESS OF MESSAGE
2539  012554  000753                    14300           BR      1$              ;GO WAIT FOR NEXT CHARACTER
2540  012556  023727  000700  000177    14400   3$:     CMP     TEMPCH,#177     ;TEST IF CHAR IS A RUBOUT
2541  012564  001421                    14500           BEQ     4$              ;BRANCH IF RUBOUT
2542  012566  012701  013010            14600           MOV     #MG24,R1
2543  012572  113721  000700            14700           MOVB    TEMPCH,(R1)+
2544  012576  112721  000040            14800           MOVB    #40,(R1)+
2545  012602  112721  000040            14900           MOVB    #40,(R1)+
2546  012606  112721  000040            15000           MOVB    #40,(R1)+
2547  012612  004737  012660            15100           JSR     PC,STRLN        ;STORE CODE INTO MESSAGE
2548  012616  012700  013010            15200           MOV     #MG24,R0        ;ADDR OF CHAR INTO R0
2549  012622  004737  012736            15300           JSR     PC,MOVNUM       ;MOVE CHAR INTO MESSAGE
2550  012626  000750                    15400           BR      2$              ;TYPE MESSAGE
2551  012630  004737  012660            15500   4$:     JSR     PC,STRLN        ;RUBOUT, CONVERT AND STOR CODE
2552  012634  012700  013014            15600           MOV     #MG25,R0        ;ADDR. OF DEL INTO R0
2553  012640  004737  012736            15700           JSR     PC,MOVNUM       ;MOVE DEL INTO MESSAGE
2554  012644  104000                    15800           TYPE                    ;TYPE MESSAGE
2555  012646  014304                    15900           E023M                   ;ADDR OF MESSAGE
2556  012650  104007                    16000           TYPEM
2557  012652  014255                    16100           ECOEND
2558  012654  104005                    16200           CHAIN                   ;CHAIN TO NEXT TEST
2559  012656  000712                    16300           BR      1$              ;REPEAT TEST
2560  012660  012702  000003            16400   STRLN:  MOV     #3,R2           ;COUNT OF 3 TO R2
2561  012664  012701  014306            16500           MOV     #LINE5,R1       ;ADDR OF MESG TO R1
2562  012670  062701  000003            16600           ADD     #3,R1           ;POINT TO LAST SPACE IN MESG
2563  012674  013700  000704            16700   1$:     MOV     PCHAR,R0        ;MOVE OCTAL CODE TO R0
2564  012700  042700  177770            16800           BIC     #177770,R0      ;SAVE LS OCTAL CHAR.
2565  012704  062700  000060            16900           ADD     #60,R0          ;MAKE ASCII
2566  012710  110041                    17000           MOVB    R0,-(R1)        ;MOVE INTO MESG
2567  012712  005302                    17100           DEC     R2              ;DECREMENT CHAR COUNTER
```

```
2568  012714  001407                17200          BEQ     2$              ;BRANCH IF 3 MOVED
2569  012716  006237  000704        17300          ASR     PCHAR           ;NOT THREE, SHIFT NEXT OCTAL
2570  012722  006237  000704        17400          ASR     PCHAR           ;CHARACTER TO THE RIGHT
2571  012726  006237  000704        17500          ASR     PCHAR           ;
2572  012732  000760                17600          BR      1$              ;CONVERT AND STORE NEXT CHAR.
2573  012734  000207                17700  2$:     RTS     PC              ;RETURN TO CALLER
2574  012736  012701  014312        17800  MOVNUM: MOV     #LINE5A,R1      ;ADDR OF LINES IN R1
2575  012742  012702  000004        17900          MOV     #4,R2           ;COUNT OF 4 TO R2
2576  012746  112021                18000  1$:     MOVB    (R0)+,(R1)+     ;MOV 4 CHARS TO MESG AREA
2577  012750  005302                18100          DEC     R2              ;DECREMENT COUNTER
2578  012752  001375                18200          BNE     1$              ;BRANCH IF NOT ALL DONE
2579  012754  105737  000702        18300          TSTB    PARITY          ;TEST PARITY FLAG
2580  012760  001003                18400          BNE     2$              ;BRANCH IF ODD PARITY
2581  012762  012700  014354        18500          MOV     #EVEN,R0        ;SET ADDRESS FOR EVEN PARITY MESG
2582  012766  000402                18600          BR      3$              ;CONTINUE
2583  012770  012700  014360        18700  2$:     MOV     #ODD,R0         ;SET ADDRESS FOR ODD PARITY MESG
2584  012774  012702  000004        18800  3$:     MOV     #4,R2           ;COUNT OF 4 TO R2
2585  013000  112021                18900  4$:     MOVB    (R0)+,(R1)+     ;MOVE 4 CHARS TO MESG AREA
2586  013002  005302                19000          DEC     R2              ;DECREMENT COUNTER
2587  013004  001375                19100          BNE     4$              ;BRANCH IF NOT DONE
2588  013006  000207                19200          RTS     PC              ;RETURN
2589                                19300
2590  013010  020040  020040        19400  MG24:   .ASCII  /    /          ;SAVE CHARACTER CODE
2591                                19500
2592                                19600          .EVEN
2593                                19700
2594  013014  042504  020114        19800  MG25:   .ASCII  /DEL /          ;MNEMONIC FOR RUBOUT
2595                                19900
2596                                20000          .EVEN
```

```
2598                                    20200   ;XXXXXXXXXX
2599                                    20300   ;E024-- SELECTED PATTERN ECHO TEST-- SELECT 1 TO 256
2600                                    20400   ;       CHARACTERS.  EACH WILL BE ECHOED
2601                                    20500   ;       AND STORED UNTIL THE CNTL/C IS SELECTED.
2602                                    20600   ;       AT THAT TIME ALL CHARACTERS WILL BE PRINTED AS
2603                                    20700   ;       A CONTINOUS STRING UNTIL EITHER THE RUBOUT IS
2604                                    20800   ;       SELECTED TO TERMINATE OR THE CNTL/C IS SELECTED
2605                                    20900   ;       AGAIN.  A TERMINATEING CNTL/C FOLLOWED BY ANOTHER
2606                                    21000   ;       CNTL/C WILL ALWAYS CAUSE THE LAST INPUTTED STRING TO
2607                                    21100   ;       BE PRINTED.  A TERMINATING CNTL/C FOLLOWED BY A CHARACTER OTHER THAN A
2608                                    21200   ;       RUBOUT WILL CAUSE A NEW STRING TO BE INPUTTED.
2609                                    21300   ;XXXXXXXXXX
2610                                    21400
2611  013020  000024                    21500   E024:   24                      ;TEST NUMBER
2612  013022  013566                    21600           E025                    ;NEXT TEST
2613  013024  104016                    21700           PRTHDR                  ;TYPE TEST HEADER
2614  013026  005001                    21800   E024B:  CLR     R1              ;CLEAR CHARACTER COUNT
2615  013030  012702  013164            21900           MOV     #BUFR,R2        ;ADDRESS OF BUFFER TO R2
2616  013034  104020                    22000   1$:     READ                    ;WAIT FOR INPUT
2617  013036  012700  000036            22100           MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
2618  013042  104010                    22200           DELAY
2619  013044  022737  000177  000700    22300           CMP     #177,TEMPCH     ;TEST IF RUBOUT
2620  013052  001440                    22400           BEQ     TERM            ;BRANCH IF RUBOUT
2621  013054  022737  000003  000700    22500           CMP     #3,TEMPCH       ;TEST IF CNTL-C
2622  013062  001413                    22600           BEQ     OUTPUT          ;BRANCH IF CNTL-C
2623  013064  020127  000400            22700           CMP     R1,#256.        ;YES, CHECK IF CHAR CNT IS EQ, GT 256
2624  013070  103361                    22800           BHIS    1$              ;BRANCH IF YES, IGNORE CHAR
2625  013072  113722  000700            22900           MOVB    TEMPCH,(R2)+    ;STORE CHAR INTO BUFFER
2626  013076  005201                    23000           INC     R1              ;INCREMENT CHARACTER COUNT
2627  013100  104017                    23100           PRNT                    ;CHECK IF PRINTER READY
2628  013102  113777  000700  165510    23200           MOVB    TEMPCH,@TPB     ;ECHO CHAR
2629  013110  000751                    23300           BR      1$              ;GO WAIT FOR NEXT CHAR
2630                                    23400   ;
2631                                    23500   ;SECTION TO OUTPUT CONTINOUS STRING
2632                                    23600   ;
2633  013112  020227  013164            23700   OUTPUT: CMP     R2,#BUFR        ;CHECK IF POINTER IS AT START OF TABLE
2634  013116  001403                    23800           BEQ     1$              ;YES, BRANCH
2635  013120  113722  000700            23900           MOVB    TEMPCH,(R2)+    ;NO, STORE  C IN TABLE
2636  013124  104013                    24000           SCRLF                   ;SEND A CR LF
2637  013126  012702  013164            24100   1$:     MOV     #BUFR,R2        ;BUFFER ADDRESS TO R2
2638  013132  021227  000003            24200           CMP     (R2),#3         ;CHECK IF FIRST CHAR IS  C
2639  013136  001733                    24300           BEQ     E024B           ;YES, LOOK FOR INPUT AGAIN
2640  013140  112200                    24400   2$:     MOVB    (R2)+,R0        ;GET CHARACTER
2641  013142  020027  000003            24500           CMP     R0,#3           ;DONE STRING?
2642  013146  001767                    24600           BEQ     1$              ;YES, RESTART STRING
2643  013150  104015                    24700           PRINTC                  ;PRINT CHAR
2644  013152  000772                    24800           BR      2$              ;CONTINUE
2645  013154  104007                    24900   TERM:   TYPEM                   ;OUTPUT TERMINATION MESSAGE
2646  013156  014255                    25000           ECOEND
2647  013160  104005                    25100           CHAIN                   ;CHAIN TO NEXT TEST
2648  013162  000721                    25200           BR      E024B           ;REPEAT TEST
2649  013164  000003                    25300   BUFR:   3                       ;INITIALIZE FIRST CHAR AS CNTL-C IN TABLE
2650  013166  000400                    25400           .BLKB   256.            ;256 CHARACTER BUFFER
```

```
2652                                    25600   ;XXXXXXXXXXX
2653                                    25700   ;
2654                                    25800   ;E025-- BELL ECHO TEST-- A MESSAGE IS PRINTED AND
2655                                    25900   ;       THE TEST WAITS FOR SOME PRINTABLE CHARACTER
2656                                    26000   ;       TO BE SELECTED ON THE KEYBOARD (GT040).  THIS
2657                                    26100   ;       TEST IS VALID ONLY IF THE PAPER WIDTH IS GT 64
2658                                    26200   ;       COLUMNS.  IF LT64 COLUMNS AN ILLEGAL BELL TEST
2659                                    26300   ;       MESSAGE IS PRINTED.
2660                                    26400   ;
2661                                    26500   ;XXXXXXXXXXX
2662                                    26600   ;
2663  013566  000025                    26700   E025:   25                      ;TEST NUMBER
2664  013570  012116                    26800           E020                    ;NEXT TEST HEADER
2665  013572  104016                    26900           PRTHDR                  ;PRINT HEADER
2666  013574  023727  000652  000101    27000   1$:     CMP     WIDTH,#101      ;TEST IF COLUMN COUNTIS EQ,GT 64
2667  013602  103427                    27100           BLO     4$              ;BRANCH IF NOT
2668  013604  104007                    27200           TYPEM                   ;TYPE TEST MESG
2669  013606  014130                    27300           E025MA
2670  013610  000402                    27400           BR      3$              ;WAIT FOR CHAR
2671  013612  104000                    27500   2$:     TYPE                    ;TYPE TEST MESG ON TERM CHAR RCVD ON
2672  013614  014130                    27600           E025MA
2673  013616  104020                    27700   3$:     READ                    ;WAIT FOR OPERATOR RESPONSE
2674  013620  012700  000036            27800           MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
2675  013624  104010                    27900           DELAY
2676  013626  023727  000700  000040    28000           CMP     TEMPCH,#40      ;TEST IF PRINTABLE
2677  013634  103770                    28100           BLO     3$              ;BRANCH IF NON-PRINTABLE
2678  013636  022737  000177  000700    28200           CMP     #177,TEMPCH     ;CHECK IF CHAR IS RUBOUT
2679  013644  001410                    28300           BEQ     5$              ;BRANCH IF YES
2680  013646  104017                    28400           PRNT                    ;CHECK IF PRINTER IS READY
2681  013650  113777  000700  164742    28500           MOVB    TEMPCH,@TPB     ;PRINT CHAR.  (BELL SHOULD SOUND)
2682  013656  104013                    28600           SCRLF                   ;SEND A CRLF
2683  013660  000754                    28700           BR      2$              ;REPEAT
2684  013662  104007                    28800   4$:     TYPEM                   ;TYPE ERROR MESSAGE
2685  013664  014230                    28900           E025MB
2686  013666  104007                    29000   5$:     TYPEM                   ;PRINT TERMINATION
2687  013670  014255                    29100           ECOEND
2688  013672  104005                    29200           CHAIN                   ;EXIT TO NEXT TEST
2689  013674  000737                    29300           BR      1$              ;REPEAT TEST
```

```
2691
```

```
                                                00100           .SBTTL  MISC. DIAGNOSTIC MESSAGES
                                                00200   ;
2694  013676  001007  007600  040515  00300    STARTM: .ASCII  <7><2><ACRLF><17>/MAINDEC-11-DZLAC-C/<ACRLF>
      013704  047111  042504  026503
      013712  030461  042055  046132
      013720  041501  041455     200
2695  013725     114  031501  020066  00400            .ASCII  /LA36 TERMINAL DIAGNOSTIC/<ACRLF>
      013732  042524  046522  047111
      013740  046101  042040  040511
      013746  047107  051517  044524
      013754  100103
2696  013756  046104  030461  023040  00500            .ASCIZ  /DL11 & KL11 INTERFACE/<ACRLF><12>
      013764  045440  030514  020061
      013772  047111  042524  043122
      014000  041501  100105  000012
2697  014006  005200  047105  020104  00600    ENDPAS: .ASCII  <ACRLF><12>/END OF PASS  /
      014014  043117  050040  051501
      014022  020123     040
2698  014025     060  030060  100060  00700    PASMES: .ASCIZ  /0000/<ACRLF><12>
      014032  000012
2699  014034  041600  047117  047523  00800    DL11S:  .ASCII  <ACRLF>/CONSOLE & /
      014042  042514  023040     040
2700  014047     060  020060  046104  00900    DL11S1: .ASCIZ  /00 DL11'S UNDER TEST/<ACRLF><12>
      014054  030461  051447  052440
      014062  042116  051105  052040
      014070  051505  100124  000012
2701  014076  001007  007600  052012  01000    HDRMSG: .ASCIZ  <7><2><ACRLF><17><12>/TEST #/
      014104  051505  020124  000043
2702  014112  030060  020060  047503  01100    HDR0:   .ASCIZ  /000 COLUMNS/<ACRLF><12>
      014120  052514  047115  100123
      014126  000012
2703  014130  054524  042520  040440  01200    E025MA: .ASCII  /TYPE ANY PRINTABLE CHARACTER /
      014136  054516  050040  044522
      014144  052116  041101  042514
      014152  041440  040510  040522
      014160  052103  051105     040
2704  014165     101  042116  046040  01300            .ASCIZ  /AND LISTEN FOR BELL ...../
      014172  051511  042524  020116
      014200  047506  020122  042502
      014206  046114  027056  027056
      014214  027056  027056  027056
      014222  027056  027056  000056
2705  014230  047200  052117  042440  01400    E025MB: .ASCIZ  <ACRLF>/NOT ENOUGH COLUMNS/<ACRLF>
      014236  047516  043525  020110
      014244  047503  052514  047115
      014252  100123     000
2706  014255     200  041505  047510  01500    ECOEND: .ASCIZ  <ACRLF>/ECHO TEST TERMINATED/<ACRLF>
      014262  052040  051505  020124
      014270  042524  046522  047111
      014276  052101  042105  000200
2707  014304  020040                   01600    E023M:  .ASCII  /  /
2708  014306  020040  020040           01700    LINE5:  .ASCII  /    /          ;MESG FOR TEST E024
2709  014312  020040  020040  020040   01800    LINE5A: .ASCIZ  /        /<ACRLF>
      014320  020040  000200
2710  014324  007600  051412  046105  01900    MESG3:  .ASCIZ  <ACRLF><17><12>/SELECT TEST NUMBER  /
      014332  041505  020124  042524
```

```
                014340  052123  047040  046525
                014346  042502  020122  000040
2711            014354  053105  047105          02000   EVEN:   .ASCII  /EVEN/
2712            014360  042117  020104          02100   ODD:    .ASCII  /ODD /
2713            014364  054524  042520  040440 02200   OPMSG.  .ASCIZ  /TYPE ANY CHARACTER/
                014372  054516  041440  040510
                014400  040522  052103  051105
                014406     000
2714            014407     125  042523  051440 02300   NOSWR:  .ASCIZ  /USE SOFTWARE SWITCH REG AT MEMORY ADDR 176/<7>
                014414  043117  053524  051101
                014422  020105  053523  052111
                014430  044103  051040  043505
                014436  040440  020124  042515
                014444  047515  054522  040440
                014452  042104  020122  033461
                014460  003466     000
2715                                    02400
2716                                    02500
2717            000001                  02600           .END
```

```
ACRLF = 000200
ADJR4   012032
ADTENP  004070
AREAD = 104021
AT0     005276
AT0X    005300
AT1     005330
AT10    005754
AT11    006012
AT12    006052
AT13    006126
AT14    006206
AT15    006272
AT16    006372
AT17    006440
AT2     005362
AT20    006510
AT21    006602
AT22    006702
AT23    007010
AT24    007122
AT25    007222
AT26    007300
AT3     005414
AT4     005446
AT5     005536
AT6     005614
AT7     005704
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BRCTR   000664
BTOASC= 104023
BUFR    013164
CHAIN = 104005
CHAINN  001412
CHAINY  001534
CHALT = 104006
CHLT    000720
CKPOS   011764
CNTLSW  000634
CNTR    000716
CNVCTR  004062
CONADD  000602
CONIT   003700
CONSET  003704
CONVEC  000604
COUNT3  000666
CR    = 104022
CRBUF   000646
CRLF  = 104012
CTRA    000650
CURTST  000676
DELAY = 104010
DIGIT   004064
DIRTN   012112
DISPRE  000174
DLADR   000606
DLCNT   000656
DLNR    000610
DLY     003426
DL11S   014034
DL11S1  014047
ECOEND  014255
EHALT = 104002
EHLT    003336
EMTINT  002722
EMTTAB  002776
ENDPAS  014006
END2    001332
END2A   001320
END3    001262
END4    001334
E020    012116
E021    012166
E021A   012174
E022    012224
E022A   012232
E023    012476
E023M   014304
E024    013020
E024B   013026
E025    013566
E025MA  014130
E025MB  014230
ERR     003310
ERRHLT  003344
ERROR = 104001
EVEN    014354
FORWD = 104024
FORWDA  003604
FORWDB  003612
FSTDL   000632
HDRMSG  014076
HDRO    014112
HERE    011532
ICTR    000660
IDEZ    007706
INCHK   000710
LEVEL   000654
LF    = 104014
LFCNT   000706
LINE3   010444
LINE5   014306
LINE5A  014312
LOGICA  011522
LSI11 = 001000
MACHER  000004
MESG3   014324
MG24    013010
MG25    013014
MONIC   012272
MOVNUM  012736
NEXT    001654
NEXT1   001674
NITRSW= 004000
NOSWR   014407
NPCODE  010137
NXTST   000640
ODD     014360
OPEN  = 000000
OPMSG   014364
OUTPUT  013112
PARITY  000702
PASCNT  012114
PASMES  014025
PCHAR   000704
PFAIL   003450
POPSP = 005726
POPSP2= 022626
PRGID   000644
PRGTAB  002522
PRINTC= 104015
PRNT  = 104017
PRTHDR= 104016
PRTY4 = 000200
PRTY7 = 000340
PSW   = 177776
PT0     007402
PT1     007456
PT10    011266
PT11    011424
PT12    011456
PT12A   011464
PT17    011546
PT17B   011536
PT17D   011576
PT2     007600
PT3     010164
PT4     010304
PT5     010462
PT5AL   010630
PT6     010666
PT7     011054
READ  = 104020
READC = 104025
READ1   004202
REF     010414
REPT    000662
RESTRT  003506
RTNNO   000636
SAVR6   003504
SCOPSW= 040000
SCOPTR  000642
SCRLF = 104013
SELHLT  000734
SKIP    001630
SPARET  003072
SPBOT   000600
SPC     007570
SPCNT   000674
SPSP    007550
SP2     007562
SR      000714
START   001010
STARTM  013676
STARTX  001024
START1  000736
START2  000754
START3  000772
STLSPV  003376
STLSRV  003346
STPCHV= 104004
STPPA   003414
STPRA   003364
STRDRV= 104003
STRLN   012660
SWREG   000176
TEMP    000712
TEMPCH  000700
TENPWR  004066
TERM    013154
TESTC   002460
TIMER   000672
TKB     000614
TKLVL   000624
TKS     000612
TKVTR   000622
TPB     000620
TPBS    003774
TPLVL   000630
TPS     000616
TPSS    003772
TPVTR   000626
TTYCTL= 104011
TTY1    001776
TTY1B   002062
TYP     003076
TYPE  = 104000
TYPEM = 104007
TYPM    003154
WAITF   001700
WIDTH   000652
XCSR    000670
$AREAD  003630
$BTASC  003776
$CR     003216
$CRLF   003204
$FORWD  003552
$LF     003206
$PRHDR  003226
$PRNT   004304
$PRTC   004314
$READ   004102
$READC  004174
$SCRLF  003132
        = 014463

. ABS.   014463     000

ERRORS DETECTED:  0

DZLACD,DZLACD/DOC=DZLACD/DIAG
```

RUN-TIME: 2 6 .2 SECONDS
RUN-TIME RATIO: 249/9=27.2
CORE USED: 5K (9 PAGES)

DOCUMENT PAGES: 75