```
1                       .REM_
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16                                      IDENTIFICATION
17
18
19
20                      PRODUCT CODE:           AC-8898F-MC
21
22                      PRODUCT NAME:           CZLACFO LA36 TERM (DL11 & KL11)
23
24                      DATE:                   AUGUST 1978
25
26                      MAINTAINER:             DIAGNOSTIC GROUP
27
28
29
30                      THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE  WITHOUT  NOTICE
31                      AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
32                      CORPORATION.  DIGITAL EQUIPMENT CORPORATION ASSUMES NO  RESPONSIBILITY
33                      FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.
34
35                      THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE  PURCHASER
36                      UNDER  A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED
37                      (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR  USE  IN  SUCH
38                      SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.
39
40                      DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR
41                      RELIABILITY  OF  ITS  SOFTWARE  ON  EQUIPMENT  THAT IS NOT SUPPLIED BY
42                      DIGITAL.
43
44                              COPYRIGHT (C) 1974,1979 BY DIGITAL EQUIPMENT CORPORATION
```

HISTORY

1.0 DECO CZLAC-F-0

1.1.0 CLOSED PROBLEM REPORT AA3318

1.1.1 IT WAS REPORTED THAT A DL11-A OPERATED AT 110 BAUD CAUSED FAILURE IN THE AREAD ROUTINE BECAUSE THE 200 MSEC. DELAY IS NOT OF SUFFICIENT DURATION TO ALLOW SETTING OF THE RECEIVER REGISTER STATUS "DONE" BIT THROUGH THE MAINTENANCE BIT FACILITY. THE TIME DELAY WAS INCREASED FROM 200 TO 600 MSEC.

1.2.0 CLOSED PROBLEM REPORT AA3643

1.2.1 TESTS 56,57,60,61,62,63,64,65,AND 66 DO NOT RUN PROPERLY WHEN RUN ON AN LSI-11. THIS PROBLEM WAS RESOLVED BY CHANGING THE BRANCH AFTER THE CHAIN COMMAND TO GO BACK TO TEST FOR THE LSI-11 SWITCH IN ORDER TO EFFECT THE APPROPRIATE ACTION DURING EACH TEST.

1.2.2 SECOND TIME-OUT IN TEST 64 ALLOWS EXCESSIVE WAIT FOR OPERATOR RESPONSE. THE TIME DELAY WAS REDUCED FROM "177777" TO "600".

1.2.3 COMMON ROUTINE TYPE DOES NOT SAVE THE CONTENTS OF R0 RESULTING IN THE LOSS OF THIS INFORMATION AND CONSEQUENT FAILURE. INSTRUCTIONS WERE INCLUDED TO SAVE THE CONTENTS OF R0 ON ENTRY INTO THE ROUTINE AND TO RESTORE THEM UPON EXIT.

1.2.4 LOSS OF STACK CONTENTS FOR NON-LSI-11 COMPUTERS DUE TO INCORRECT SEQUENCE OF INSTRUCTIONS IN TEST 65 WAS ALSO REPORTED IN PROBLEM REPORT AA3803. REFER TO 1.3.1.

1.3.0 CLOSED PROBLEM REPORT AA3803

1.3.1 TESTING OF NON-LSI-11 COMPUTERS RESULTS IN THE PROGRAM HANGING UP BECAUSE THE STACK GETS POPPED AWAY IN TEST 65. THE BRANCH AFTER THE TEST FOR THE LSI-11 SWITCH IN TEST 65 SHOULD GO TO THE CHAIN COMMAND FOR PROPER EXIT FROM THE TEST FOR NON-LSI-11 MACHINES. THIS CHANGE SUPERSEDES THE CHANGE RELEASED IN DEPO MD-11-DZLAC-D-1.

1.4.0 PROBLEM REPORT AA2390

1.4.1 PROCESSORS WITHOUT HARDWARE SWITCH REGISTERS ARE NOT ABLE TO LOOP ON ERROR WITH THIS PROGRAM. INSTALLED ^G FUNCTION TO PERMIT CHANGING SWITCH REGISTER SETTING DURING DIAGNOSTIC TESTING.

TABLE OF CONTENTS

1.0 ABSTRACT

THIS DIAGNOSTIC IS DIVIDED INTO THREE BASIC SECTIONS:

1. A CHECK OF THE CONSOLE TERMINAL INTERFACE LOGIC.

2. A CHECK OF THE PRINTING CHARACTERISTICS AND CONTROL LOGIC.

3. AN ECHO PORTION DESIGNED TO CHECK THE KEYBOARD AND TO AID IN THE DIAGNOSIS OF TERMINAL PROBLEMS.

PATTERNS USED BY THE PRINTING TESTS WERE CHOSEN FOR EASE OF VISUAL VERIFICATION. THE ECHO TESTS WERE DESIGNED FOR MAXIMUM FLEXIBILITY, WITH TEST 24 ALLOWING ANY DESIRED PATTERN TO BE USED.

2.0 REQUIREMENTS

2.1 EQUIPMENT AND ASSIGNMENTS

THE DIAGNOSTIC IS WRITTEN TO RUN ON ALL MODELS OF THE PDP-11 COMPUTER WITH EITHER A KL11 OR DL11 CONSOLE TERMINAL INTERFACE. THE DIAGNOSTIC IS PRESET TO TEST UP TO 16 ADDITIONAL TERMINALS (ON DL11'S) ASSIGNED BETWEEN ADDRESSES 776500 AND 776676. THIS PRESET QUANTITY (16) AND PRESET ADDRESS (776500) CAN BE CHANGED BY DEPOSITING THE QUANTITY IN DLNR AND THE STARTING ADDRESS IN DLADR. FOR EXAMPLE, TO ALLOW FOR UP TO 31 ADDITIONAL TERMINALS, THE ADDRESS 775610 COULD BE PLACED INTO DLADR AND THE OCTAL EQUIVALENT OF 31, I.E., (37) WOULD BE PLACED INTO DLNR. THE NUMBER OF ADDITIONAL DL11'S ACTUALLY TESTED WILL BE ADJUSTED AUTOMATICALLY DOWNWARD BASED UPON THE FIRST DL11 ADDRESS (WITHIN THE IMPLIED RANGE) FOUND TO BE UNRESPONSIVE. THUS IF THERE IS NO DL11 PRESENT TO MATCH THE ADDRESS IN DLADR ONLY THE CONSOLE TERMINAL WILL BE TESTED. THEREFORE, ALL DL11'S IN EXCESS OF THE CONSOLE TERMINAL MUST HAVE CONTIGUOUS ADDRESS ASSIGNMENTS WITH THE LOWEST ADDRESS CORRESPONDING TO THE VALUE IN DLADR.

THE CONSOLE TERMINAL (ASSIGNED STANDARD) CAN BE REASSIGNED BY PLACING THE ADDRESS OF ITS RECEIVER STATUS REGISTER INTO CONADD AND ITS RECEIVER INTERRUPT VECTOR INTO CONVEC. THIS REASSIGNMENT CAN BE MADE TO A TERMINAL WITHIN THE SET OF TERMINALS IMPLIED BY DLNR AND DLADR WITHOUT ADVERSE EFFECT. NOTE THAT A TERMINAL WITH A SLOWER SPEED (IF ANY) WILL DETERMINE THE SPEED AT WHICH ALL OF THE TERMINALS ARE TESTED. SUCH A TERMINAL SHOULD GENERALLY BE EXCLUDED FROM THE TEST, OR TESTED SEPARATELY. (REFER TO THE SYMBOL DEFINITIONS IN THE LISTING FOR THE ABOVE MENTIONED LOCATIONS.)

2.2 STORAGE

THE DIAGNOSTIC PROGRAM USES ALL OF 4K OF MEMORY WITH EXCEPTION OF THE AREA USED BY THE ABSOLUTE LOADER.

2.3 PRELIMINARY PROGRAMS

ANY APPLICABLE PDP-11 DIAGNOSTICS SHOULD BE RUN ON THE PROCESSOR. IF ANY ERRORS ARE ENCOUNTERED DURING THE INTERFACE CHECK, REFER TO THE APPROPRIATE INTERFACE DIAGNOSTIC FOR FURTHER HELP IN LOCATING THE PROBLEM IF NEEDED.

2.4 ADDITIONAL PROGRAMS

THIS DIAGNOSTIC IS FOR VERIFICATION OF BASIC TERMINAL FUNCTIONS ONLY. IF THE TERMINALS UNDER TEST HAVE HARDWARE OPTIONS INSTALLED RUN DIAGNOSTIC MAINDEC-11-DZLAF-A , THE LA36 TERMINAL OPTIONS TEST.

3.0 LOADING PROCEDURE AND INITIALIZATION

LOAD THE LA36 DIAGNOSTIC PROGRAM TAPE FOLLOWING NORMAL PROCEDURES. BEFORE STARTING THE PROGRAM, REFER TO THE DESCRIPTION OF THE ROUTINE "DLY". TIME DELAYS USED BY THE PROGRAM ARE A FUNCTION OF THE CPU MODEL AND MEMORY TYPE AND SHOULD BE SET-UP BEFORE RUNNING THE DIAGNOSTIC. THE ROUTINE IS PRESET FOR A PDP-11/05 WITH CORE MEMORY. REFER TO SECTION 2.1 FOR NON-STANDARD TERMINAL ADDRESSES AND FOR TESTING MULTIPLE DL11 INTERFACES.

IF A HARDWARE SWITCH REGISTER DOES NOT EXIST, THE PROGRAM WILL USE THE CONTENTS OF LOCATION 176 AS THE VALUE OF THE SWITCHES. WHEN THE PROGRAM STARTS UP, IT WILL ASK FOR A VALUE TO BE ENTERED BEFORE THE PROGRAM WILL PROCEED.

IF THE CPU IS AN LSI-11, 11/03 BE SURE TO SET SWITCH REGISTER BIT 9 TO A 1. SPECIAL TESTS ARE RUN ON THE DLV11 INTERFACE.

4.0 STARTING PROCEDURE

4.1 STARTING ADDRESSES

200(8) = RUN WITH SWITCH REGISTER CONTROL
- PERFORM CONSOLE TERMINAL I/O TESTS.

204(8) = RUN WITH SWITCH REGISTER CONTROL
- SKIP CONSOLE TERMINAL I/O TESTS.

210(8) - RUN WITH KEYBOARD CONTROL
- PERFORM CONSOLE TERMINAL I/O TESTS.

214(8) - RUN WITH KEYBOARD CONTROL
- SKIP CONSOLE TERMINAL I/O TESTS.

4.2 SWITCH REGISTER CONTROL WITH I/O TESTS

A. SET THE SWITCH REGISTER TO 200(8) AND PRESS THE LOAD ADDRESS SWITCH.

B. SET SWITCH REGISTER BIT 9 TO A 1 IF THE PROCESSOR IS AN LSI-11, 11/03. REFER TO SECTION 5.1.5.

C. SET THE SWITCH REGISTER BITS 7-0 EQUAL TO THE PAPER WIDTH IN TERMS OF THE NUMBER OF COLUMNS (OCTAL). REFER TO SECTION 5.1.8.

D. SET THE SWITCH REGISTER BIT 8 EQUAL TO 1 OR 0 AND PRESS THE START SWITCH. A MESSAGE WILL BE PRINTED INDICATING THE NUMBER OF DL11'S BEING TESTED. REFER TO SECTION 5.1.6.

E. IF BIT 8 WERE ZERO WHEN STARTING, THE PRINTER TESTS ARE EXECUTED SEQUENTIALLY, AFTER THE ENTIRE SERIES OF I/O TESTS ARE EXECUTED.

F. IF BIT 8 WAS SET WHEN THE START SWITCH WAS PRESSED, THE ENTIRE SERIES OF I/O TESTS WILL BE EXECUTED AND THE CPU WILL HALT AT LOCATION SELHLT. THE PROGRAM WILL THEN BE WAITING FOR CONTROL VIA THE SWITCH REGISTER.

4.3 SWITCH REGISTER CONTROL - WITHOUT I/O TESTS

SAME AS SECTION 4.2 EXCEPT IN STEP A, SET THE SWITCH REGISTER TO 204(8).

4.4 KEYBOARD CONTROL - WITH I/O TESTS

A. SET THE SWITCH REGISTER TO 210(8) AND PRESS THE LOAD ADDRESS SWITCH.

B. SET THE SWITCH REGISTER BITS 7-0 EQUAL TO THE PAPER WIDTH IN TERMS OF THE NUMBER OF COLUMNS (OCTAL). REFER TO SECTION 5.1.8.

C. SET SWITCH REGISTER BIT 9 TO A 1 IF THE PROCESSOR IS AN LSI-11, 11/03. REFER TO SECTION 5.1.5.

D. SET SWITCH 8 AND PRESS THE START SWITCH. A MESSAGE WILL BE PRINTED INDICATING THE NUMBER OF DL11'S BEING TESTED. REFER TO SECTION 5.1.6.

E. IF BIT 8 WAS ZERO WHEN STARTING, THE PRINTER TESTS ARE EXECUTED SEQUENTIALLY AFTER THE ENTIRE SERIES OF I/O TESTS ARE EXECUTED.

F. IF BIT 8 WERE SET WHEN THE START SWITCH WAS PRESSED, THE ENTIRE SERIES OF I/O TESTS WILL BE EXECUTED FOLLOWED BY THE SELECT TEST MESSAGE. THE PROGRAM WILL THEN BE WAITING FOR A TEST SELECTION VIA ANY TERMINAL KEYBOARD. REFER TO SECTION 5.2.

4.5 KEYBOARD CONTROL - WITHOUT I/O TESTS

SAME AS SECTION 4.4 EXCEPT IN STEP A, SET THE SWITCH REGISTER TO 214 (8).

5.0 OPERATING PROCEDURE

THE PROGRAM CAN BE CONTROLLED IN EITHER OF TWO METHODS: BY THE CONSOLE SWITCH REGISTER OR FROM THE KEYBOARD OF THE TERMINAL(S) UNDER TEST.

5.1 SWITCH REGISTER CONTROL

THE 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 HAL 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 11 DOWN WILL CAUSE THE PROGRAM TO RESUME NORMAL OPERATION AT THE COMPLETION OF THE TEST.

5.1.5 SWITCH 9

PLACING SWITCH 9 UP AT THE START OF THE TEST WILL CAUSE AN AUTOMATIC CHANGE IN THE DELAY TIMING, AND THE EXECUTION OF SPECIAL DLV11 I/O TESTS. THE DLV11 HAS NO MAINTENANCE MODE AND WILL CAUSE 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, AND 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 LEAVE SWITCHES 7 TO 0 DOWN TO TEST THE FULL 132(10) COLUMNS.

5.2 KEYBOARD CONTROL

THE PROGRAM WILL BE UNDER KEYBOARD CONTROL WHENEVER THE DIAGNOSTIC IS STARTED AT LOCATION 210 OR 214. SWITCHES ON THE CONSOLE SWITCH REGISTER WILL HAVE NO EFFECT WHEN UNDER TERMINAL CONTROL EXCEPT FOR SWITCH 15. THE I/O TESTS CANNOT BE SELECTED WHEN UNDER KEYBOARD CONTROL.

TO STOP A TEST AT ANY TIME, TYPE THE "RUBOUT" OR "DELETE" KEY ON ANY KEYBOARD. ANY TERMINAL MAY STOP THE TEST AND SELECT THE NEXT TEST IF SWITCH 13 IS DOWN. WHEN A TEST IS STOPPED BY TYPING A "RUBOUT" OR "DELETE", THE TEST WILL TERMINATE AND THE FOLLOWING MESSAGE WILL BE TYPED:

SELECT TEST NUMBER

AT THIS TIME, TYPE THE DESIRED TEST NUMBER FOLLOWED BY ANY ONE OF THE FOLLOWING CONTROL CHARACTERS:

. (PERIOD)	= RUN THE SELECTED TEST ONCE AND RETURN FOR ANOTHER TEST SELECTION.
L	= LOOP ON THE SELECTED TEST UNTIL A "RUBOUT" IS TYPED.
S	= START THE TEST SEQUENCE WITH THE SELECTED TEST. CONTINUE TO LOOP ON THE PRINTING TEST SEQUENCE UNTIL A "RUBOUT" IS TYPED.

THE "L" OR "S" MAY BE EITHER UPPER OR LOWER CASE, BUT THE TEST NUMBER MUST ALWAYS BE A 2 DIGIT OCTAL NUMBER. THE TEST NUMBER AND TERMINATOR ARE ECHOED BY THE PROGRAM, THUS EACH CHARACTER WILL BE PRINTED TWICE IF THE TERMINAL IS IN HALF DUPLEX. FOR ALL ECHO TESTS, THE "L" AND "S" WILL ONLY RUN THE TEST ONCE (THE SAME AS IF TYPING A PERIOD). FOR ALL OPTION TESTS, THE "S" WILL 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.

5.3 ^G FUNCTION

THE PROGRAM WILL SENSE ^G COMMANDS WHEN STARTED @ LOCATION 200 OR 204. CONSOLE SWITCH REG. VALUES WILL BE ENTERED FROM THE TTY AT START UP TIME WITH THE OLD VALUE INDICATED. TO CHANGE A VALUE TYPE ^G. IF TEST WAS STARTED @ LOC 200, AT THE END OF A PRINT TEST THE NEW VALUE CAN BE ENTERED. IF TEST WAS STARTED AT LOC 204, DATA CAN BE CHANGED BEFORE TEST FINISHES. ^U ALLOWS INCORRECT DATA ENTRIES TO BE DELETED AND A NEW ENTRY STARTED.

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*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.

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 0'S AND SPACES, STARTING WITH A 0. AT THE END OF THE LINE THE PRINT HEAD IS RETURNED TO THE LEFT MARGIN WITH A CARRIAGE RETURN. THE SPACES ARE THEN FILLED IN BY SPACING THE PRINT HEAD OUT FROM THE LEFT MARGIN TO THE FIRST SPACE, PRINTING AN "X", AND EXECUTING A CARRIAGE RETURN. THIS PATTERN IS REPEATED UNTIL THE LINE IS COMPLETED. CHECK TO SEE THAT ALL X'S ARE IN THE MIDDLE OF THE SPACE BETWEEN THE TWO ZEROES ON EITHER SIDE OF IT.

EXAMPLE:

```
0X0X0X0X0X0X0X0X0X0X0X0X0X0X0X0X
```

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THIS TEST WILL PRINT A LINE OF 0'S AND SPACES, THEN PRINT A DIAGONAL LINE OF X'S. TO CORRECTLY CHECK THE ENCODER, THE AUTO LINE FEED OPTION SHOULD BE DISABLED.

EXAMPLE:

```
0 0 0 0 0 0 0 0 0 0
 X
   X
     X
       X
         X
           X
             X
               X
                 X
```

6.1.5 TEST 4 - MULTIPLE LINE FEED TEST

THIS TEST CHECKS THE LINE FEED CAPABILITY OF THE PRINTER BY SENDING VARIOUS GROUPS OF LINE FEEDS INTERSPACED WITH REFERENCE LINES. THE NUMBER PRINTED AS THE REFERENCE LINE INDICATES THE NUMBER OF LINE FEEDS THAT FOLLOW. THE FIRST AND LAST LINES ALSO CONTAIN A STRING OF DASHES AS REFERENCE POINTS FOR MEASURING THE TOTAL DISTANCE BETWEEN THE TWO DASHED LINES, I.E., 63(10) LINES.

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY CARRIAGE RETURN, THE NUMBER PRINTED WILL INDICATE ONE LESS THAN THE NUMBER OF LINE FEEDS (THE NUMBER OF BLANK LINES) THAT FOLLOW. THE TOTAL DISTANCE BETWEEN THE TWO DASHED LINES WILL THEN BE 69 LINES.

EXAMPLE:

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

04



08









16
   \
    15 BLANK LINES
   /
32
   \
    31 BLANK LINES
   /
00-------------------------------------------------
```

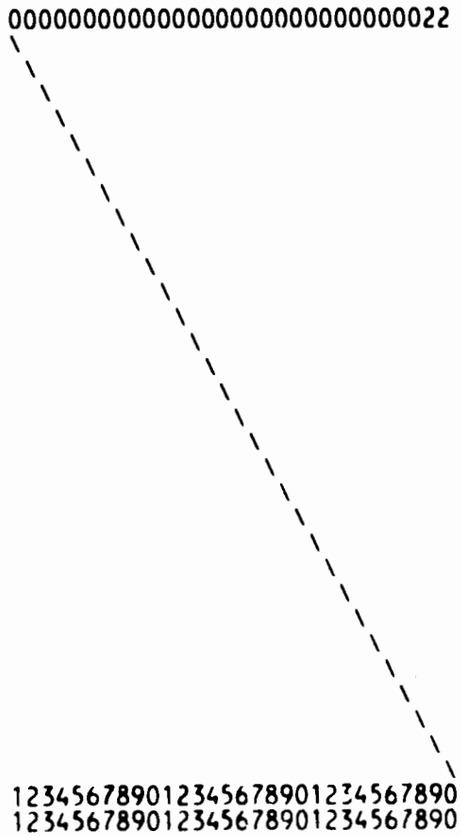
6.1.6 TEST 5 - SINGLE LINE FEED TEST

THIS TEST IS DESIGNED TO CHECK THE TIMING OF SINGLE LINE FEEDS AND THE CAPABILITY OF DOING LINE FEEDS IN ALL COLUMNS. TWO REFERENCE LINES ARE USED BY THIS TEST (AND TEST 6) WHICH ALSO CAN BE USED TO EASILY CHECK THE NUMBER OF COLUMNS THE PRINTER IS PRINTING.

THE FIRST REFERENCE LINE CONTAINS 130(10) ZEROES FOLLOWED BY TWO 2'S IF TESTING 132(10) COLUMNS. IF LESS THAN 132 COLUMNS, THE LINE WILL CONTAIN 0'S FOR TWO LESS THAN THE MAXIMUM NUMBER OF COLUMNS FOLLOWED BY THE TWO 2'S. THIS REFERENCE LINE IS A QUICK CHECK FOR 132(10) COLUMNS IF TESTING THE FULL 132(10) COLUMNS. THE SECOND REFERENCE LINE PRINTS A STRING OF NUMBERS (1 TO 9 & 0) REPEATED TO THE MAXIMUM COLUMN. THIS LINE, AGAIN, CAN BE USED AS A QUICK CHECK OF THE NUMBER OF COLUMNS.

THE LINE FEED TEST IS ACCOMPLISHED BY: PRINTING THE FIRST REFERENCE LINE OF 0'S AND TWO 2'S; THEN EITHER SENDING 60(10) 3'S, IF TESTING 132(10) COLUMNS, OR WAITING 1.8 SECONDS FOR AN LCV, IF TESTING LESS THAN 132(10) COLUMNS. IF TESTING 132(10) COLUMNS, NOTHING SHOULD HAPPEN, EXCEPT FOR AN LCV, AT THE END OF THE LINE. THE 3'S SHOULD BE LOST AND NEVER PRINTED. AFTER THE LCV, WITH THE PRINT HEAD AT THE EXTREME RIGHT, A CARRIAGE RETURN - LINE FEED WILL BE SENT FOLLOWED BY REPEATED BACKSLASHES "\" AND LINEFEEDS TO PRINT A DIAGONAL LINE DOWN THE PAPER. WHEN A BACKSLASH IS PRINTED IN THE MAXIMUM COLUMN, A CARRIAGE RETURN WILL BE SENT IMMEDIATELY AFTER THE LINE FEED AND THE SECOND REFERENCE LINE OF SEQUENTIAL NUMBERS WILL BE PRINTED. AFTER COMPLETING THE LINE, A CARRIAGE RETURN - LINE FEED WILL BE SENT AND THE PROGRAM WILL WAIT ONE SECOND FOR THE CARRIAGE RETURN FUNCTION TO COMPLETE. AFTER THE DELAY, THE REFERENCE LINE WILL BE REPEATED, THE LAST LINE BEING GUARANTEED TO BE CORRECT. ANY TIMING PROBLEMS DURING THE LINE FEEDS WILL SHOW AS MISPRINTS OR MISSING CHARACTERS DURING THE FIRST 16(10) CHARACTERS OF THE MIDDLE REFERENCE LINE. ALSO, ANY PAPER FEED PROBLEMS WILL CAUSE MISALIGNMENT OF THE SLASHES FORMING THE DIAGONAL LINE.

EXAMPLE:



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
          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          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 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 LOGARITHMIC INCREASE:

NEW DELAY = OLD DELAY + OLD DELAY/16 + OLD DELAY/128

THE LAST TWO LINES DO JUST THE REVERSE. THE TIME INTERVAL BETWEEN CHARACTERS IS DECREASED FROM 1.8 SECONDS TO 30(10) MILLISECONDS USING THE FOLLOWING FORMULA TO AGAIN CREATE A LOGARITHMIC DECREASE:

NEW DELAY = OLD DELAY - OLD DELAY/16 - OLD DELAY/128

LOOK FOR POSSIBLE MISALIGNMENT OF THE CHARACTERS OR SPACES BETWEEN CHARACTERS AS AN INDICATION OF TIMING PROBLEMS.

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THERE WILL BE A BLANK LINE BETWEEN EACH PRINTED LINE.

EXAMPLE:

```
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
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
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 OCCURRED ON THE LINE. THE TEST PRINTS A MESSAGE:

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

AFTER THE TEST MESSAGE IS PRINTED, TYPE ANY PRINTABLE CHARACTER ON THE KEYBOARD. THE CHARACTER WILL BE ECHOED AND THE BELL SHOULD RING. THE MESSAGE WILL THEN BE TYPED AGAIN. TYPE THE "RUBOUT" KEY TO TERMINATE THE TEST AT ANY TIME.

6.4 STANDARD I/O TESTS

THESE TESTS ARE DESIGNED AS A BRIEF CHECK OF THE CONSOLE TERMINAL INTERFACE LOGIC. 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.

```
        .TITLE  CZLACFO LA36 TERM (DL11 & KL11)
;
;LA36 DIAGNOSTIC (DL11 & KL11 INTERFACE)
;
;
;AUTHORS:    ROBERT W. BAKER
;            R. QUENNEVILLE
;            RALPH A. SCHAUBER
;            JOHN V. CHATALIAN

;
;COPYRIGHT 1974,1979  DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754


        .SBTTL  SWITCH REGISTER OPTIONS
;
;SWITCH          POSITION        FUNCTION
;
;  15            UP (1)          HALT AT COMPLETION OF CURRENT TEST
;                DOWN (0)        CONTINUE NORMAL TEST SEQUENCE
;
;  14            UP (1)          CONTINUE ON ERROR
;                DOWN (0)        HALT ON ERROR
;
;  13            UP (1)          DRIVE ONLY CONSOLE TERMINAL
;                DOWN (0)        DRIVE ALL TERMINALS
;
;  11            UP (1)          LOOP ON INDIVIDUAL TEST
;                DOWN (0)        NORMAL TEST SEQUENCE
;
;  09            UP (1)          CPU TYPE IS AN LSI-11, PDP-11/03
;                DOWN (0)        ALL OTHER PDP-11 CPU'S
;
;  08            UP (1)          HALT TO SELECT TEST AT END OF CURRENT TEST
;                DOWN (0)        LOOP ON TEST SEQUENCE
;
;  05-00                         TEST # SELECTION
;
;  07-00                         # OF COLUMNS AT START-UP
```

```
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
```

```
        .SBTTL  SPECIAL OPERATIONAL INFORMATION

;
;1.--   THE STANDARD CONSOLE TERMINAL INTERRUPT VECTOR AND REGISTER
;       ADDRESSES ARE USED. TO REDEFINE THE LOCATION OF THE CONSOLE
;       TERMINAL THE SYMBOLIC LOCATIONS "CONADD" AND "CONVEC" SHOULD
;       BE CHANGED BEFORE START UP.
;
;2.--   BEFORE START UP REFER TO THE DESCRIPTION OF THE ROUTINE "DLY".
;       TIMING IS A FUNCTION OF THE PDP-11 MODEL AND MEMORY TYPE AND
;       SHOULD BE SET UP BEFORE RUNNING THE DIAGNOSTIC.
;
;3.--   IF CPU IS A PDP-11/03 , LSI-11 SET SWITCH REGISTER
;       BIT 09 TO A 1. SPECIAL TESTS ARE RUN ON THE DLV11.
;
;4.--   SYSTEMS WITHOUT A HARDWARE SWITCH REGISTER SHOULD USE
;       MEMORY LOCATION 176 AS A SOFTWARE SWITCH REGISTER.
;
;5.--   THIS DIAGNOSTIC IS FOR VERIFICATION OF BASIC TERMINAL
;       FUNCTIONS ONLY. IF THE TERMINAL UNDER TEST HAS HARDWARE
;       OPTIONS INSTALLED RUN DIAGNOSTIC MAINDEC-11-DZLAF-A, THE
;       LA36 TERMINAL OPTIONS TEST.
;
;6.--   CONTROL G FUNCTION PERMITS CHANGING THE SOFTWARE SWITCH REGISTER
;       AT ANY TIME AFTER A  TEST.  ALSO, AT TEST START, IF THERE IS
;       NO HARDWARE SWITCH REGISTER, A MESSAGE WILL ASK FOR SOFTWARE
;       SWITCH REGISTER SETTINGS WHICH MUST BE PROVIDED VIA TTY KEYBOARD.
;       CONTROL U ALLOWS RESTARTING THE PROCESS IF AN ERROR WAS MADE.
;       THE NUMBER MUST BE A SIX DIGIT OCTAL VALUE.
;


;                       ECO HISTORY
;CHGF1 - NEW STORAGE LOCATIONS FOR REFERENCE IN ^G MODE.
;CHGF2 - CHANGE START UP FOR ENTERING SOFTWARE SWITCH REGISTER INFO.
;CHGF3 - CHECK FOR ^G BETWEEN TESTS.
;CHGF4 - CLEAR INTERRUPT ENABLE BIT TO PREVENT PREMATURE INTERRUPTS.
;CHGF5 - ROUTINE TO CHECK PRINTOUT, UPDATE SOFTWARE SWITCH REG.
;CHGF6 - NEW MESSAGES FOR SOFTWARE SWITCH REG ROUTINES.
;CHGF7 - COMBINING LSI-11 TESTS TO ELIMINATE CLEARING DONE BIT IN TKS
```

```
1225                                    .SBTTL  SYSTEM EQUATES
1226                                ;
1227                                ;
1228                                ;       REGISTER EQUATES
1229                                ;
1230            000000                      R0=%0
1231            000001                      R1=%1
1232            000002                      R2=%2
1233            000003                      R3=%3
1234            000004                      R4=%4
1235            000005                      R5=%5
1236            000006                      SP=%6
1237            000007                      PC=%7
1238            177776                      PSW=177776
1239                                ;
1240                                ;       SYSTEM EQUATES
1241                                ;
1242            000001                      BIT0=1
1243            000002                      BIT1=2
1244            000004                      BIT2=4
1245            000010                      BIT3=10
1246            000020                      BIT4=20
1247            000040                      BIT5=40
1248            000100                      BIT6=100
1249            000200                      BIT7=200
1250            000400                      BIT8=400
1251            001000                      BIT9=1000
1252            002000                      BIT10=2000
1253            004000                      BIT11=4000
1254            010000                      BIT12=10000
1255            020000                      BIT13=20000
1256            040000                      BIT14=40000
1257            100000                      BIT15=100000
1258            000000                      OPEN=0
1259            040000                      SCOPSW=BIT14            ;SCOPE SWITCH
1260            004000                      NITRSW=BIT11            ;TEST LOOP SWITCH
1261            005726                      POPSP=5726              ;POP STACK ONCE
1262            022626                      POPSP2=22626            ;POP STACK TWICE
1263            000340                      PRTY7=340               ;PRIORITY LEVEL DEFINITIONS
1264            000200                      PRTY4=200
1265            000200                      ACRLF=200
1266            001000                      LSI11=BIT9              ;FLAG FOR LSI-11,11/03
1267                                ;
1268                                ;       PROGRAM TRAP EQUATES
1269                                ;
1270            104000                      TYPE=EMT+0
1271            104001                      ERROR=EMT+1
1272            104002                      EHALT=EMT+2
1273            104003                      STRDRV=EMT+3
1274            104004                      STPCHV=EMT+4
1275            104005                      CHAIN=EMT+5
1276            104006                      CHALT=EMT+6
1277            104007                      TYPEM=EMT+7
1278            104010                      DELAY=EMT+10
1279            104011                      TTYCTL=EMT+11
1280            104012                      CRLF=EMT+12
```

```
1281    104013          SCRLF=EMT+13
1282    104014          LF=EMT+14
1283    104015          PRINTC=EMT+15
1284    104016          PRTHDR=EMT+16
1285    104017          PRNT=EMT+17
1286    104020          READ=EMT+20
1287    104021          AREAD=EMT+21
1288    104022          CR=EMT+22
1289    104023          BTOASC=EMT+23
1290    104024          FORWD=EMT+24
1291    104025          READC EMT+25
```

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

                000000                  .=0

000000  000002                          .+2                     ;UNASSIGNED TRAP
000002  000000                          HALT
000004  000006                MACHER:   .+2                     ;SP OVERFLOW, BUS ERROR TRAP
000006  000000                          HALT
000010  000012                          .+2                     ;RESERVED INSTRUCTION TRAP
000012  000000                          HALT
000014  000016                          .+2                     ;TRACE TRAP
000016  000000                          HALT
000020  000022                          .+2                     ;TRAP TO CALL IOX
000022  000000                          HALT
000024  000026                          .+2                     ;POWER FAIL TRAP
000026  000000                          HALT
000030  003010                          EMTINT                  ;EMT TRAP
000032  000340                          PRTY7

                000042                  .=42

000042  000000                          0

                000046                  .=46

000046  011524                          LOGICAL

                000052                  .=52

000052  010000                          010000

                000174                  .=174

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

000200  000137  001026                  JMP     START           ;START UP WITH I/O TESTS RUNNING
000204  000137  000754                  JMP     START1          ;START UP, SKIP ALL I/O TESTS
000210  000137  000772                  JMP     START2          ;START UP TERMINAL CONTROL WITH I/O TESTS
000214  000137  001010                  JMP     START3          ;START UP TERMINAL CONTROL WITHOUT I/O TESTS

                000600                  .=600

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

```
                                        .SBTTL  SYMBOL DEFINITIONS
                                ;
                                ;
                                ;
000602  177560                CONADD: 177560                  ;ADDR OF CONSOLE RECEIVER STATUS REG
000604  000060                CONVEC: 60                      ;CONSOLE TERMINAL INTERRUPT VECTOR
000606  176500                DLADR:  176500                  ;ADDRESS OF FIRST DL11, DEFAULT TO DL11-A,B
                                                            ;IF DL11-C,D,E,, THEN
                                                            ;SET TO 175610 FOR FIRST 16 (OF 31) OR
                                                            ;SET TO 176000 FOR LAST 16 (OF 31)
                                                            ;OR SET OTHER DESIRED START ADDRESS
000610  000020                DLNR:   16.                     ;# OF DL11'S TO BE INITIALLY ASSUMED
000612  177560                TKS:    177560                  ;CONSOLE RECEIVER STATUS REG
000614  177562                TKB:    177562                  ;CONSOLE RECEIVER BUFFER
000616  177564                TPS:    177564                  ;CONSOLE TRANSMITTER STATUS REG
000620  177566                TPB:    177566                  ;CONSOLE TRANSMITTER BUFFER
000622  000060                TKVTR:  60                      ;C.T. RECEIVER INTERRUPT VECTOR
000624  000200                TKLVL:  PRTY4                   ;C.T. RECEIVER PRIORITY LEVEL
000626  000064                TPVTR:  64                      ;C.T. TRANSMITTER INTERRUPT VECTOR
000630  000200                TPLVL:  PRTY4                   ;C.T. TRANSMITTER PRIORITY LEVEL
000632  000000                FSTDL:  OPEN                    ;ADDRESS OF FIRST ACTIVE DL11
000634  000000                CNTLSW: OPEN                    ;CONSOLE TERMINAL CONTROL SWITCH
000636  000000                RTNNO:  OPEN                    ;CONTAINS CURRENT TEST NUMBER
000640  000000                NXTST:  OPEN                    ;CONTAINS ADDRESS OF NEXT TEST
000642  000000                SCOPTR: OPEN                    ;CONTAINS ADDRESS OF TEST SCOPE ENTRY
000644  000000                PRGID:  OPEN                    ;CONTAINS TEST PROGRAM INDICATORS
000646  000000                CRBUF:  OPEN
000650  000000                CTRA:   OPEN
000652  000000                WIDTH:  OPEN                    ;CURRENT PAPER WIDTH, BINARY
000654  000000                LEVEL:  OPEN                    ;LEVEL OF EXECUTION
000656  000000                DLCNT:  OPEN                    ;# OF MULTIPLE DL11S
000660  000000                ICTR:   OPEN                    ;I/O TEST ITERATION COUNT
000662  000000                REPT:   OPEN                    ;TEMP STORAGE FOR TESTS E021 & E022
000664  000000                BRCTR:  OPEN                    ;COUNTER FOR ROUTINE "AREAD"
000666  000000                COUNT3: OPEN                    ;COUNTER FOR ROUTINE "PRINTC"
000670  000000                XCSR:   OPEN                    ;ADDRESS OF MULTIPLE DL11 STATUS
000672  000251                TIMER:  251                     ;1 MSEC COUNTER FOR ROUTINE "DELAY"
000674  000000                SPCNT:  OPEN                    ;COUNTER FOR TEST ROUTINE "PT3"
000676  000000                CURTST: OPEN                    ;ADDRESS OF CURRENT TEST
000700  000000                TEMPCH: OPEN                    ;TEMP STOR FOR ECHO TESTS
000702  000000                PARITY: OPEN                    ;PARITY FLAG FOR RECEIVED CHAR
000704  000000                PCHAR:  OPEN                    ;CHAR CODE WITH PARITY BIT
000706  000000                LFCNT:  OPEN                    ;COUNTER FOR TEST ROUTINE "PT4"
000710  000000                INCHK:  OPEN                    ;CHECK FOR INPUT FLAG
000712  000000                TEMP:   OPEN                    ;TEMPORARY WORKING STORAGE
000714  177570                SR:     177570                  ;SW REG ADDRESS
000716  000000                CNTR:   OPEN                    ;TIME COUNTER FOR LSI-11 TESTS
                                ;******* CHGF1  **************
000720  177570                DISPLAY: 177570                 ;DISPLAY REGISTER
000722  000000                TIB:    OPEN                    ;TEMPORARY KYBD BUFFER STORAGE
000724  000000                TEMPST: OPEN                    ;TEMPORARY WORK LOCATION
000726  000000                COUNT:  OPEN                    ;LOOP COUNT FOR TEST
000730  000000                FILL:   OPEN                    ;ZERO FILL SWITCH
000732  000000                MODE:   OPEN                    ;NUMBER OF DIGITS TO TYPE
000734  000000                CNT:    OPEN                    ;ITERATION COUNT
                                ;*******************************
```

1397

```
                        .SBTTL  PROGRAM INITIALIZATION & CONTROL

                        ;********
                        ;COMMON HALT---WHEN IN SWITCH REGISTER CONTROL THE CPU
                        ;       WILL BE ADVANCED TO THIS COMMON HALT WHERE
                        ;       A NEW TEST WILL BE EXPECTED TO BE STARTED
                        ;********

000736  005737  000654  CHLT:   TST     LEVEL           ;TEST CURRENT LEVEL
000742  001403          SELHLT: BEQ     SELHLT          ;BRANCH IF 0, DO NOT HALT
000744  011600                  MOV     @SP,R0          ;PUT ADDRESS OF CALLER INTO R0
000746  005740                  TST     -(R0)
000750  000000                  HALT
000752  000002          SELHLT: RTI                     ;RETURN FROM INTERRUPT
000754  012737  177777  005356  START1: MOV     #177777,ATOX    ;FORCE END OF I/O TESTS
000762  012737  104006  001772          MOV     #CHALT,WAITF    ;FORCE SR CONTROL
000770  000424                  BR      STARTX
000772  012737  104011  001772  START2: MOV     #TTYCTL,WAITF   ;FORCE TERMINAL CONTROL
001000  012737  005406  005356          MOV     #AT1,ATOX       ;FORCE ALL I/O TESTS
001006  000415                  BR      STARTX
001010  012737  104011  001772  START3: MOV     #TTYCTL,WAITF   ;FORCE TERMINAL CONTROL
001016  012737  177777  005356          MOV     #177777,ATOX    ;FORCE END OF I/O TESTS
001024  000406                  BR      STARTX
001026  012737  005406  005356  START:  MOV     #AT1,ATOX       ;FORCE ALL I/O TESTS
001034  012737  104006  001772          MOV     #CHALT,WAITF    ;FORCE SR CONTROL
001042  012706  000600          STARTX: MOV     #SPBOT,SP       ;SET STACK POINTER
                        ;**************
001046  013746  000006  CHGF2:  MOV     6,-(SP)         ;SAVE CURRENT VECTOR
001052  013746  000004          MOV     4,-(SP)
001056  012737  001076  000004          MOV     #64$,@#4        ;SET UP LOC. 4 FOR NON-EXISTANT
                                                        ;MEMORY TRAP
001064  022777  177777  177622          CMP     #-1,@SR         ;REFERENCE HARDWARE SWITCH REGISTER
001072  001405                  BEQ     65$             ;IF IT = -1,USE SOFT SW REG
001074  000412                  BR      66$             ;THEN USE HARDWARE SW REG
001076  022626          64$:    CMP     (SP)+,(SP)+     ;CORRECT THE STACK, NO HDWE SW REG
001100  012737  000202  000672          MOV     #202,TIMER      ;ADJUST TIMER FOR LSI-11
001106  012737  000176  000714  65$:    MOV     #SWREG,SR       ;POINT TO SOFT SW REG
001114  012737  000174  000720          MOV     #DISPREG,DISPLAY  ;POINT TO SOFT DISP REG
001122  012637  000004  66$:    MOV     (SP)+,@#4       ;RESTORE VECTORS
001126  012637  000006          MOV     (SP)+,@#6
001132  022737  000176  000714  67$:    CMP     #SWREG,SR       ;IS SOFT SWREG SELECTED
001140  001002                  BNE     60$             ;IF NOT,BR OVER NEXT OPER
001142  004737  013752                  JSR     PC,CNTLU        ;SOFT SW REG INPUT ROUTINE
001146  013701  000622  60$:    MOV     TKVTR,R1
001152  012721  014176                  MOV     #TTINTS,(R1)+   ;VECTOR ADDRESS TO LOC 60
001156  013721  000624                  MOV     TKLVL,(R1)+     ;PRIORITY TO LOC 62
001162  005777  177426                  TST     @TKB            ;CLEAR DONE FLAG
001166  012777  000100  177416          MOV     #100,@TKS       ;SET TTY INTERRUPT ON
001174  005037  000710                  CLR     INCHK           ;ALLOW INPUT CHECKING
001200  012737  000006  000004          MOV     #6,MACHER       ;CLEAN UP
001206  005037  000644                  CLR     PRGID           ;INITIALIZE PROGRAM FLAGS
001212  005037  000634                  CLR     CNTLSW          ;INITIALIZE TERMINAL CONTROL SWITCH
001216  005037  000654                  CLR     LEVEL           ;INITIALIZE LEVEL
001222  012737  003546  000024          MOV     #PFAIL,24       ;SET ADDR POWER FAIL ROUTINE
001230  004737  003776                  JSR     PC,CONIT        ;SET UP CONSOLE TERMINAL ADDRESS
                        ;**************
```

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

001234  017701  177454          MOV     @SR,R1          ;PUT (SR) INTO R1
001240  042701  177400          BIC     #177400,R1      ;SAVE ONLY BITS 0-7
001244  020127  000204          CMP     R1,#204         ;TEST NO. COLUMN GT132
001250  003003                  BGT     2$              ;COLUMNS GT132, DEFAULT TO 132
001252  020127  000035  1$:     CMP     R1,#35          ;CHECK IF NO. COLUMNS LT 30
001256  101002                  BHI     3$              ;NOT LT 30 NOR GT 132
001260  012701  000204  2$:     MOV     #204,R1         ;COLUMNS LT 30 OR GT 132, DEFAULT
001264  010137  000652  3$:     MOV     R1,WIDTH        ;SAVE NO. COLUMNS IN WIDTH
001270  012700  014627          MOV     #HDR0,R0        ;ADDR TO STORE ASC11 COLUMN VALUE
001274  012702  000003          MOV     #3,R2           ;DO A 3 CHAR. CONVERSION
001300  104023                  BIOASC                  ;CONVERT NO. COLUMNS TO ASCII
001302  000401          4$:     BR      5$
001304  000410                  BR      6$
001306  012700  000000  5$:     MOV     #0,P0           ;TRANSMIT A
001312  104015                  PRINTC                  ;NULL CODE
001314  104007                  TYPEM                   ;TYPE PROGRAM TITLE FIRST TIME RUN
001316  014376                  STARTM
001320  012737  000240  001302  MOV     #NOP,4$
```

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

001326  012737  001406  000004  6$:     MOV     #END2A,MACHER   ;INITIALIZE TIME OUT TRAP
001334  013700  000606                  MOV     DLADR,R0        ;ADDRESS OF FIRST DL11 TO R0
001340  013701  000610                  MOV     DLNR,R1         ;SET DL CHECK COUNT
001344  005037  000656                  CLR     DLCNT           ;INITIALIZE DLCNT
001350  005710                  END3:   TST     (R0)            ;IS DL PRESENT?
001352  012737  001420  000004          MOV     #END2,MACHER    ;YES, RESET TIME OUT TRAP
001360  010037  000632                  MOV     R0,FSTDL        ;STORE ADDRESS OF FIRST DL11
001364  000401                          BR      2$              ;CONTINUE
001366  005710                  1$:     TST     (R0)            ;IS DL11 PRESENT
001370  062700  000010          2$:     ADD     #10,R0          ;POINTER AND DL11 ADDRESS
001374  005237  000656                  INC     DLCNT           ;INCREMENT COUNT OF DL11'S
001400  005301                          DEC     R1              ;DECREMENT DL CHECK COUNT, DONE?
001402  001407                          BEQ     END4            ;BRANCH IF DONE
001404  000770                          BR      1$              ;CHECK PRESENCE OF NEXT DL11
001406  005301                  END2A:  DEC     R1              ;DONE DL CHECK?
001410  001404                          BEQ     END4            ;YES, EXIT
001412  062700  000010                  ADD     #10,R0          ;NO, CHECK NEXT DL
001416  000754                          BR      END3            ;CONTINUE
001420  022626                  END2:   POPSP2                  ;DL11 NOT PRESENT
001422  013701  000656          END4:   MOV     DLCNT,R1        ;GET # DL11'S
001426  012700  014564                  MOV     #DL11S1,R0      ;ADR OF ASCII CHAR STORAGE
001432  012702  000002                  MOV     #2,R2           ;# OF ASCII CHARS
001436  104023                          BTOASC                  ;CONVERT NUMBER
001440  104007                          TYPEM                   ;TYPE MESSAGE
001442  014551                          DL11S

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

001444  005037  000636                  CLR     RTNNO           ;SET ROUTINE NO - 0
001450  005037  000654                  CLR     LEVEL           ;SET LEVEL - 0
001454  023727  005356  177777          CMP     ATOX,#177777    ;SEE IF I/O IS TO BE SKIPPED
001462  001517                          BEQ     SKIP
001464  012737  005354  000640          MOV     #ATO,NXTST      ;ADDRESS OF FIRST I/O TEST
001472  104024                          FORWD                   ;SET UP TEST PARAMETERS
001474  000177  177176                  JMP     @CURTST         ;GO TO I/O TEST ROUTINE
```

```
1527                                    ;**********
1528                                    ;CHAINN--   THIS PORTION IS THE COMMON RETURN
1529                                    ;           FOR ALL THREE CLASSES OF TESTS.
1530                                    ;
1531                                    ;       1--IF AN ERROR OCCURRED DURING AN I/O TEST THE
1532                                    ;          OPERATOR CAN CAUSE THAT TEST TO BE LOOPED
1533                                    ;          WITHOUT ANY FURTHER ERROR HALTS BY
1534                                    ;          SETTING THE "SCOPE" BIT (#14) ON THE SR=1.
1535                                    ;          RESETTING SR BIT 14 TO 0 WILL ALLOW THE
1536                                    ;          ERROR HALT TO OCCUR AGAIN IF IT STILL EXISTS.
1537                                    ;
1538                                    ;       2--IF THE OPERATOR IS IN THE MAINTENANCE MODE, (BIT 8=1
1539                                    ;          AT START UP TIME), AND BIT 11=1, THE SELECTED PROGRAM WILL
1540                                    ;          FIRST HALT. THEN HIT CONTINUE AND PROGRAM WILL LOOP CONTINUOUSLY.
1541                                    ;          IF BIT 8=0, AND BIT 11=0, THE SELECTED PROGRAM WILL BE
1542                                    ;          ADVANCED TO THE NEXT TEST IN ITS CLASS.  AS LONG AS BITS
1543                                    ;          8 AND 11 = 0, THE CLASS OF TESTS SELECTED WILL BE CON-
1544                                    ;          UOSLY SEQUENCED THROUGH.
1545                                    ;          IF BIT 8=1, AND BIT 11=0, THEN THE CPU WILL HALT AT LOC
1546                                    ;          SELHLT AND WAIT FOR THE NEXT TEST NUMBER TO BE SET IN
1547                                    ;          THE SWITCH REGISTER.
1548                                    ;**********
1549  001500                            CHGF3:
1550  001500  004737  013700           CHAINN: JSR     PC,CKSWR        ;SEE IF A ^G HAS BEEN GIVEN
1551  001504  032737  000001  000634           BIT     #1,CNTLSW       ;CHECK IF TERMINAL CONTROL
1552  001512  001401                           BEQ     1$              ;BRANCH IF NOT
1553  001514  104011                           TTYCTL                  ;GO TO TERMINAL CONTROL
1554  001516  005737  000644           1$:     TST     PRGID           ;TEST ERROR BIT IN PRGID
1555  001522  100016                           BPL     3$              ;BRANCH IF ERROR BIT NOT SET
1556  001524  032777  040000  177162           BIT     #SCOPSW,@SR     ;ERROR, CHECK IF SCOPE OPTION ON
1557  001532  001407                           BEQ     2$              ;BRANCH IF NO SCOPING
1558  001534  022737  177777  000642           CMP     #-1,SCOPTR      ;YES, CHECK IF OK TO SCOPE THIS TEST
1559  001542  001403                           BEQ     2$              ;BRANCH IF NOT OK
1560  001544  017716  177072                   MOV     @SCOPTR,@SP     ;PUT ADDR OF SCOPE ENTRY INTO STACK
1561  001550  000002                           RTI                     ;GO TO SCOPE ENTRY IN TEST
1562  001552  042737  100000  000644   2$:     BIC     #BIT15,PRGID    ;CLEAR ERROR IND. IN PRGID
1563  001560  005737  000654           3$:     TST     LEVEL           ;CHECK LEVEL
1564  001564  001405                           BEQ     4$              ;BRANCH IF LEVEL=0
1565  001566  032777  004000  177120           BIT     #NITRSW,@SR     ;TEST LOOP SWITCH ON ( 1)
1566  001574  001405                           BEQ     5$              ;BRANCH IF NO LOOP TEST
1567  001576  000002                           RTI                     ;GO BACK TO TEST
1568  001600  005337  000660           4$:     DEC     ICTR            ;DECREMENT TEST ITERATION COUNT
1569  001604  001407                           BEQ     6$              ;BRANCH IF COUNT=0
1570  001606  000002                           RTI                     ;NOT ZERO, REPEAT TEST
1571  001610  032777  000400  177076   5$:     BIT     #BIT8,@SR       ;TEST IF SEQUENCE TEST (BIT8)
1572  001616  001402                           BEQ     6$              ;BRANCH TO NEXT TEST IF BIT8 0
1573  001620  000137  001772                   JMP     WAITF           ;GO WAIT FOR MORE INPUT
1574  001624  022626                   6$:     POPSP2                  ;POP 2 OFF STACK
1575  001626  000240                   CHAINY: NOP                     ;THIS FORMERLY WAS RESET
1576  001630  005777  177060                   TST     @SR             ;CHECK SR
1577  001634  100003                           BPL     1$              ;BRANCH IF NO HALT WANTED
1578  001636  113700  000636                   MOVB    RTNNO,R0        ;CURRENT TEST NUMBER TO R0
1579  001642  000000                           HALT                    ;HALT (NOT FOR TEST SELECTION)
1580  001644  005737  000654           1$:     TST     LEVEL           ;TEST THE CURRENT LEVEL
1581  001650  001420                           BEQ     3$              ;BRANCH IF 0
1582  001652  012737  000006  000004           MOV     #6,MACHER       ;CLEAN UP
```

```
1583  001660  012706  000600                    MOV     #SPBOT,SP       ;SET UP STACK POINTER
1584  001664  104024                            FORWD                   ;SET UP VALUES FOR NEXT TEST
1585  001666  022737  177777  000640            CMP     #-1,NXTST       ;END OF I/O TESTS (=-1)
1586  001674  001004                            BNE     2$              ;BRANCH IF NOT END
1587  001676  012737  005354  000640            MOV     #ATO,NXTST      ;RESET NXTST TO FIRST I/O TEST
1588  001704  104024                            FORWD                   ;SET UP VALUES FOR NEXT TEST
1589  001706  000177  176764            2$:     JMP     @CURTST         ;GO TO TEST
1590  001712  022737  177777  000640    3$:     CMP     #-1,NXTST       ;END OF I/O TESTS (=-1)
1591  001720  001012                            BNE     NEXT            ;BRANCH IF NOT
1592  001722  032777  000400  176764    SKIP:   BIT     #BIT8,@SR       ;TEST IF WANT TEST SELECTION RIGHT AWAY
1593  001730  001016                            BNE     NEXT1           ;BRANCH IF NOT
1594  001732  052737  000200  000644            BIS     #BIT7,PRGID     ;BYPASS SCOPING
1595  001740  012737  007374  000640            MOV     #PTO,NXTST      ;PROD TESTING, GO TO PRINTER TESTS
1596  001746  012737  000006  000004    NEXT:   MOV     #6,MACHER       ;CLEAN UP
1597  001754  012706  000600                    MOV     #SPBOT,SP       ;SET UP STACK POINTER
1598  001760  104024                            FORWD                   ;SET UP NEXT TEST PARAMETERS
1599  001762  000177  176710                    JMP     @CURTST         ;GO TO ROUTINE
1600  001766  005237  000654            NEXT1:  INC     LEVEL
1601
1602                                            ;*********
1603                                            ;WAIT FOR FURTHER INSTRUCTIONS:
1604                                            ;       -LOAD PROGRAM NUMBER INTO BITS 0-5 OF THE SR
1605                                            ;       -SET SR BIT 11=1 TO LOOP ON SELECTED TEST
1606                                            ;       -SET SR BIT 11=0 AND BIT 8=0 TO LOOP THROUGH
1607                                            ;        SEQUENCE OF SELECTED TESTS.
1608                                            ;       -SET SR BIT 11=0 AND BIT 8=1 TO HALT AGAIN AFTER
1609                                            ;        EXECUTING TEST ONCE
1610                                            ;*********
1611
1612  001772  104006                    WAITF:  CHALT                   ;OR TTYCTL IF START WAS AT 210
1613  001774  012737  000006  000004            MOV     #6,MACHER       ;CLEAN UP
1614  002002  012706  000600                    MOV     #SPBOT,SP       ;SET UP STACK POINTER
1615  002006  017700  176702                    MOV     @SR,R0          ;GET CURRENT SW REG
1616  002012  042700  177700                    BIC     #177700,R0
1617  002016  020027  000037                    CMP     R0,#37          ;TEST IF PROG NO. IS I/O TEST
1618  002022  101403                            BLOS    1$              ;BRANCH IF EQ OR LT 37. AN ECHO OR PRINTER
1619  002024  005037  000644                    CLR     PRGID           ;I/O TEST, CLEAR PRGID
1620  002030  000403                            BR      2$
1621  002032  052737  000200  000644    1$:     BIS     #BIT7,PRGID     ;BYPASS SCOPING
1622  002040  000241                    2$:     CLC                     ;CLEAR C BIT
1623  002042  006100                            ROL     R0              ;GET PROGRAM ADDRESS OUT OF
1624  002044  016037  002614  000640            MOV     PRGTAB(R0),NXTST;PROGRAM ADDRESS TABLE
1625  002052  023727  000640  001772            CMP     NXTST,#WAITF    ;TEST IF LEGAL TEST NO.
1626  002060  001744                            BEQ     WAITF           ;BRANCH IF ILLEGAL
1627  002062  104024                            FORWD                   ;SET UP TEST PARAMETERS
1628  002064  000177  176606                    JMP     @CURTST         ;GO TO TEST
```

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

002070  022626                  TTY1:   POPSP2                  ;POP 2 FROM STACK
002072  105777  176514                  TSTB    @TKS            ;TEST IF ANY INPUT
002076  100013                          BPL     1$              ;BRANCH IF NOT
002100  017705  176510                  MOV     @TKB,R5         ;GET CHAR
002104  042705  177600                  BIC     #177600,R5      ;MASK BITS
002110  020527  000177                  CMP     R5,#177         ;CHECK IF RUBOUT
002114  001004                          BNE     1$              ;BRANCH IF NOT
002116  042737  004400  000634          BIC     #4400,CNTLSW    ;CLEAR LOOP BITS
002124  000413                          BR      TTY1B
002126  032737  004000  000634  1$:     BIT     #NITRSW,CNTLSW  ;CHECK IF LOOP ON TEST
002134  001401                          BEQ     2$              ;BRANCH IF NO LOOP ON TEST
002136  000002                          RTI                     ;LOOP ON TEST
002140  032737  000400  000634  2$:     BIT     #BIT8,CNTLSW    ;TEST IF LOOP ON SEQUENCE
002146  001402                          BEQ     TTY1B           ;BRANCH IF NO LOOP ON SEQUENCE
002150  000137  001626                  JMP     CHAINY          ;CHAIN TO NEXT TEST
002154  012737  177777  000710  TTY1B: MOV     #-1,INCHK       ;STOP INPUT CHECKING
002162  012700  000036                  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
002166  104010                          DELAY
002170  104007                          TYPEM
002172  015041                          MESG3                   ;TYPE MESSAGE
002174  005037  000710                  CLR     INCHK           ;ALLOW INPUT CHECKING AGAIN
002200  104020                  1$:     READ                    ;WAIT FOR INPUT
002202  023727  000700  000040          CMP     TEMPCH,#40      ;TEST IF CHAR IS A SPACE
002210  001773                          BEQ     1$              ;BRANCH IF YES
002212  012700  000036                  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
002216  104010                          DELAY
002220  104017                          PRNT                    ;READY?
002222  117777  176366  176370          MOVB    @TKB,@TPB       ;ECHO CHAR
002230  004737  002552                  JSR     PC,TESTC        ;CHECK IF CHAR IS OK
002234  000541                          BR      8$              ;NO, ERROR
002236  010005                          MOV     R0,R5           ;OK, PUT CHAR INTO R5
002240  006305                          ASL     R5              ;SHIFT INTO POSITION 5-3
002242  006305                          ASL     R5
002244  006305                          ASL     R5
002246  104020                  2$:     READ                    ;WAIT FOR NEXT CHAR
002250  023727  000700  000040          CMP     TEMPCH,#40      ;CHECK IF A SPACE
002256  001773                          BEQ     2$              ;BRANCH IF SPACE
002260  012700  000036                  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
002264  104010                          DELAY
002266  104017                          PRNT                    ;READY?
002270  117777  176320  176322          MOVB    @TKB,@TPB       ;ECHO CHAR
002276  004737  002552                  JSR     PC,TESTC        ;CHECK IF CHAR IS OK
002302  000516                          BR      8$              ;ERROR IN CHAR
```

```
 1685  002304  060005                          ADD     R0,R5           ;OK,R5 NOW = OCTAL TEST NO.
 1686  002306  104020                  3$:     READ                    ;WAIT FOR TERMINATION CHARACTER
 1687  002310  023727  000700  000040          CMP     TEMPCH,#40      ;CHECK IF SPACE
 1688  002316  001773                          BEQ     3$              ;BRANCH IF SPACE
 1689  002320  012700  000036                  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
 1690  002324  104010                          DELAY
 1691  002326  104017                          PRNT                    ;READY?
 1692  002330  117777  176260  176262          MOVB    @TKB,@TPB       ;ECHO CHAR
 1693  002336  012737  004001  000634          MOV     #4001,CNTLSW    ;SET BITS 11 & 0
 1694  002344  023727  000700  000114          CMP     TEMPCH,#114     ;NO, IS IT AN "L" ?
 1695  002352  001427                          BEQ     5$              ;BRANCH IF YES
 1696  002354  023727  000700  000154          CMP     TEMPCH,#154     ;CHECK LOWER CASE
 1697  002362  001423                          BEQ     5$
 1698  002364  023727  000700  000123          CMP     TEMPCH,#123     ;NO, IS IT AN "S"
 1699  002372  001414                          BEQ     4$              ;BRANCH IF YES
 1700  002374  023727  000700  000163          CMP     TEMPCH,#163     ;CHECK LOWER CASE
 1701  002402  001410                          BEQ     4$
 1702  002404  023727  000700  000056          CMP     TEMPCH,#56      ;NO, IS IT A "." ?
 1703  002412  001052                          BNE     8$              ;NO, ERROR
 1704  002414  012737  000001  000634          MOV     #1,CNTLSW       ;YES SET ONLY BIT 0 IN CONTROL WD
 1705  002422  000403                          BR      5$
 1706  002424  012737  000401  000634  4$:     MOV     #401,CNTLSW     ;SET BITS 8 & 0
 1707  002432  012737  000006  000004  5$:     MOV     #6,MACHER       ;CLEAN UP
 1708  002440  012706  000600                  MOV     #SPBOT,SP       ;INIT SP
 1709  002444  020527  000040                  CMP     R5,#40          ;IS THIS AN I/O TEST
 1710  002450  103033                          BHIS    8$              ;BRANCH IF YES
 1711  002452  020527  000030                  CMP     R5,#30          ;IS THIS AN OPTION TEST?
 1712  002456  103007                          BHIS    6$              ;SKIP IF YES
 1713  002460  020527  000020                  CMP     R5,#20          ;IS THIS AN ECHO TEST
 1714  002464  103404                          BLO     6$              ;BRANCH IF NOT
 1715  002466  012737  000001  000634          MOV     #1,CNTLSW       ;FORCE ECHO TEST TO A SINGLE RUN
 1716  002474  000402                          BR      7$              ;LEAVE THIS TERMINAL AS CONSOLE
 1717  002476  004737  003776          6$:     JSR     PC,CONIT        ;RESET CONSOLE TERMINAL ADDRESS
 1718  002502  052737  000200  000644  7$:     BIS     #BIT7,PRGID     ;BYPASS SCOPING
 1719  002510  000241                          CLC                     ;CLEAR C BIT
 1720  002512  006105                          ROL     R5
 1721  002514  016537  002614  000640          MOV     PRGTAB(R5),NXTST;ADDR OF TEST TO NXTST
 1722  002522  023727  000640  001772          CMP     NXTST,#WAITF    ;CHECK IF TEST EXISTS
 1723  002530  001403                          BEQ     8$              ;BRANCH IF NOT
 1724  002532  104024                          FORWD                   ;SET UP TEST PARAMETERS
 1725  002534  000177  176136                  JMP     @CURTST         ;GO TO TEST
 1726  002540  104017                  8$:     PRNT                    ;CHECK IF PRINTER IS READY
 1727  002542  112777  000077  176050          MOVB    #77,@TPB        ;SEND A "?"
 1728  002550  000601                          BR      TTYIB           ;TRY AGAIN
```

```
 1729                                   ;TESTC--CHECKS THAT THE INPUTTED CHARACTER IS BETWEEN 0 AND 7 INCLUSIVE
 1730
 1731  002552  023727  000700  000060  TESTC:  CMP     TEMPCH,#60      ;CHECK IF NUMERIC AND EQ OR GT 0
 1732  002560  103001                          BHIS    1$              ;BRANCH IF OK
 1733  002562  000207                          RTS     PC              ;ERROR RETURN
 1734  002564  023727  000700  000067  1$:     CMP     TEMPCH,#67      ;CHECK IF EQ OR LT 7
 1735  002572  101401                          BLOS    2$              ;BRANCH IF OK
 1736  002574  000207                          RTS     PC              ;ERROR RETURN
 1737  002576  062716  000002          2$:     ADD     #2,@SP          ;SET UP RETURN ADDRESS
 1738  002602  013700  000700                  MOV     TEMPCH,R0       ;GET CHAR
 1739  002606  042700  177770                  BIC     #177770,R0      ;SAVE ONLY THE DIGIT
 1740  002612  000207                          RTS     PC              ;NORMAL RETURN
```

```
 1741  002614  007374
 1742  002616  007450
 1743  002620  007572
 1744  002622  010166
 1745  002624  010306
 1746  002626  010464
 1747  002630  010670
 1748  002632  011056
 1749  002634  011270
 1750  002636  011426
 1751  002640  011460
 1752  002642  001772
 1753  002644  001772
 1754  002646  001772
 1755  002650  001772
 1756  002652  011550
 1757  002654  012120
 1758  002656  012170
 1759  002660  012226
 1760  002662  012500
 1761  002664  013022
 1762  002666  013570
 1763  002670  001772
 1764  002672  001772
 1765  002674  001772
 1766  002676  001772
 1767  002700  001772
 1768  002702  001772
 1769  002704  001772
 1770  002706  001772
 1771  002710  001772
 1772  002712  001772
 1773  002714  005354
 1774  002716  005406
 1775  002720  005440
 1776  002722  005472
 1777  002724  005524
 1778  002726  005614
 1779  002730  005672
 1780  002732  005762
 1781  002734  006032
 1782  002736  006070
 1783  002740  006130
 1784  002742  006204
 1785  002744  006264
 1786  002746  006350
 1787  002750  006450
 1788  002752  006516
 1789  002754  006566
 1790  002756  006660
 1791  002760  006764
 1792  002762  007072
 1793  002764  007204
 1794  002766  001772
 1795  002770  001772
 1796  002772  001772

PRGTAB: PT0                     ;DATA PATH TEST
        PT1                     ;PRINTER CHARACTER TEST
        PT2                     ;NON-PRINTING CHARACTER TEST
        PT3                     ;CARRIAGE RETURN TEST
        PT4                     ;MULTIPLE LINE FEED TEST
        PT5                     ;SINGLE LINE FEED TEST
        PT6                     ;BACKSPACE TEST
        PT7                     ;OVERPRINT TEST
        PT10                    ;PRINTING FREQUENCY SWEEP TEST
        PT11                    ;RIBBON FEED TEST
        PT12                    ;PRINTER BELL TEST
        WAITF                   ;SPARE
        WAITF                   ;SPARE
        WAITF                   ;SPARE
        WAITF                   ;SPARE
        PT17                    ;LIFE TEST
        E020                    ;CHARACTER ECHO TEST
        E021                    ;LINE ECHO TEST, FAST RATE
        E022                    ;LINE ECHO TEST, SLOW RATE
        E023                    ;CHARACTER/CODE ECHO TEST
        E024                    ;SELECTIVE PATTERN ECHO TEST
        E025                    ;BELL ECHO TEST
        WAITF                   ;SPARE
        WAITF                   ;SPARE
        WAITF           ;SPARE
        WAITF           ;SPARE
        WAITF           ;SPARE
        WAITF           ;PRARE
        WAITF           ;SPARE
        WAITF           ;SPARE
        WAITF                   ;SPARE
        WAITF                   ;SPARE
        AT0                     ;I/O TEST NO. 40
        AT1                     ;I/O TEST NO. 41
        AT2                     ;I/O TEST NO. 42
        AT3                     ;I/O TEST NO. 43
        AT4                     ;I/O TEST NO. 44
        AT5                     ;I/O TEST NO. 45
        AT6                     ;I/O TEST NO. 46
        AT7                     ;I/O TEST NO. 47
        AT10                    ;I/O TEST NO. 50
        AT11                    ;I/O TEST NO. 51
        AT12                    ;I/O TEST NO. 52
        AT13                    ;I/O TEST NO. 53
        AT14                    ;I/O TEST NO. 54
        AT15                    ;I/O TEST NO. 55
        AT16                    ;I/O TEST NO. 56
        AT17                    ;I/O TEST NO. 57
        AT20                    ;I/O TEST NO. 60
        AT21                    ;I/O TEST NO. 61
        AT22                    ;I/O TEST NO. 62
        AT23                    ;I/O TEST NO. 63
        AT24                    ;LSI TEST NO. 64
        WAITF                   ;SPARE
        WAITF                   ;SPARE
        WAITF                   ;SPARE
```

```
1797  002774  001772                    WAITF                   ;SPARE
1798  002776  001772                    WAITF                   ;SPARE
1799  003000  001772                    WAITF                   ;SPARE
1800  003002  001772                    WAITF                   ;SPARE
1801  003004  001772                    WAITF                   ;SPARE
1802  003006  001772                    WAITF                   ;SPARE
1803
1804                                    ;**********
1805                                    ;EMTINT ------------SERVICE ROUTINE FOR TRAPS THROUGH
1806                                    ;                   LOCATION 30.
1807                                    ;**********
1808
1809  003010  011646                    EMTINT: MOV     @SP,-(SP)       ;PUSH STACKED PC TO GET A WORK COPY.   (Q)
1810  003012  162716  000002                    SUB     #2,@SP          ;SUB 2 TO POINT TO CALLING TRAP INSTR.
1811  003016  017616  000000                    MOV     @(SP),@SP       ;PLACE TRAP INSTR INTO THIS STACK WORK AREA.
1812  003022  121627  000035                    CMPB    @SP,#35         ;EXAMINE ITS RIGHT SIDE.  (Q)
1813  003026  101402                            BLOS    2$              ;BRANCH IF WITHIN RANGE OF ESTABLISHED TABLE.
1814  003030  000000                    1$:     HALT                    ;ELSE HALT.
1815  003032  000776                            BR      1$
1816  003034  006116                    2$:     ROL     @SP             ;MULT INSTR BY 2 TO GET WORD DISPLACEMENT.
1817  003036  042716  177001                    BIC     #177001,@SP     ;STRIP OFF OP CODE AND LS BIT.
1818  003042  062716  003064                    ADD     #EMTTAB,@SP     ;ADD IN STARTING ADDRESS OF TABLE.
1819  003046  017616  000000                    MOV     @(SP),@SP       ;FROM TABLE GET OUT DESIRED POINTER.
1820  003052  005046                            CLR     -(SP)           ;PUSH A ZERO PSW.
1821  003054  012746  003062                    MOV     #3$,-(SP)       ;PUSH A PC = TO #3$ OF THIS ROUTINE.
1822  003060  000002                            RTI                     ;DO RTI (POP-POP) TO ESTABLISH THE ZERO PSW.
1823  003062  000136                    3$:     JMP     @(SP)+          ;JMP TO ROUTINE LEAVING STACK AS FOUND.
1824
1825  003064  003164                    EMTTAB: TYP                     ;MESSAGE OUTPUT ROUTINE
1826  003066  003406                            ERR                     ;I/O TEST ERROR ROUTINE
1827  003070  003434                            EHLT                    ;UNCONDITIONAL HALT
1828  003072  003444                            STLSRV                  ;KEYBOARD VECTOR/PRIORITY SETUP
1829  003074  003474                            STLSPV                  ;PRINTER VECTOR/PRIORITY SETUP
1830  003076  001500                            CHAINN                  ;COMMON TEST EXIT
1831  003100  000736                            CHLT                    ;SR BIT 15 HALT
1832  003102  003252                            TYPM                    ;MESSAGE OUTPUT ROUTINE, MULTI DEVICES
1833  003104  003524                            DLY                     ;DELAY ROUTINE
1834  003106  002070                            TTY1                    ;CONSOLE TERMINAL CONTROL
1835  003110  003302                            $CRLF                   ;CARRIAGE RETURN-LINE FEED TO ALL DL11'S
1836  003112  003230                            $SCRLF                  ;CARRIAGE RETURN-LINE FEED TO CONSOLE
1837  003114  003304                            $LF                     ;LINE FEED ONLY (TO ALL)
1838  003116  004412                            $PRTC                   ;PRINT CHAR
1839  003120  003324                            $PRHDR                  ;PRINT TEST HEADER
1840  003122  004402                            $PRNT                   ;PRINTER READY
1841  003124  004200                            $READ                   ;READ CHAR
1842  003126  003726                            $AREAD                  ;I/O TEST READ ROUTINE
1843  003130  003314                            $CR                     ;CARRIAGE RETURN ONLY (TO ALL)
1844  003132  004074                            $BTASC                  ;BINARY TO ASCII CONVERSION
1845  003134  003650                            $FORWD                  ;FORWARD ROUTINE ( BETWEEN TESTS )
1846  003136  004272                            $READC                  ;READ CONSOLE KYBD ONLY
1847  003140  003160                            SPARET                  ;SPARE EMT
1848  003142  003160                            SPARET                  ;SPARE EMT
1849  003144  003160                            SPARET                  ;SPARE EMT
1850  003146  003160                            SPARET                  ;SPARE EMT
1851  003150  003160                            SPARET                  ;SPARE EMT
1852  003152  003160                            SPARET                  ;SPARE EMT
```

```
 1853  003154  003160                    SPARET                  ;SPARE EMT
 1854  003156  003160                    SPARET                  ;SPARE EMT
 1855  003160  000000                SPARET: HALT                ;HALT IF TRAP TO UNDEFINED
 1856  003162  000776                    BR      SPARET          ;EMT IS ATTEMPTED.
 1857
 1858
 1859
 1860
 1861                                    .SBTTL  COMMON ROUTINES USED BY LA36 TESTS
 1862
 1863                                ;**********
 1864                                ;
 1865                                ;THIS SECTION CONTAINS MOST ROUTINES CALLED BY
 1866                                ;THE VARIOUS TESTS EITHER BY TRAPPING THROUGH LOCATION
 1867                                ;30 OR BY SUBROUTINE CALLS (JSR PC,***)
 1868                                ;
 1869                                ;**********
 1870                                ;
 1871                                ;TYPE-- A COMMON ROUTINE USED TO TYPE MESSAGES ON THE
 1872                                ;       CONSOLE TERMINAL ONLY. THE NULL CHARACTER TERMINATES
 1873                                ;       THE MESSAGE. CALLED THROUGH AN EMT TRAP.
 1874                                ;       CALLING SEQUENCE
 1875                                ;       TYPE
 1876                                ;       MESG    ;ADDRESS OF MESSAGE
 1877                                ;
 1878                                ;**********
 1879  003164  010046                TYP:    MOV     R0,-(SP)        ;SAVE R0
 1880  003166  016601  000002            MOV     2(SP),R1        ;GET POINTER TO ADDR. OF MESG.
 1881  003172  062766  000002  000002    ADD     #2,2(SP)
 1882  003200  011101                    MOV     (R1),R1         ;ADDR. OF MESG TO R1
 1883  003202  112100                1$:     MOVB    (R1)+,R0        ;GET CHAR
 1884  003204  100403                    BMI     2$              ;BRANCH IF WANT AUTO CR-LF
 1885  003206  001004                    BNE     3$              ;PRINT CHAR IF NOT NULL
 1886  003210  012600                    MOV     (SP)+,R0        ;RESTORE R0
 1887  003212  000002                    RTI                     ;EXIT IF NULL CHAR
 1888  003214  104013                2$:     SCRLF                   ;YES, SEND CR-LF
 1889  003216  000771                    BR      1$              ;GET NEXT CHAR
 1890  003220  104017                3$:     PRNT                    ;PRINTER READY?
 1891  003222  110077  175372            MOVB    R0,@TPB         ;LOAD PRINTER BUFFER WITH CHAR
 1892  003226  000765                    BR      1$              ;GO GET NEXT CHAR
 1893
 1894  003230  104017                $SCRLF: PRNT                    ;PRINTER READY?
 1895  003232  112777  000015  175360    MOVB    #15,@TPB        ;SEND CR
 1896  003240  104017                    PRNT                    ;PRINTER READY?
 1897  003242  112777  000012  175350    MOVB    #12,@TPB        ;SEND LF
 1898  003250  000002                    RTI                     ;RETURN TO CALLER
```

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

003252  011601          TYPM:   MOV     (SP),R1         ;GET POINTER TO ADDR OF MESG
003254  062716  000002          ADD     #2,@SP
003260  011101                  MOV     (R*),R1         ;ADDR OF MESG TO R1
003262  112100          1$:     MOVB    (R1)+,R0        ;GET CHAR
003264  100402                  BMI     2$              ;BRANCH IF WANT AUTO CR-LF
003266  001003                  BNE     3$              ;CONTINUE IF NOT NULL
003270  000002                  RTI                     ;RETURN
003272  104012          2$:     CRLF                    ;YES, SEND CR-LF
003274  000772                  BR      1$              ;NEXT CHAR
003276  104015          3$:     PRINTC                  ;PRINT CHAR
003300  000770                  BR      1$              ;GO GET NEXT CHAR.

003302  104022          $CRLF:  CR                      ;SEND CR
003304  012700  000012  $LF:    MOV     #12,R0          ;SET LF CHAR
003310  104015                  PRINTC                  ;SEND IT
003312  000002                  RTI                     ;RETURN TO CALLER

003314  012700  000015  $CR:    MOV     #15,R0          ;SET CR CHAR
003320  104015                  PRINTC                  ;SEND IT
003322  000002                  RTI                     ;RETURN

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

003324  012700  000000  $PRHDR: MOV     #0,R0           ;TRANSMIT
003330  104015                  PRINTC                  ;NUL CODE.
003332  104007                  TYPEM                   ;PRINT MESSAGE
003334  014613                  HDRMSG
003336  013700  000636          MOV     RTNNO,R0        ;GET TEST NUMBER
003342  006200                  ASR     R0              ;GET FIRST DIGIT
003344  006200                  ASR     R0
003346  006200                  ASR     R0
003350  042700  177770          BIC     #177770,R0      ;MASK FIRST DIGIT
003354  062700  000060          ADD     #60,R0          ;MAKE ASCII
003360  104015                  PRINTC                  ;PRINT DIGIT
003362  013700  000636          MOV     RTNNO,R0        ;GET TEST NUMBER AGAIN
003366  042700  177770          BIC     #177770,R0      ;MASK LAST DIGIT
003372  062700  000060          ADD     #60,R0          ;MAKE ASCII
003376  104015                  PRINTC                  ;PRINT DIGIT
003400  104012                  CRLF                    ;CR-LF
003402  104014                  LF                      ;BLANK LINE
003404  000002                  RTI                     ;RETURN
```

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

003406  032777  040000  175300 ERR:    BIT     #SCOPSW,@SR     ;CHECK SCOPE SWITCH
003414  001404                        BEQ     1$              ;BRANCH IF NO SCOPE
003416  005737  000644              TST     PRGID           ;SCOPING WANTED, FIRST ERROR?
003422  100001                      BPL     1$              ;BRANCH AND HALT ON FIRST ERROR
003424  000002                      RTI                     ;SCOPE EXIT
003426  052737  100000  000644 1$:    BIS     #BIT15,PRGID    ;SET ERROR INDICATOR
003434  011600                  EHLT:   MOV     @SP,R0
003436  005740                      TST     -(R0)           ;ADDRESS OF CALL INTO R0
003440  000000                      HALT
003442  000002                  ERRHLT: RTI                     ;RETURN TO TEST FOLLOWING CALL

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

003444  017637  000000  003464 STLSRV: MOV     @(SP),STPRA+2   ;SET RETURN ADR AND VECTOR
003452  062716  000002              ADD     #2,@SP
003456  013701  000622              MOV     TKVTR,R1
003462  012721  000000          STPRA:  MOV     #0,(R1)+
003466  013721  000624              MOV     TKLVL,(R1)+
003472  000002                      RTI

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

003474  017637  000000  003514 STLSPV: MOV     @(SP),STPPA+2   ;SET RETURN ADR AND VECTOR
003502  062716  000002              ADD     #2,@SP
003506  013701  000626              MOV     TPVTR,R1
003512  012721  000000          STPPA:  MOV     #0,(R1)+
003516  013721  000630              MOV     TPLVL,(R1)+
003522  000002                      RTI                     ;RETURN TO CALLER
```

```
                        ;**********
                        ;
                        ;DELAY--A COMMON ROUTINE TO DELAY PROCESSING
                        ;       A GIVEN NUMBER OF MSEC.
                        ;       CALLING SEQUENCE;
                        ;       MOV   #5,R0   ;R0 CONTAINS THE NUMBER OF MSEC DELAY DESIRED
                        ;       DELAY
                        ;
                        ;       THE DELAY IS EFFECTED BY THE EXECUTION OF THE LOOP;
                        ;          1$:  DEC  R1
                        ;               BNE  1$
                        ;
                        ;       SINCE THE EXECUTION TIMES OF THE PDP11 LINE DOES VARY FROM
                        ;       MACHINE TO MACHINE, THE VALUE AT SYMBOLIC LOCATION
                        ;       "TIMER" MUST BE CHANGED TO THE APPROPRIATE VALUE AS SHOWN BELOW
                        ;       BEFORE STARTING THE DIAGNOSTIC. "TIMER" IS INITIALIZED
                        ;       FOR AN 11/05,11/10(=251).
                        ;
                        ;MACHINE        05&10   35&40   15&20   LSI&03          11/45 & 11/70
                        ;                                                BIPOLAR        MOS       CORE
                        ;
                        ;LOOP: DEC R1   3.4     .99     2.3               .30           .51     .90
                        ;      BNE LOOP 2.5     1.76    2.6               .60           .98    1.13
                        ;  TIME=         5.9USEC 2.75    4.9     7.7     .90USEC        1.49USEC        2.03USEC
                        ; SET TIMER     251     554     314     202     2127           1237    755
                        ;
                        ;XXXXXXXXXX
                        ;
003524  010146          DLY:    MOV     R1,-(SP)        ;SAVE R1
003526  013701  000672  1$:     MOV     TIMER,R1        ;MOV 1 MSEC LOOP CNT TO R1
003532  005301          2$:     DEC     R1              ;DECREMENT COUNT
003534  001376                  BNE     2$              ;BRANCH IF NOT ZERO
003536  005300                  DEC     R0              ;DEC NO. OF MSEC DELAY
003540  001372                  BNE     1$              ;DELAY AGAIN IF NOT ZERO
003542  012601                  MOV     (SP)+,R1        ;ALL DONE RESTORE R1
003544  000002                  RTI
```

```
2043                    ;**********
2044                    ;
2045                    ;PFAIL--POWER FAIL ROUTINE
2046                    ;       SAVE ALL REGISTERS AND SET RESTART ADDRESS
2047                    ;       INTO LOCATION 24
2048                    ;
2049                    ;RESTART--POWER FAIL RECOVERY
```

```
2050                    ;       RESTORE ALL REGISTERS AND GO TO START
```

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

 003546  010046                         PFAIL:  MOV     R0,-(SP)
 003550  010146                                 MOV     R1,-(SP)
 003552  010246                                 MOV     R2,-(SP)
 003554  010346                                 MOV     R3,-(SP)
 003556  010446                                 MOV     R4,-(SP)
 003560  010546                                 MOV     R5,-(SP)
 003562  013746  000024                         MOV     24,-(SP)
 003566  010637  003602                         MOV     SP,SAVR6        ;SAVE STACK POSITION
 003572  012737  003604  000024                 MOV     #RESTRT,24      ;STORE RESTART ADDRESS
 003600  000000                                 HALT
 003602  000000                         SAVR6:  .WORD   0
 003604  104007                         RESTRT: TYPEM
 003606  003640                                 1$
 003610  013706  003602                         MOV     SAVR6,SP        ;RESTORE STACK POINTER
 003614  012637  000024                         MOV     (SP)+,24        ;RESTORE PFAIL ADDRESS
 003620  012605                                 MOV     (SP)+,R5
 003622  012604                                 MOV     (SP)+,R4
 003624  012603                                 MOV     (SP)+,R3
 003626  012602                                 MOV     (SP)+,R2
 003630  012601                                 MOV     (SP)+,R1
 003632  012600                                 MOV     (SP)+,R0
 003634  000137  001026                         JMP     START

 003640  050200  053517  051105         1$:     .ASCIZ  <ACRLF>/POWER/<ACRLF>
 003646  000200
                                                .EVEN
```

```
2080                                    ;**********
2081                                    ;
2082                                    ;FORWARD--THIS ROUTINE TRANSFERS THE 2 OR 4 ARGUMENTS
2083                                    ;       FROM THE TEST ROUTINE. THEY ARE:
2084                                    ;
2085                                    ;       1- ROUTINE NUMBER
2086                                    ;       2- ADDRESS OF NEXT TEST
2087                                    ;       3- ITERATION COUNT (I/O TESTS ONLY)
2088                                    ;       4- SCOPE ENTRY ADDRESS (I/O TESTS ONLY)
2089                                    ;
2090                                    ;**********
2091
2092  003650  013705  000640           $FORWD: MOV     NXTST,R5        ;ADDR OF NEXT TEST TO R5
2093  003654  012537  000636                   MOV     (R5)+,RTNNO     ;GET NUMBER OF NEXT TEST
2094  003660  012537  000640                   MOV     (R5)+,NXTST     ;GET ADDR OF FOLLOWING TEST
2095  003664  105737  000644                   TSTB    PRGID           ;CHECK IF I/O TEST
2096  003670  100407                           BMI     FORWDB          ;SKIP THE FETCH OF ITER CNT AND SCOPE
2097  003672  012537  000660                   MOV     (R5)+,ICTR      ;GET ITERATION COUNT
2098  003676  012537  000642                   MOV     (R5)+,SCOPTR    ;GET SCOPE ENTRY POINT
2099  003702  010537  000676           FORWDA: MOV     R5,CURTST       ;ENTRY POINT TO TEST IN CUR TST
2100  003706  000002                           RTI                     ;EXIT
2101  003710  012737  177777  000642 FORWDB: MOV     #-1,SCOPTR      ;FORCE NO SCOPE
2102  003716  012737  000001  000660         MOV     #1,ICTR         ;FORCE INTERATION COUNT OF 1
2103  003724  000766                         BR      FORWDA
```

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

003726  012737  000600  000664  $AREAD: MOV     #600,BRCTR      ;SET UP 600 MSEC DELAY
003734  052777  000004  174654          BIS     #4,@TPS         ;SET MAINTENANCE BIT
003742  005077  174652                  CLR     @TPB            ;LOAD PRINTER BUFFER
003746  105777  174640          1$:     TSTB    @TKS            ;CHECK DONE BIT
003752  100410                          BMI     2$              ;BRANCH IF DONE
003754  012700  000001                  MOV     #1,R0           ;ONE TO R0
003760  104010                          DELAY                   ;DELAY 1 MSEC.
003762  005337  000664                  DEC     BRCTR           ;600 MSEC OVER
003766  001367                          BNE     1$              ;BRANCH IF NO
003770  104002                          EHALT
003772  000755                          BR      $AREAD          ;TRY AGAIN
003774  000002                  2$:     RTI                     ;RETURN TO TEST

                                        ;**********
                                        ;
                                        ;CONIT--THIS ROUTINE SETS UP THE DEVICE ADDRESSES
                                        ;       AND INTERRUPT VECTORS FOR THE CONSOLE
                                        ;       TERMINAL.
                                        ;
                                        ;**********

003776  013700  000602          CONIT:  MOV     CONADD,R0       ;CONSOLE KEYBOARD STATUS ADDR TO R0
004002  010037  000612          CONSET: MOV     R0,TKS          ;KEYBOARD STATUS ADDRESS (777560) TO TKS
004006  005720                          TST     (R0)+           ;INCREMENT R0 BY TWO
004010  010037  000614                  MOV     R0,TKB          ;KEYBOARD DATA ADDR (777562) TO TKB
004014  005720                          TST     (R0)+           ;INCREMENT R0 BY TWO
004016  013737  000616  004070          MOV     TPS,TPSS        ;SAVE TPS OF LAST TERMINAL
004024  010037  000616                  MOV     R0,TPS          ;PRINTER STATUS ADDR(777564) TO TPS
004030  005720                          TST     (R0)+           ;INCREMENT R0 BY TWO
004032  013737  000620  004072          MOV     TPB,TPBS        ;SAVE TPB OF LAST TERMINAL
004040  010037  000620                  MOV     R0,TPB          ;PRINTER DATA ADDR (777566) TO TPB
004044  013737  000604  000622          MOV     CONVEC,TKVTR    ;KEYBOARD INTERRUPT VECTOR (60) TO TKVTR
004052  013737  000604  000626          MOV     CONVEC,TPVTR
004060  062737  000004  000626          ADD     #4,TPVTR        ;PRINTER INTERRUPT VECTOR (64) TO TPVTR
004066  000207                          RTS     PC

004070  000000          TPSS:   .WORD   0               ;LAST TERM STATUS REG ADR
004072  000000          TPBS:   .WORD   0               ;LAST TERM BUFFER REG ADR
```

```
                                ;**********
                                ;
                                ;BINARY TO ASCII CONVERSION (1 TO 5 ASCII CHARACTERS)
                                ;CALLING SEQUENCE
                                ;       MOV     ADDRESS OF LOC. TO STORE FIRST ASCII CHAR. INTO R0
                                ;       MOV     BINARY NUMBER TO BE CONVERTED INTO R1
                                ;       MOV     NUMBER TO BE CONVERTED AS A POWER OF TEN INTO R2
                                ;       BTOASC
                                ;
                                ;**********

004074  010237  004160          $BTASC: MOV     R2,CNVCTR       ;SAVE TEN POWER
004100  006302                          ASL     R2              ;R2*2
004102  062702  004166                  ADD     #ADTENP,R2      ;CALCULATE ADDRESS OF
                                                                ;STARTING TEN POWER
004106  014237  004164          1$:     MOV     -(R2),TENPWR    ;POWER OF TEN VALUE TO TEN PWR
004112  005037  004162                  CLR     DIGIT           ;CLEAR CURRENT DIGIT
004116  163701  004164          2$:     SUB     TENPWR,R1       ;SUBTRACT TEN POWER FROM BINARY VALUE
004122  103403                          BCS     3$              ;BRANCH IF END
004124  005237  004162                  INC     DIGIT
004130  000772                          BR      2$
004132  063701  004164          3$:     ADD     TENPWR,R1       ;RESTORE SUBTRACTED VALUE
004136  062737  000060  004162          ADD     #60,DIGIT       ;CONVERT (DIGIT) TO ASCII
004144  113720  004162                  MOVB    DIGIT,(R0)+     ;PUT ASCII CHAR INTO USER BUFFER
004150  005337  004160                  DEC     CNVCTR          ;FINISHED ALL CHARS. CALLED FOR
004154  001354                          BNE     1$              ;BRANCH IF NOT FINISHED
004156  000002                          RTI                     ;YES, EXIT
004160  000000          CNVCTR: .WORD   0               ;CONVERSION CHARACTER COUNT
004162  000000          DIGIT:  .WORD   0               ;CONVERTED CHARACTER
004164  000000          TENPWR: .WORD   0               ;CURRENT TEN POWER
004166  000001  000012  000144  ADTENP: .WORD   1.,10.,100.,1000.,10000.
004174  001750  023420
```

```
                                        ;XXXXXXXXXX
                                        ;
                                        ;READ-- A COMMON ROUTINE WHICH CHECKS THE KEYBOARD
                                        ;       DONE FLAG & SETS A FLAG INDICATING CHAR PARITY
                                        ;
                                        ;XXXXXXXXXX

004200  004737  003776          $READ:  JSR     PC,CONIT        ;RESET CONSOLE ADR AND VECTORS
004204  005737  000656                  TST     DLCNT           ;CHECK IF MULTI DL11'S AVAILABLE
004210  001430                          BEQ     $READC          ;NONE, WAIT FOR CONSOLE INPUT
004212  013737  000656  000666  1$:     MOV     DLCNT,COUNT3    ;SET DL11 COUNT
004220  013737  000632  000670          MOV     FSTDL,XCSR      ;ADDRESS OF FIRST DL'1 INTO XCSR
004226  105777  174436          2$:     TSTB    @XCSR           ;TEST IF ANY INPUT
004232  100005                          BPL     3$              ;CONTINUE IF NO INPUT
004234  013700  000670                  MOV     XCSR,R0         ;SET THIS DL11 AS CONSOLE
004240  004737  004002                  JSR     PC,CONSET
004244  000415                          BR      READ1           ;READ CHAR AND RETURN
004246  005337  000666          3$:     DEC     COUNT3          ;DECREMENT DL11 COUNT
004252  001404                          BEQ     4$              ;TEST CONSOLE WHEN DONE DL11'S
004254  062737  000010  000670          ADD     #10,XCSR        ;NEXT DL11 ADDRESS
004262  000761                          BR      2$              ;CONTINUE
004264  105777  174322          4$:     TSTB    @TKS            ;CHECK CONSOLE
004270  100350                          BPL     1$              ;WAIT, NO INPUT
004272  105777  174314          $READC: TSTB    @TKS            ;CHECK KEYBOARD DONE FLAG
004276  100375                          BPL     $READC          ;BRANCH IF NOT SET
004300  117737  174310  000700  READ1:  MOVB    @TKB,TEMPCH     ;SAVE CHARACTER
004306  113737  000700  000704          MOVB    TEMPCH,PCHAR    ;SAVE CODE WITH PARITY BIT
004314  042737  177400  000704          BIC     #177400,PCHAR   ;MASK UNWANTED BITS
004322  113737  000700  000703          MOVB    TEMPCH,PARITY+1 ;SAVE CHAR WITH PARITY BIT
004330  042737  177600  000700          BIC     #177600,TEMPCH  ;MAKE IT 7 BIT ASCII
004336  023727  000700  000004          CMP     TEMPCH,#4       ;DISREGARD EOT
004344  001715                          BEQ     $READ
004346  012700  000011                  MOV     #11,R0          ;SET SHIFT COUNT
004352  042737  000377  000702          BIC     #377,PARITY     ;CLEAR PARITY FLAG
004360  005300                  1$:     DEC     R0              ;DECREMENT SHIFT COUNT
004362  001406                          BEQ     2$              ;EXIT IF DONE
004364  106337  000703                  ASLB    PARITY+1        ;SHIFT CODE
004370  103373                          BCC     1$              ;CONTINUE IF BIT WAS ZERO
004372  105137  000702                  COMB    PARITY          ;CHANGE PARITY FLAG IF BIT WAS ONE
004376  000770                          BR      1$              ;CONTINUE
004400  000002                  2$:     RTI                     ;SET, RET. TO CALLER

                                        ;XXXXXXXXXX
                                        ;
                                        ;PRINT-- A COMMON ROUTINE TO CHECK THE PRINTER READY FLAG
                                        ;
                                        ;XXXXXXXXXX

004402  105777  174210          $PRNT:  TSTB    @TPS            ;CHECK PRINTER READY FLAG
004406  100375                          BPL     $PRNT           ;BRANCH IF NOT SET
004410  000002                          RTI                     ;SET, RETURN
```

```
                        ;*********
                        ;
                        ;PRINTC--SENDS A CHARACTER AT A TIME FIRST TO THE
                        ;       CONSOLE  DL11 THEN TO ALL MULTIPLE DL11S IF
                        ;       SR BIT 13 IS = 0. IF THE REFERENCED PRINTER
                        ;       READY BIT IS NOT SET, THE CHARACTER WILL NOT BE
                        ;       SENT TO THAT PRINTER. ENTER WITH CHARACTER IN R0.
                        ;       CALL:   PRINTC
                        ;
                        ;*******

004412  013737  000602  000712  $PRTC:  MOV     CONADD,TEMP     ;SET CONSOLE ADR
004420  062737  000004  000712          ADD     #4,TEMP
004426  105777  174260          1$:     TSTB    @TEMP
004432  100375                          BPL     1$              ;WAIT FOR CONSOLE READY
004434  062737  000002  000712          ADD     #2,TEMP         ;SET ADR
004442  010077  174244                  MOV     R0,@TEMP        ;LOAD CONSOLE PRINTER BUFFER
004446  032777  020000  174240          BIT     #BIT13,@SR      ;CHECK SW 13
004454  001003                          BNE     2$              ;SEND ALL TERMS IF SW13 DOWN
004456  005737  000656                  TST     DLCNT           ;CHECK IF MULTIPLE DL11'S
004462  001002                          BNE     3$              ;CHECK FOR INPUT IF THERE
004464  000137  005122          2$:     JMP     18$
004470  013737  000656  000666  3$:     MOV     DLCNT,COUNT3    ;PUT NO. DL11'S INTO COUNT3
004476  013737  000632  000670          MOV     FSTDL,XCSR      ;ADDR OF FIRST DL INTO XCSR
004504  005737  000710          4$:     TST     INCHK           ;CHECK FOR INPUT?
004510  001140                          BNE     13$
004512  023727  000636  000020          CMP     RTNNO,#20       ;PRINTING TEST?
004520  002004                          BGE     5$              ;BRANCH IF NOT
004522  022737  104011  001772          CMP     #TTYCTL,WAITF   ;KEYBOARD CONTROL?
004530  001130                          BNE     13$             ;SKIP INPUT CHECK IF NOT
004532  105777  174132          5$:     TSTB    @XCSR           ;TEST IF ANY INPUT
004536  100125                          BPL     13$             ;CONTINUE IF NO INPUT
004540  062737  000002  000670          ADD     #2,XCSR         ;SET BUFFER ADDRESS
004546  017737  174116  000700          MOV     @XCSR,TEMPCH
004554  042737  177600  000700          BIC     #177600,TEMPCH
004562  023727  000700  000003          CMP     TEMPCH,#3       ;CHECK IF CONTROL-C
004570  001006                          BNE     6$              ;CONTINUE IF NOT
004572  023727  000636  000024          CMP     RTNNO,#24       ;CHECK IF TEST 24
004600  001002                          BNE     6$              ;CONTINUE IF NOT CONTROL-C
004602  000137  005226                  JMP     20$
004606  023727  000700  000177  6$:     CMP     TEMPCH,#177     ;CHECK IF RUBOUT
004614  001427                          BEQ     9$              ;YES, CHECK TEST NUMBER
004616  023727  000636  000017          CMP     RTNNO,#17       ;TEST 17?
004624  001003                          BNE     7$              ;BRANCH IF NOT
004626  013703  000700                  MOV     TEMPCH,R3       ;SAVE CHARACTER
004632  000461                          BR      12$             ;CONTINUE
004634  023727  000636  000021  7$:     CMP     RTNNO,#21       ;TEST 21?
004642  001004                          BNE     8$              ;BRANCH IF NOT
004644  013737  000700  000662          MOV     TEMPCH,REPT     ;SAVE CHARACTER
004652  000451                          BR      12$             ;CONTINUE
004654  023727  000636  000022  8$:     CMP     RTNNO,#22       ;TEST 22?
004662  001056                          BNE     14$             ;CONTINUE IF NOT
004664  013737  000700  000662          MOV     TEMPCH,REPT     ;SAVE CHARACTER
004672  000441                          BR      12$             ;CONTINUE
004674  023727  000636  000021  9$:     CMP     RTNNO,#21       ;CHECK IF TEST 21
004702  001011                          BNE     10$             ;NO, CHECK IF TEST 22
```

```
2293  004704  022626                        POPSP2                  ;ASJUST STACK
2294  004706  012700  000036                MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
2295  004712  104010                        DELAY
2296  004714  104007                        TYPEM                   ;YES, TEST 21
2297  004716  014772                        ECOEND                  ;PRINT TERMINATION MESSAGE
2298  004720  104005                        CHAIN                   ;CHAIN TO NEXT TEST
2299  004722  000137  012176                JMP     E021A           ;REPEAT TEST IF LOOP ON TEST SW SET
2300  004726  023727  000636  000022  10$:  CMP     RTNNO,#22       ;CHECK IF TEST 22
2301  004734  001011                        BNE     11$             ;NO, CHECK IF TEST 24
2302  004736  022626                        POPSP2                  ;ADJUST STACK
2303  004740  012700  000036                MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
2304  004744  104010                        DELAY
2305  004746  104007                        TYPEM                   ;YES, PRINT TERMINATION MESSAGE
2306  004750  014772                        ECOEND
2307  004752  104005                        CHAIN                   ;CHAIN TO NEXT TEST
2308  004754  000137  012234                JMP     E022A           ;REPEAT TEST IF LOOP ON TEST SW SET
2309  004760  023727  000636  000024  11$:  CMP     RTNNO,#24       ;TEST 24?
2310  004766  001133                        BNE     22$             ;WAIT FOR NEXT TEST IF NOT TEST 24
2311  004770  022626                        POPSP2                  ;RESET STACK
2312  004772  000137  013156                JMP     TERM            ;TERMINATE TEST
2313  004776  012700  000036          12$:  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
2314  005002  104010                        DELAY
2315  005004  013700  000700                MOV     TEMPCH,R0       ;SET NEW CHARACTER
2316  005010  000403                        BR      14$             ;CONTINUE
2317  005012  062737  000002  000670  13$:  ADD     #2,XCSR         ;SET STATUS ADDRESS IN XCSR
2318  005020  062737  000002  000670  14$:  ADD     #2,XCSR
2319  005026  013737  000602  000712        MOV     CONADD,TEMP     ;CHECK IF CONSOLE TERMINAL
2320  005034  062737  000004  000712        ADD     #4,TEMP         ;IS THIS DL
2321  005042  023737  000712  000670        CMP     TEMP,XCSR
2322  005050  001420                        BEQ     17$
2323  005052  105777  173612          15$:  TSTB    @XCSR           ;TEST PRINTER READY
2324  005056  100375                        BPL     15$             ;WAIT FOR READY
2325  005060  062737  000002  000670        ADD     #2,XCSR         ;SET XCSR TO PRINTER BUFFER
2326  005066  010077  173576                MOV     R0,@XCSR        ;LOAD CHARACTER INTO BUFFER
2327  005072  005337  000666          16$:  DEC     COUNT3          ;DECREASE COUNT OF DL11'S
2328  005076  001411                        BEQ     18$             ;ALL DONE,EXIT
2329  00' 00  062737  000002  000670        ADD     #2,XCSR         ;SET XCSR TO NEXT DL11 PRINTER STATUS
2330  005106  000137  004504                JMP     4$              ;GO TEST NEXT DL11 READY FLAG
2331  005112  062737  000002  000670  17$:  ADD     #2,XCSR         ;SET XCSR TO PRINTER BUFFER
2332  005120  000764                        BR      16$             ;DO NOT LOAD BUFFER
2333  005122  005737  000710          18$:  TST     INCHK           ;WANT INPUT CHECK?
2334  005126  001111                        BNE     26$             ;NO, BRANCH
2335  005130  023727  000636  000020        CMP     RTNNO,#20       ;PRINTING TEST?
2336  005136  002004                        BGE     19$             ;BRANCH IF NOT
2337  005140  022737  104011  001772        CMP     #TTYCTL,WAITF   ;KEYBOARD CONTROL?
2338  005146  001101                        BNE     26$             ;SKIP INPUT CHECK IF NOT
2339  005150  105777  173426          19$:  TSTB    @CONADD         ;TEST IF ANY INPUT
2340  005154  100076                        BPL     26$             ;BRANCH IF NONE
2341  005156  013737  000602  000712        MOV     CONADD,TEMP     ;SET ADR
2342  005164  062737  000002  000712        ADD     #2,TEMP
2343  005172  117737  173514  000700        MOVB    @TEMP,TEMPCH
2344  005200  042737  177600  000700        BIC     #177600,TEMPCH  ;MASK UNWANTED BITS
2345  005206  023727  000700  000003        CMP     TEMPCH,#3       ;CHARACTER = CONTROL-C?
2346  005214  001013                        BNE     21$             ;CONTINUE IF NOT
2347  005216  023727  000636  000024        CMP     RTNNO,#24       ;TEST 24?
2348  005224  001007                        BNE     21$             ;CONTINUE IF NOT
```

```
 2349  005226  012700  000036          20$:    MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
 2350  005232  104010                          DELAY
 2351  005234  104012                          CRLF                    ;SEND CR-LF
 2352  005236  022626                          POPSP2                  ;RESET STACK
 2353  005240  000137  013030                  JMP     E024B           ;RETURN TO TEST
 2354  005244  023727  000700  000177  21$:    CMP     TEMPCH,#177     ;CHECK IF RUBOUT
 2355  005252  001006                          BNE     23$             ;BRANCH IF NO
 2356  005254  000607                          BR      9$
 2357  005256  012737  000001  000634  22$:    MOV     #1,CNTLSW       ;CLEAR LOOP AND SEQUENCE BITS
 2358  005264  000137  002154                  JMP     TTY1B           ;GO WAIT FOR NEXT TEST
 2359  005270  010046                  23$:    MOV     R0,-(SP)        ;SAVE R0
 2360  005272  012700  000036                  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
 2361  005276  104010                          DELAY
 2362  005300  012600                          MOV     (SP)+,R0        ;RESTORE R0
 2363  005302  023727  000636  000017          CMP     RTNNO,#17       ;CHECK IF TEST 17
 2364  005310  001002                          BNE     24$             ;BRANCH IF NOT TEST 17
 2365  005312  013703  000700                  MOV     TEMPCH,R3       ;STORE INPUTTED CHARACTER
 2366  005316  023727  000636  000021  24$:    CMP     RTNNO,#21       ;CHECK IF TEST 21
 2367  005324  001003                          BNE     25$             ;BRANCH IF NOT TEST 21
 2368  005326  013737  000700  000662          MOV     TEMPCH,REPT     ;STORE INPUTTED CHARACTER
 2369  005334  023727  000636  000022  25$:    CMP     RTNNO,#22       ;CHECK IF TEST 22
 2370  005342  001003                          BNE     26$             ;BRANCH IF NOT TEST 22
 2371  005344  013737  000700  000662          MOV     TEMPCH,REPT     ;STORE INPUTTED CHARACTER
 2372  005352  000002                  26$:    RTI                     ;RETURN TO TEST
 2373
```

2374

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

005354  000040                  AT0:    40                      ;TEST NUMBER
005356  005406                  AT0X:   AT1                     ;NEXT TEST
005360  000012                          10.                     ;ITERATION COUNT
005362  005372                          1$                      ;SCOPE ENTRY
005364  012737  005402  000004          MOV     #3$,MACHER      ;SET UP MACHINE ERROR TRAP
005372  005777  173214          1$:     TST     @TKS            ;REFERENCE RECEIVER STATUS WORD
005376  104005                  2$:     CHAIN                   ;CHAIN TO NEXT TEST
005400  000774                          BR      1$              ;REPEAT TEST
005402  104001                  3$:     ERROR                   ;ERROR TRAPPED WHEN REFERENCING
005404  000774                          BR      2$              ;RECEIVER STATUS WORD (TKS)

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

005406  000041                  AT1:    41                      ;TEST NUMBER
005410  005440                          AT2                     ;NEXT TEST
005412  000012                          10.                     ;ITERATION COUNT
005414  005424                          1$                      ;SCOPE ENTRY
005416  012737  005434  000004          MOV     #3$,MACHER      ;SET UP MACHINE ERROR TRAP
005424  005777  173164          1$:     TST     @TKB            ;REFERENCE RECEIVER BUFFER
005430  104005                  2$:     CHAIN                   ;CHAIN TO NEXT TEST
005432  000774                          BR      1$              ;REPEAT TEST
005434  104001                  3$:     ERROR                   ;TRAPPED WHEN REFERENCING
005436  000774                          BR      2$              ;RECEIVER BUFFER (TKB)
```

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

005440  000042                  AT2:    42                      ;TEST NUMBER
005442  005472                          AT3                     ;NEXT TEST
005444  000012                          10.                     ;ITERATION COUNT
005446  005456                          1$                      ;SCOPE ENTRY
005450  012737  005466  000004          MOV     #3$,MACHER      ;SET UP MACHINE ERROR TRAP
005456  005777  173134          1$:     TST     @TPS            ;REFERENCE TRANSMITTER STATUS
005462  104005                  2$:     CHAIN                   ;CHAIN TO NEXT TEST
005464  000774                          BR      1$              ;REPEAT TEST
005466  104001                  3$:     ERROR                   ;TRAPPED WHEN REFERENCING
005470  000774                          BR      2$              ;TRANSMITTER STATUS WORD

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

005472  000043                  AT3:    43                      ;TEST NUMBER
005474  005524                          AT4                     ;NEXT TEST
005476  000012                          10.                     ;ITERATION COUNT
005500  005510                          1$                      ;SCOPE ENTRY
005502  012737  005520  000004          MOV     #3$,MACHER      ;SET UP ERROR TRAP
005510  005777  173104          1$:     TST     @TPB            ;REFERENCE TRANSMITTER BUFFER
005514  104005                  2$:     CHAIN                   ;CHAIN TO NEXT TEST
005516  000774                          BR      1$              ;REPEAT TEST
005520  104001                  3$:     ERROR                   ;TRAPPED WHEN REFERENCING
005522  000774                          BR      2$              ;TRANSMITTER BUFFER.
```

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

005524  000044                  AT4:    44                      ;TEST NUMBER
005526  005614                          AT5                     ;NEXT TEST
005530  000012                          10.                     ;ITERATION COUNT
005532  005546                          1$                      ;SCOPE ENTRY
005534  012746  000340                  MOV     #PRTY7,-(SP)    ;SET PRIORITY 7
005540  012746  005546                  MOV     #1$,-(SP)
005544  000002                          RTI
005546  052777  000100  173036  1$:     BIS     #BIT6,@TKS      ;SET INTERRUPT ENABLE BIT
005554  032777  000100  173030          BIT     #BIT6,@TKS      ;CHECK IF BIT IS SET
005562  001002                          BNE     3$              ;BRANCH IF SET
005564  104001                  2$:     ERROR                   ;NOT SET, ERROR
005566  000410                          BR      5$              ;CHAIN TO NEXT TEST
005570  042777  000100  173014  3$:     BIC     #BIT6,@TKS      ;CLEAR INTERRUPT ENABLE BIT
005576  032777  000100  *73006          BIT     #BIT6,@TKS      ;CHECK IF BIT IS CLEARED
005604  001401                          BEQ     5$              ;BRANCH IF CLEARED
005606  104001                  4$:     ERROR                   ;NOT CLEARED, ERROR
005610  104005                  5$:     CHAIN                   ;CHAIN TO NEXT TEST
005612  000755                          BR      1$              ;DO TEST AGAIN

                                        ;**********
                                        ;AT5--  TEST #45--CHECKS THAT THE RECEIVER INTERRUPT
                                        ;       ENABLE BIT CAN BE CLEARED WITH RESET INSTRUCTION.
                                        ;**********
                                        ;
005614  000045                  AT5:    45                      ;TEST NUMBER
005616  005672                          AT6                     ;NEXT TEST
005620  000012                          10.                     ;ITERATION COUNT
005622  005636                          1$                      ;SCOPE ENTRY
005624  012746  000340                  MOV     #PRTY7,-(SP)    ;SET PRIORITY TO 7
005630  012746  005636                  MOV     #1$,-(SP)
005634  000002                          RTI
005636  052777  000100  172746  1$:     BIS     #BIT6,@TKS      ;SET INTERRUPT ENABLE BIT
005644  105777  172746          3$:     TSTB    @TPS            ;BE SURE PRINTER IS DONE WITH DL11S1 MESSAGE
005650  001775                          BEQ     3$              ;BEFORE ALLOWING FOLLOWING RESET.
005652  000005                          RESET                   ;RESET
005654  032777  000100  172730          BIT     #BIT6,@TKS      ;TEST INTERRUPT ENABLE BIT
005662  001401                          BEQ     2$              ;BRANCH IF CLEARED
005664  104001                          ERROR                   ;STILL SET,ERROR
005666  104005                  2$:     CHAIN                   ;CHAIN TO NEXT ROUTINE
005670  000762                          BR      1$              ;REPEAT TEST
```

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

005672  000046                 AT6:    46                          ;TEST NUMBER
005674  005762                         AT7                         ;NEXT TEST
005676  000012                         10.                         ;ITERATION COUNT
005700  005714                         1$                          ;SCOPE ENTRY
005702  012746  000340                 MOV     #PRTY7,-(SP)        ;SET PRIORITY TO 7
005706  012746  005714                 MOV     #1$,-(SP)
005712  000002                         RTI
005714  052777  000100  172674 1$:     BIS     #BIT6,@TPS          ;SET INTERRUPT ENABLE BIT
005722  032777  000100  172666         BIT     #BIT6,@TPS          ;CHECK THAT BIT IS SET
005730  001002                         BNE     2$                  ;BRANCH IF SET
005732  104001                         ERROR                       ;NOT SET, ERROR
005734  000410                         BR      3$                  ;CHAIN TO NEXT TEST
005736  042777  000100  172652 2$:     BIC     #BIT6,@TPS          ;CLEAR INTERRUPT ENABLE BIT
005744  032777  000100  172644         BIT     #BIT6,@TPS          ;CHECK IF BIT IS CLEARED
005752  001401                         BEQ     3$                  ;BRANCH IF CLEARED
005754  104001                         ERROR                       ;NOT CLEARED, ERROR
005756  104005                 3$:     CHAIN                       ;CHAIN TO NEXT TEST
005760  000755                         BR      1$                  ;DO AGAIN

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

005762  000047                 AT7:    47                          ;TEST NUMBER
005764  006032                         AT10                        ;NEXT TEST
005766  000012                         10.                         ;ITERATION COUNT
005770  006004                         1$                          ;SCOPE ENTRY
005772  012746  000340                 MOV     #PRTY7,-(SP)        ;SET PRIORITY TO 7
005776  012746  006004                 MOV     #1$,-(SP)
006002  000002                         RTI
006004  052777  000100  172604 1$:     BIS     #BIT6,@TPS          ;SET INTERRUPT BIT
006012  000005                         RESET                       ;RESET
006014  032777  000100  172574         BIT     #BIT6,@TPS          ;CHECK IF BIT IS CLEARED
006022  001401                         BEQ     2$                  ;BRANCH IF CLEARED
006024  104001                         ERROR                       ;ERROR, RESET DID NOT CLEAR BIT
006026  104005                 2$:     CHAIN                       ;CHAIN TO NEXT ROUTINE
006030  000765                         BR      1$                  ;REPEAT TEST
```

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

006032  000050                  AT10:   50                      ;TEST NUMBER
006034  006070                          AT11                    ;NEXT TEST
006036  000012                          10.                     ;ITERATION COUNT
006040  006042                          1$                      ;SCOPE ENTRY
006042  032777  001000  172644  1$:     BIT     #LSI11,@SR     ;SKIP TEST IF AN LSI-11
006050  001005                          BNE     2$
006052  000005                          RESET                   ;RESET
006054  105777  172536                  TSTB    @TPS            ;CHECK TRANSMIT READY BIT
006060  100401                          BMI     2$              ;BRANCH IF SET
006062  104001                          ERROR                   ;ERROR, RESET DID NOT SET READY BIT
006064  104005                  2$:     CHAIN                   ;CHAIN TO NEXT TEST
006066  000765                          BR      1$              ;DO AGAIN

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

006070  000051                  AT11:   51                      ;TEST NUMBER
006072  006130                          AT12                    ;NEXT TEST
006074  000012                          10.                     ;ITERATION COUNT
006076  006100                          1$                      ;SCOPE ENTRY
006100  012700  000226          1$:     MOV     #226,R0
006104  104010                          DELAY                   ;DELAY 150 MSEC.
006106  000005                          RESET                   ;RESET
006110  005077  172504                  CLR     @TPB            ;LOAD TRANSMITTER BUFFER
006114  105777  172476                  TSTB    @TPS            ;CHECK TRANSMIT READY BIT
006120  100001                          BPL     2$              ;BRANCH IF CLEARED
006122  104001                          ERROR                   ;NOT CLEARED, ERROR
006124  104005                  2$:     CHAIN                   ;CHAIN TO NEXT TEST
006126  000764                          BR      1$              ;REPEAT TEST
```

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

006130  000052                  AT12:  52                      ;TEST NUMBER
006132  006204                         AT13                    ;NEXT TEST
006134  000012                         10.                     ;ITERATION COUNT
006136  006144                         1$                      ;SCOPE ENTRY
006140  104004                         STPCHV                  ;SET UP TRANSMITTER INTERRUPT VECTOR
006142  006200                         4$                      ;TO 4$
006144  000005                  1$:    RESET                   ;SEE CHAINY COMMENT
006146  005077  172444                 CLR     @TPS            ;DISABLE TRANSMIT INTERRUPT
006152  005046                         CLR     -(SP)           ;SET PRIORITY TO ZERO
006154  012746  006162                 MOV     #2$,-(SP)
006160  000002                         RTI
006162  052777  000100  172426  2$:    BIS     #BIT6,@TPS      ;ENABLE TRANSMIT INTERRUPT
006170  000240                         NOP
006172  104001                         ERROR                   ;TRANSMIT READY DID NOT CAUSE INTERRUPT
006174  104005                  3$:    CHAIN                   ;CHAIN TO NEXT TEST
006176  000762                         BR      1$              ;REPEAT TEST
006200  022626                  4$:    POPSP2                  ;INTERRUPT OCCURRED, CLEAN STACK
006202  000774                         BR      3$              ;CHAIN TO NEXT TEST

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

006204  000035                  AT13:  35                      ;TEST NUMBER
006206  006264                         AT14                    ;NEXT TEST
006210  000012                         10.                     ;ITERATION COUNT
006212  006220                         1$                      ;SCOPE ENTRY
006214  104004                         STPCHV                  ;SET UP TRANSMIT INTERRUPT
006216  006256                         4$                      ;VECTOR TO 4$
006220  013746  000630          1$:    MOV     TPLVL,-(SP)     ;SET PROCESSOR TO SAME LEVEL AS XMITTER
006224  012746  006232                 MOV     #2$,-(SP)
006230  000002                         RTI
006232  005077  172360          2$:    CLR     @TPS            ;DISABLE TRANSMITTER INTERRUPTS
006236  052777  000100  172352         BIS     #BIT6,@TPS      ;ENABLE TRANSMITTER INTERRUPTS
006244  000240                         NOP
006246  005077  172344          3$:    CLR     @TPS            ;OK, NO INTERRUPT OCCURRED
006252  104005                         CHAIN                   ;CHAIN TO NEXT TEST
006254  000761                         BR      1$              ;REPEAT TEST
006256  022626                  4$:    POPSP2                  ;INTERRUPT OCCURRED,ERROR,CLEAN
006260  104001                         ERROR                   ;UP STACK
006262  000771                         BR      3$              ;CHAIN TO NEXT TEST
```

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

006264  000054          AT14:  54                      ;TEST NUMBER
006266  006350                 AT15                    ;NEXT TEST
006270  000012                 10.                     ;ITERATION COUNT
006272  006300                 1$                      ;SCOPE ENTRY
006274  104004                 STPCHV                  ;SET UP TRANSMIT INTERRUPT
006276  006336                 3$                      ;VECTOR TO 3$
006300  005077  172312    1$:  CLR     @TPS             ;DISABLE TRANSMIT INTERRUPTS
006304  013746  000630         MOV     TPLVL,-(SP)      ;SET PROCESSOR PRIORITY ONE
006310  162716  000040         SUB     #40,(SP)         ;LEVEL LOWER THAN TRANSMITTER
006314  012746  006322         MOV     #2$,-(SP)
006320  000002                 RTI
006322  052777  000100  172266 2$:  BIS     #BIT6,@TPS       ;ENABLE TRANSMITTER INTERRUPTS
006330  000240                 NOP
006332  104001                 ERROR                   ;NO INTERRUPT, ERROR
006334  000401                 BR      4$              ;CHAIN TO NEXT TEST
006336  022626          3$:    POPSP2                  ;INTERRUPT OCCURED, OK, CLEAN STACK
006340  005077  172252    4$:  CLR     @TPS             ;DISABLE TRANSMITTER INTERRUPTS
006344  104005                 CHAIN                   ;CHAIN TO NEXT TEST
006346  000754                 BR      1$              ;REPEAT TEST
```

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

006350  000055          AT15:   55                      ;TEST NUMBER
006352  006450                  AT16                    ;NEXT TEST
006354  000012                  10.                     ;ITERATION COUNT
006356  006360                  1$                      ;SCOPE ENTRY
006360  104004          1$:     STPCHV                  ;SET TRANSMIT INTERRUPT VECTOR
006362  006422                  4$                      ;TO 4$
006364  005077  172226          CLR     @TPS            ;DISABLE TRANSMITTER INTERRUPTS
006370  005046                  CLR     -(SP)           ;SET PROCESSOR PRIORITY TO ZERO
006372  012746  006400          MOV     #2$,-(SP)
006376  000002                  RTI
006400  052777  000100  172210 2$:     BIS     #BIT6,@TPS      ;ENABLE TRANSMITTER INTERRUPTS
006406  000240                  NOP
006410  104001                  ERROR                   ;ERROR1, TRANSMITTER FAILED TO INTERRUPT
006412  005077  172200          3$:     CLR     @TPS            ;DISABLE TRANSMITTER INTERRUPTS
006416  104005                  CHAIN                   ;CHAIN TO NEXT TEST
006420  000757                  BR      1$              ;REPEAT TEST
006422  012777  006442  172176 4$:     MOV     #6$,@TPVTR      ;INTERRUPT OCCURRED, CHANGE INTERRUPT
006430  012716  006436          MOV     #5$,@SP         ;VECTOR TO 6$ AND RETURN TO 5$
006434  000002                  RTI                     ;RETURN FROM INTERRUPT
006436  000240          5$:     NOP
006440  000764                  BR      3$              ;CHAIN TO NEXT TEST
006442  022626          6$:     POPSP2                  ;ERROR2, TRANSMITTER REINTERRUPTED
006444  104001                  ERROR                   ;AFTER RTI WITH READY BIT LEFT ON.
006446  000761                  BR      3$              ;CLEAN STACK, CHAIN TO NEXT TEST.

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

006450  000056          AT16:   56                      ;TEST NUMBER
006452  006516                  AT17                    ;NEXT TEST
006454  000012                  10.                     ;ITERATION COUNT
006456  006460                  1$                      ;SCOPE ENTRY
006460  032777  001000  172226 1$:     BIT     #LSI11,@SR      ;SKIP TEST IF LSI-11
006466  001011                  BNE     3$
006470  012700  000226          MOV     #226,R0
006474  104010                  DELAY                   ;DELAY 150 MSEC.
006476  104021          2$:     AREAD                   ;ENABLE RECEIVER
006500  000005                  RESET                   ;RESET
006502  105777  172104          TSTB    @TKS            ;TEST DONE BIT
006506  100001                  BPL     3$              ;BRANCH IF DONE IS CLEARED
006510  104001                  ERROR                   ;NOT CLEARED, ERROR
006512  104005          3$:     CHAIN                   ;CHAIN TO NEXT TEST
006514  000761                  BR      1$              ;REPEAT TEST
```

```
                                        ;**********
                                        ;AT17-- TEST#57--CHECKS THAT REFERENCING THE RECEIVER BUFFER
                                        ;       CLEARS THE DONE BIT.
                                        ;**********

006516  000057                  AT17:  57                      ;TEST NUMBER
006520  006566                         AT20                    ;NEXT TEST
006522  000012                         10.                     ;ITERATION COUNT
006524  006526                         1$                      ;SCOPE ENTRY
006526  032777  001000  172160  1$:    BIT     #LSI11,@SR     ;CHECK FOR LSI-11
006534  001012                         BNE     3$              ;SKIP TEST IF SET
006536  012700  000226                 MOV     #226,R0
006542  104010                         DELAY                   ;DELAY 150 MSEC.
006544  104021                  2$:    AREAD                   ;ENABLE RECEIVER
006546  105777  172042                 TSTB    @TKB            ;REFERENCE RECEIVER BUFFER
006552  105777  172034                 TSTB    @TKS            ;TEST DONE BIT
006556  100001                         BPL     3$              ;BRANCH IF NOT SET
006560  104001                         ERROR                   ;DONE BIT IS SET, ERROR
006562  104005                  3$:    CHAIN                   ;CHAIN TO NEXT TEST
006564  000760                         BR      1$              ;REPEAT TEST

                                        ;**********
                                        ;AT20-- TEST#60--CHECK THAT THE RECEIVER DONE BIT IS ABLE TO
                                        ;       CAUSE AN INTERRUPT.
                                        ;**********

006566  000060                  AT20:  60                      ;TEST NUMBER
006570  006660                         AT21                    ;NEXT TEST
006572  000012                         10.                     ;ITERATION COUNT
006574  006602                         1$                      ;SCOPE ENTRY
006576  104003                         STRDRV                  ;SET UP RECEIVER INTERRUPT
006600  006652                         4$                      ;VECTOR TO 4$
006602  032777  001000  172104  1$:    BIT     #LSI11,@SR     ;CHECK FOR LSI-11
006610  001021                         BNE     5$              ;SKIP TEST IF SET
006612  012700  000226                 MOV     #226,R0
006616  104010                         DELAY                   ;DELAY 150 MSEC
006620  104021                         AREAD                   ;ENABLE RECEIVER
006622  005077  171764          2$:    CLR     @TKS            ;DISABLE RECEIVER INTERRUPTS
006626  005046                         CLR     -(SP)           ;SET PROCESS STATUS TO ZERO
006630  012746  006636                 MOV     #3$,-(SP)
006634  000002                         RTI
006636  052777  000100  171746  3$:    BIS     #BIT6,@TKS     ;ENABLE RECEIVER INTERRUPT
006644  000240                         NOP
006646  104001                         ERROR                   ;ERROR,RECEIVER FAILED TO INTERRUPT
006650  000401                         BR      5$              ;CHAIN TO NEXT TEST
006652  022626                  4$:    POPSP2                  ;OK, CLEAN STACK
006654  104005                  5$:    CHAIN                   ;CHAIN TO NEXT TEST
006656  000751                         BR      1$              ;REPEAT TEST
```

```
                                        ;**********
                                        ;AT21-- TEST#61--TESTS THAT THE RECEIVER DONE DOES NOT CAUSE AN
                                        ;       INTERRUPT WHEN THE PROCESSOR IS AT THE SAME LEVEL AS
                                        ;       THE RECEIVER'S INTERRUPT REQUEST LEVEL.
                                        ;**********

006660  000061                  AT21:   61                      ;TEST NUMBER
006662  006764                          AT22                    ;NEXT TEST
006664  000012                          10.                     ;ITERATION COUNT
006666  006700                          ADD2                    ;SCOPE ENTRY
006670  104003                          STRDRV                  ;SET RECEIVER VECTOR TO ADD1
006672  006756                          ADD1
006674  005077  171712          CHGF4:  CLR     @TKS            ;DISABLE INTERRUPTS
006700  032777  001000  172006  ADD2:   BIT     #LSI11,@SR      ;CHECK FOR LSI-11
006706  001017                          BNE     ADD3            ;SKIP TEST IF SET
006710  012700  000226                  MOV     #226,R0
006714  104010                          DELAY                   ;DELAY 150 MSEC
006716  104021                          AREAD                   ;ENABLE RECEIVER
006720  005077  171666          2$:     CLR     @TKS            ;DISABLE RECEIVER INTERRUPTS
006724  013746  000624                  MOV     TKLVL,-(SP)     ;SET PROCESSOR PRIORITY TO SAME LEVEL AS RECEIVER
006730  012746  006736                  MOV     #3$,-(SP)
006734  000002                          RTI
006736  052777  000100  171646  3$:     BIS     #BIT6,@TKS      ;ENABLE RECEIVER INTERRUPTS
006744  000240                          NOP
006746  005077  171640          ADD3:   CLR     @TKS            ;OK, NO INTERRUPT OCCURRED
006752  104005                          CHAIN                   ;CHAIN TO NEXT TEST
006754  000751                          BR      ADD2            ;REPEAT TEST
006756  022626                  ADD1:   POPSP2                  ;ERROR, RECEIVER INTERRUPTED, CLEAN STACK
006760  104001                          ERROR
006762  000771                          BR      ADD3            ;BRANCH ADD3
```

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

006764  000062          AT22:   62                      ;TEST NUMBER
006766  007072                  AT23                    ;NEXT TEST
006770  000012                  10.                     ;ITERATION COUNT
006772  007000                  1$                      ;SCOPE ENTRY
006774  104003                  STRDRV                  ;SET RECEIVER INTERRUPT
006776  007060                  4$                      ;VECTOR TO 4$
007000  032777  001000  171706 1$:     BIT     #LSI11,@SR      ;CHECK FOR LSI11
007006  001025                  BNE     5$              ;SKIP TEST IF SET
007010  012700  000226                  MOV     #226,R0
007014  104010                  DELAY                   ;DELAY 150 MSEC
007016  104021                  AREAD                   ;ENABLE RECEIVER
007020  005077  171566          2$:     CLR     @TKS            ;DISABLE READER INTERRUPTS
007024  013746  000624                  MOV     TKLVL,-(SP)     ;SET PROCESSOR PRIORITY ONE LEVEL
007030  012746  007036                  MOV     #3$,-(SP)
007034  000002                  RTI
007036  162737  000040  177776 3$:     SUB     #40,PSW         ;LOWER THAN READER
007044  052777  000100  171540          BIS     #BIT6,@TKS      ;ENABLE INTERRUPTS
007052  000240                  NOP
007054  104001                  ERROR                   ;FAILED TO INTERRUPT
007056  000401                  BR      5$              ;CHAIN TO NEXT TEST
007060  022626                  4$:     POPSP2                  ;OK, CLEAN STACK
007062  005077  171524          5$:     CLR     @TKS            ;DISABLE RECEIVER INTERRUPTS
007066  104005                  CHAIN                   ;CHAIN TO NEXT TEST
007070  000743                  BR      1$              ;REPEAT TEST
```

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

007072  000063                  AT23:   63                      ;TEST NUMBER
007074  007204                          AT24                    ;NEXT TEST
007076  000012                          10.                     ;ITERATION COUNT
007100  007102                          1$                      ;SCOPE ENTRY
007102  032777  001000  171604  1$:     BIT     #LSI11,@SR     ;CHECK FOR LSI-11
007110  001015                          BNE     3$              ;SKIP TEST IF SET
007112  012700  000226          2$:     MOV     #226,R0
007116  104010                          DELAY                   ;DELAY 150 MSEC
007120  104021                          AREAD                   ;ENABLE RECEIVER
007122  104003                          STRDRV                  ;SET RECEIVER INTERRUPT
007124  007156                          4$                      ;VECTOR TO 4$
007126  005077  171460                  CLR     @TKS            ;DISABLE RECEIVER INTERRUPTS
007132  052777  000100  171452          BIS     #BIT6,@TKS      ;ENABLE RECEIVER INTERRUPT
007140  000240                          NOP                     ;
007142  104001                          ERROR                   ;NO INTERRUPT, ERROR
007144  005077  171442          3$:     CLR     @TKS            ;DISABLE RECEIVER INTERRUPTS
007150  000005                          RESET                   ;RESET AFTER LAST INTERRUPT
007152  104005                          CHAIN                   ;CHAIN TO NEXT TEST
007154  000752                          BR      1$              ;REPEAT TEST
007156  012777  007176  171436  4$:     MOV     #6$,@TKVTR      ;INTERRUPT, OK, CHANGE VECTOR TO 6$
007164  012716  007172                  MOV     #5$,@SP         ;CHANGE RET ADDR TO 5$
007170  000002                          RTI                     ;RETURN
007172  000240                  5$:     NOP
007174  000763                          BR      3$              ;OK, NO ADDITIONAL INTERRUPT
007176  022626                  6$:     POPSP2                  ;ERROR, ADDITIONAL INTERRUPT
007200  104001                          ERROR
007202  000760                          BR      3$              ;CHAIN TO NEXT TEST
```

```
                                        ;*********
                                        ;AT24--TEST#64--HAVE OPERATOR TYPE A CHARACTER ON THE
                                        ;               KEYBOARD, THEN CHECK FOR RECEIVER DONE.
                                        ;               CHECK THAT RECEIVER DONE CAUSES AN INTERRUPT
                                        ;               WHEN BIT 6 (INTERRUPT ENABLE) IS SET.
                                        ;               CHECK THAT READING TKB CLEARS DONE BIT AND
                                        ;               THAT DONE CLEARED DOES NOT CAUSE AN INTERRUPT.
                                        ;
                                        ;               ALLOW 12 SECONDS FOR OPERATOR RESPONSE.
                                        ;*********

007204                                  CHGF7:
007204  000064                        AT24:   64                      ;TEST NUMBER
007206  177777                                -1                      ;LAST TEST
007210  000001                                1                       ;ITERATION COUNT
007212  007214                                1$                      ;SCOPE ENTRY
007214  032777  001000  171472        1$:     BIT     #LSI11,@SR      ;SKIP TEST IF NOT AN LSI-11
007222  001455                                BEQ     10$
007224  005777  171362                        TST     @TKS            ;SHOULD BE CLEAR
007230  001401                                BEQ     2$
007232  104001                                ERROR                   ;RECEIVER STATUS NOT -0
007234  012700  000600                2$:     MOV     #600,R0         ;1/2 SEC DELAY
007240  012737  000030  000716                MOV     #30,CNTR        ;SET UP FOR 12 SEC WAIT
007246  104000                                TYPE
007250  015101                                OPMSG                   ;MESSAGE TO TYPE A CHARACTER
007252  104010                        3$:     DELAY                   ;1/2 SECOND
007254  105777  171332                        TSTB    @TKS            ;CHECK DONE BIT
007260  100407                                BMI     5$              ;SET - EXIT LOOP
007262  005337  000716                        DEC     CNTR
007266  001403                                BEQ     4$              ;TIME HAS RUN OUT...
007270  012700  000600                        MOV     #600,R0         ;ANOTHER 1/2 SEC
007274  000766                                BR      3$              ;CONTINUE WAIT
007276  104001                        4$:     ERROR                   ;NO RECEIVER DONE, OR
                                                                        ;OPERATOR DID NOT RESPOND
007300  104003                        5$:     STRDRV                  ;SET RECEIVER INTERRUPT
007302  007322                                8$                      ;TO 8$
007304  052777  000100  171300                BIS     #BIT6,@TKS      ;ENABLE INTERRUPT
007312  000240                                NOP
007314  000240                                NOP
007316  104001                        6$:     ERROR                   ;RECEIVER DID NOT INTERRUPT
007320  022626                        7$:     POPSP2                  ;CLEAN UP THE STACK
007322  017737  171266  000716        8$:     MOV     @TKB,CNTR       ;READ DATA BUFFER
007330  105777  171256                        TSTB    @TKS            ;CHECK THE DONE BIT
007334  100001                                BPL     9$              ;OK
007336  104001                                ERROR                   ;READING DATA BUFFER DID NOT CLEAR DONE
007340  104003                        9$:     STRDRV                  ;SET RECEIVER INTERRUPT
007342  007366                                11$                     ;VECTOR TO 11$
007344  052777  000100  171240                BIS     #BIT6,@TKS      ;ENABLE INTERRUPT
007352  000240                                NOP
007354  000240                                NOP
007356  005077  171230                10$:    CLR     @TKS            ;OK, CLEAN UP
007362  104005                                CHAIN                   ;EXIT TESTS
007364  000713                                BR      1$
007366  104001                        11$:    ERROR                   ;DLV INTERRUPTED WITH DONE CLEARED
007370  022626                                POPSP2                  ;CLEAN UP THE STACK
007372  000771                                BR      10$             ;EXIT TESTS
```

```
2897
2898
```

```
2899                                    .SBTTL  LA36 PRINTER TESTS
2900                                    ;
2901                                    ;THE LA36 PRINTER TESTS WILL BE EXECUTED IN A
2902                                    ;CONTINOUS LOOP OUTPUTTING TO ALL MULTIPLE DL11'S
2903                                    ;IF SR BIT 8 IS SET TO ZERO AT START UP TIME.  IF
2904                                    ;BIT 8 IS SET TO 1 AT START UP THEY MAY BE EXECUTED
2905                                    ;INDIVIDUALLY ONCE OR CONTINUALLY LOOPED, OR
2906                                    ;BECOME THE FIRST OF THE ENTIRE SEQUENCE OF PRINTER
2907                                    ;TESTS.  REFERENCE INTRUCTIONS IN THE INTRODUCTION
2908                                    ;FOR PROPER MODE OF OPERATION.
2909                                    ;
2910                                    ;
2911                                    ;XXXXXXXXXX
2912                                    ;
2913                                    ;PT0 -- DATA PATH TEST---FOUR LINES OF ALTERNATING
2914                                    ;       "*" AND "U" ARE PRINTED, OUT TO THE GIVEN PAPER
2915                                    ;       WIDTH.  THE PATTERN WILL APPEAR AS FOLLOWS.
2916                                    ;
2917                                    ;       *U*U*U*U*U*U
2918                                    ;       U*U*U*U*U*U*
2919                                    ;       *U*U*U*U*U*U
2920                                    ;       U*U*U*U*U*U*
2921                                    ;
2922                                    ;XXXXXXXXXX
2923                                    ;
2924  007374  000000           PT0:     0                       ;TEST NUMBER
2925  007376  007450                    PT1                     ;NEXT TEST
2926  007400  104016                    PRTHDR                  ;
2927  007402  104007                    TYPEM                   ;PRINT COLUMN # MESG
2928  007404  014627                    HDR0
2929  007406  012703  025125   1$:      MOV     #'U*,R3         ;SET FIRST CHAR PAIR
2930  007412  012702  000004           MOV     #4,R2           ;SET LINE COUNT
2931  007416  010300           2$:      MOV     R3,R0           ;SET CHAR PAIR
2932  007420  013701  000652           MOV     WIDTH,R1        ;SET COLUMN COUNT
2933  007424  104015           3$:      PRINTC                  ;PRINT CHAR
2934  007426  000300                    SWAB    R0              ;SET NEXT CHAR
2935  007430  005301                    DEC     R1              ;DEC COLUMN COUNT
2936  007432  001374                    BNE     3$              ;FINISH LINE
2937  007434  000303                    SWAB    R3              ;SET NEXT LINE START CHAR
2938  007436  104012                    CRLF                    ;SEND CR-LF
2939  007440  005302                    DEC     R2              ;DEC LINE COUNT
2940  007442  001365                    BNE     2$              ;FINISH TEST
2941  007444  104005                    CHAIN                   ;ALL DONE, EXIT
2942  007446  000757                    BR      1$              ;REPEAT TEST
```

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

007450  000001          PT1:    1                       ;TEST NUMBER
007452  007572                  PT2                     ;NEXT TEST
007454  104016                  PRTHDR
007456  012701  000040  1$:     MOV     #40,R1          ;SPACE TO R1
007462  012702  000100          MOV     #100,R2         ;@ TO R2
007466  012703  000140          MOV     #140,R3         ;\ TO R3
007472  110100          2$:     MOVB    R1,R0           ;CHAR TO R0
007474  004737  007542          JSR     PC,SPSP         ;SEND TWO SPACES
007500  110200                  MOVB    R2,R0           ;NEXT CHAR TO R0
007502  004737  007542          JSR     PC,SPSP         ;SEND TWO SPACES
007506  012704  000003          MOV     #3,R4           ;PRINT COUNT TO R4
007512  110300                  MOVB    R3,R0           ;THIRD CHAR TO R0
007514  104015          3$:     PRINTC                  ;PRINT THE CHAR
007516  005304                  DEC     R4              ;THREE TIMES ?
007520  001375                  BNE     3$              ;BRANCH IF NOT
007522  104012                  CRLF                    ;CARRIAGE RETURN LINE FEED
007524  122122                  CMPB    (R1)+,(R2)+     ;NEXT CHARACTERS
007526  105723                  TSTB    (R3)+
007530  020327  000200          CMP     R3,#200         ;CHECK IF ALL DONE
007534  103756                  BLO     2$              ;BRANCH IF NOT
007536  104005                  CHAIN                   ;EXIT TO NEXT TEST
007540  000746                  BR      1$              ;REPEAT TEST
007542  012704  000003  SPSP:   MOV     #3,R4           ;PRINT COUNT TO R4
007546  104015          1$:     PRINTC                  ;PRINT CHAR
007550  005304                  DEC     R4              ;THREE TIMES?
007552  001375                  BNE     1$              ;BRANCH IF NOT
007554  012700  000040  SP2:    MOV     #40,R0          ;SPACE TO R0
007560  104015                  PRINTC                  ;SEND A SPACE
007562  012700  000040  SPC:    MOV     #40,R0          ;SPACE TO R0
007566  104015                  PRINTC                  ;SEND ANOTHER
007570  000207                  RTS     PC              ;RETURN
```

```
                                        ;XXXXXXXXXX
                                        ;
                                        ;PT2 -- NON-PRINTING CHARACTER TEST.  THIS TEST
                                        ;       PRINTS THE OCTAL CODE FOLLOWED BY THE MNEMONIC
                                        ;       OF ALL NON-PRINTING CHARACTERS.  FOLLOWING EACH
                                        ;       MNEMONIC, THE PRINTER IS DRIVEN BY THE NON-PRINTING
                                        ;       CODE (000 THROUGH 037 PLUS 177)
                                        ;       ALL CONTROL CHARACTERS (INCLUDING THOSE FOR OPTIONS
                                        ;       WILL BE SKIPPED, REFER TO THE DOCUMENT FOR A LIST OF THOSE
                                        ;       TESTED.
                                        ;
                                        ;XXXXXXXXXX
                                        ;
007572  000002                  PT2:    2                       ;TEST NUMBER
007574  010166                          PT3                     ;NEXT TEST
007576  104016                          PRTHDR                  ;PRINT TEST HEADER
007600  012701  007700          1$:     MOV     #IDEZ,R1        ;ADDR OF IDENT TO R1
007604  012703  010141                  MOV     #NPCODE,R3      ;ADDR OF NON-PRINT-CODES TO R3
007610  012702  000003          2$:     MOV     #3,R2           ;NO. OF ID'S PER LINE TO R2
007614  012704  000010          3$:     MOV     #10,R4          ;NO. OF CHARS PER ID TO R4
007620  121327  000055          4$:     CMPB    (R3),#55        ;ZERO TERMINATOR IN NP TABLE?
007624  001422                          BEQ     7$              ;BRANCH IF YES
007626  112100                          MOVB    (R1)+,R0        ;GET ID CHARACTERS
007630  104015                          PRINTC                  ;AND PRINT A
007632  005304                          DEC     R4              ;GROUP OF
007634  001371                          BNE     4$              ;8 CHARACTERS
007636  112300                          MOVB    (R3)+,R0        ;GET NP CODE FROM TABLE
007640  012704  000003                  MOV     #3,R4           ;AND
007644  104015                  5$:     PRINTC                  ;TRY TO PRINT IT
007646  005304                          DEC     R4              ;THREE
007650  001375                          BNE     5$              ;TIMES
007652  005302                          DEC     R2              ;MORE TO GO ON THIS LINE ?
007654  001404                          BEQ     6$              ;BRANCH IF NO
007656  004737  007554                  JSR     PC,SP2          ;SEND 3 SPACES
007662  104015                          PRINTC
007664  000753                          BR      3$              ;BRANCH TO CONTINUE LINE
007666  104012                  6$:     CRLF
007670  000747                          BR      2$              ;GO DO NEXT LINE
007672  104012                  7$:     CRLF
007674  104005                          CHAIN                   ;CHAIN TO NEXT TEST
007676  000740                          BR      1$
                                        
                                        
                                        
007700  030060  020060  047040  IDEZ:   .ASCII  /000  NUL001  SOH002  STX/
007706  046125  030060  020061
007714  051440  044117  030060
007722  020062  051440  054124
007730  030060  020066  040440          .ASCII  /006  ACK020  DLE021  DC1/
007736  045503  031060  020060
007744  042040  042514  031060
007752  020061  042040  030503
007760  031060  020062  042040          .ASCII  /022  DC2023  DC3024  DC4/
007766  031103  031060  020063
007774  042040  031503  031060
010002  020064  042040  032103
```

```
 3036  010010  031060  020065  047040          .ASCII  /025  NAK026  SYN027  ETB/
 3037  010016  045501  031060  020066
 3038  010024  051440  047131  031060
 3039  010032  020067  042440  041124
 3040  010040  031460  020060  041440          .ASCII  /030  CAN031  EM 032  SUB/
 3041  010046  047101  031460  020061
 3042  010054  042440  020115  031460
 3043  010062  020062  051440  041125
 3044  010070  031460  020064  043040          .ASCII  /034  FS 035  GS 036  RS /
 3045  010076  020123  031460  020065
 3046  010104  043440  020123  031460
 3047  010112  020066  051040  020123
 3048  010120  031460  020067  052440          .ASCII  /037  US 177  DEL /
 3049  010126  020123  033461  020067
 3050  010134  042040  046105     040
 3051  010141     000     002     006  NPCODE: .BYTE   0,2,6,20,21,22,23,24
 3052  010144     020     021     022
 3053  010147     023     024
 3054  010151     025     025     027          .BYTE   25,25,27,30,31,32,34,35
 3055  010154     030     031     032
 3056  010157     034     035
 3057  010161     036     037     177          .BYTE   36,37,177,55
 3058  010164     055
 3059          010166                          .EVEN
```

```
                                        ;XXXXXXXXXX
                                        ;
                                        ;PT3 -- CARRIAGE RETURN TEST
                                        ;
                                        ;       THE LINE CONSISTS OF A STRING OF O'S AND
                                        ;       X'S.  FIRST, THE O'S ARE PRINTED OUT TO THE LAST
                                        ;       COLUMN WITH A SPACE SEPARATING EACH.  THEN THE
                                        ;       CARRIAGE IS SPACED TO THE FIRST BLANK SPACE, AN X
                                        ;       IS PRINTED AND THEN RETURNED TO THE MARGIN.  THIS
                                        ;       PROCESS IS CONTINUE UNTIL ALL SPACES BETWEEN
                                        ;       THE ZEROES HAVE BEEN FILLED.
                                        ;
                                        ;XXXXXXXXXX
                                        
010166  000003                  PT3:    3                       ;TEST NUMBER
010170  010306                          PT4                     ;NEXT TEST
010172  104016                          PRTHDR                  ;TYPE HEADER
010174  005037  000674          1$:     CLR     SPCNT           ;CLEAR SPACE COUNTER
010200  013701  000652                  MOV     WIDTH,R1        ;POSITION COUNTER TO R1
010204  012700  000117          2$:     MOV     #117,R0         ;"O" TO R0
010210  104015                          PRINTC                  ;PRINT THE "O"
010212  005301                          DEC     R1              ;DECREMENT POSITION COUNTER
010214  001404                          BEQ     3$              ;BRANCH IF 0
010216  004737  007562                  JSR     PC,SPC          ;SEND SPACE
010222  005301                          DEC     R1              ;DECREMENT POSITION COUNTER
010224  001367                          BNE     2$              ;BRANCH IF NOT ZERO
010226  104022                  3$:     CR                      ;SEND A CR
010230  012737  000001  000674          MOV     #1,SPCNT        ;SPACE, COUNTER SET TO 1
010236  013701  000674          4$:     MOV     SPCNT,R1        ;NO. OF SPACES TO R1
010242  004737  007562          5$:     JSR     PC,SPC          ;SEND SPACE
010246  005301                          DEC     R1              ;DECREMENT SPACE COUNTER
010250  001374                          BNE     5$              ;BRANCH IF NOT ZERO
010252  012700  000130                  MOV     #130,R0         ;"X" INTO R0
010256  104015                          PRINTC                  ;PRINT "X"
010260  104022                          CR                      ;PRINT CR
010262  062737  000002  000674          ADD     #2,SPCNT        ;INCREMENT SPACE COUNT BY 2
010270  023737  000674  000652          CMP     SPCNT,WIDTH     ;COMPARE POSITION COUNTER WITH COLM. COUNT
010276  103757                          BLO     4$              ;BRANCH IF LOWER
010300  104014                          LF                      ;SEND LF
010302  104005                          CHAIN                   ;CHAIN TO NEXT TEST
010304  000733                          BR      1$              ;REPEAT TEST
```

```
3101                                    ;XXXXXXXXXX
3102                                    ;
3103                                    ;PT4 -- MULTIPLE LINE FEED TEST -- 63 LINE FEEDS ARE
3104                                    ;       SENT WITH A REFERENCE LINE AT THE START AND END.
3105                                    ;       A NUMBER IS PRINTED WHICH INDICATES THE NUMBER OF LINE
3106                                    ;       FEEDS THAT WILL BE ISSUED BEFORE THE NEXT
3107                                    ;       NUMBER OR REFERENCE LINE IS PRINTED.
3108                                    ;
3109                                    ;XXXXXXXXXX
3110
3111  010306  000004                    PT4:    4                       ;TEST NUMBER
3112  010310  010464                            PT5                     ;NEXT TEST
3113  010312  104016                            PRTHDR                  ;TYPE HEADER
3114  010314  012737  000001  000706    1$:     MOV     #1,LFCNT        ;LINE FEED COUNT TO 1
3115  010322  013701  000652                    MOV     WIDTH,R1        ;COLUMN COUNT TO R1
3116  010326  012702  010446                    MOV     #LINE3,R2       ;ADDR OF NUMBER FIELD TO R2
3117  010332  004737  010416                    JSR     PC,REF          ;PRINT REFERENCE LINE
3118  010336  013701  000706            2$:     MOV     LFCNT,R1        ;LINE FEED COUNT TO R1
3119  010342  104014                    3$:     LF                      ;SEND LF
3120  010344  005301                            DEC     R1              ;DECREMENT COUNTER
3121  010346  001375                            BNE     3$              ;BRANCH IF NOT YET 0
3122  010350  006337  000706                    ASL     LFCNT           ;DOUBLE LINE FEED COUNT
3123  010354  022737  000100  000706          CMP     #BIT6,LFCNT     ;TEST IF COUNT IS 32
3124  010362  001406                            BEQ     4$              ;BRANCH IF -32, END
3125  010364  112200                            MOVB    (R2)+,R0        ;NUMBER TO R0
3126  010366  104015                            PRINTC                  ;PRINT IT
3127  010370  112200                            MOVB    (R2)+,R0        ;NUMBER TO R0
3128  010372  104015                            PRINTC                  ;PRINT IT
3129  010374  104022                            CR                      ;PRINT CR
3130  010376  000757                            BR      2$              ;DRIVE THE LINEFEEDS
3131  010400  013701  000652            4$:     MOV     WIDTH,R1        ;COLUMN COUNT TO R1
3132  010404  004737  010416                    JSR     PC,REF          ;SEND END REFERENCE LINE
3133  010410  104014                            LF                      ;ADVANCE PAPER
3134  010412  104005                            CHAIN
3135  010414  000737                            BR      1$              ;REPEAT TEST
3136  010416  112200                    REF:    MOVB    (R2)+,R0        ;NUMBER TO R0
3137  010420  104015                            PRINTC                  ;PRINT IT
3138  010422  112200                            MOVB    (R2)+,R0        ;NUMBER TO R0
3139  010424  104015                            PRINTC                  ;PRINT IT
3140  010426  005741                            TST     -(R1)           ;DECREASE COUNTER BY 2
3141  010430  012700  000137                    MOV     #137,R0         ;DASH (-) TO R0
3142  010434  104015                    1$:     PRINTC                  ;PRINT IT
3143  010436  005301                            DEC     R1              ;DECREMENT COLUMN COUNTER
3144  010440  001375                            BNE     1$              ;BRANCH IF NO ZERO
3145  010442  104022                            CR                      ;PRINT CR
3146  010444  000207                            RTS     PC              ;RETURN
3147
3148  010446  030460  031060  032060  LINE3:  .ASCII  /010204081632 00/
3149  010454  034060  033061  031063
3150  010462  030060
```

```
                                        ;XXXXXXXXXX
                                        ;PT5--  SINGLE LINE FEED TEST --  TESTS THE LINE FEED
                                        ;       CAPABILITY FROM ALL COLUMNS.
                                        ;XXXXXXXXXX

010464  000005                  PT5:    5                       ;TEST NUMBER
010466  010670                          PT6                     ;NEXT TEST
010470  104016                          PRTHDR                  ;TYPE HEADER
010472  013701  000652          1$:     MOV     WIDTH,R1        ;COLUMN COUNT TO R1
010476  005741                          TST     -(R1)           ;DECREASE BY 2
010500  012700  000060                  MOV     #60,R0          ;'0' TO R0
010504  104015                  2$:     PRINTC                  ;SEND 0
010506  005301                          DEC     R1              ;DECREMENT COLUMN COUNTER
010510  001375                          BNE     2$              ;BRANCH IF NOT ZERO
010512  012700  000062                  MOV     #62,R0          ;SEND A 2
010516  104015                          PRINTC
010520  104015                          PRINTC                  ;SEND A SECOND TWO
010522  023727  000652  000204         CMP     WIDTH,#132.     ;COMPARE COLUMN COUNT
010530  001404                          BEQ     3$              ;BRANCH IF EQ 132
010532  012700  003410                  MOV     #3410,R0        ;DELAY 1.8 SEC
010536  104010                          DELAY
010540  000407                          BR      5$
010542  012700  000063          3$:     MOV     #63,R0          ;3'S TO R0
010546  012701  000100                  MOV     #100,R1         ;64 TO COUNTER
010552  104015                  4$:     PRINTC                  ;SEND CHARACTER
010554  005301                          DEC     R1              ;DECREMENT COUNT
010556  001375                          BNE     4$              ;BRANCH IF NOT ZERO
010560  104012                  5$:     CRLF                    ;SEND A CR,LF
010562  013701  000652                  MOV     WIDTH,R1        ;NO. COLUMNS TO R1
010566  012700  000134          6$:     MOV     #134,R0         ;BACKSLASH TO R0
010572  104015                          PRINTC                  ;SEND IT
010574  104014                          LF                      ;PRINT LF
010576  005301                          DEC     R1              ;DECREMENT COUNTER
010600  001372                          BNE     6$              ;BRANCH IF NOT ZERO.
010602  104022                          CR                      ;SEND CR
010604  004737  010632                  JSR     PC,PT5AL        ;SEND REF LINE #1
010610  104012                          CRLF                    ;SEND A CR,LF
010612  012700  001750                  MOV     #1750,R0        ;DELAY 1 SEC
010616  104010                          DELAY
010620  004737  010632                  JSR     PC,PT5AL        ;SEND A SECOND REF. LINE
010624  104012                          CRLF                    ;SEND A CR,LF
010626  104005                          CHAIN                   ;CHAIN TO NEXT TEST
010630  000720                          BR      1$              ;REPEAT TEST
010632  013701  000652          PT5AL:  MOV     WIDTH,R1        ;COLUMN COUNT TO R1
010636  012700  000061                  MOV     #61,R0          ;"1" TO R0
010642  104015                  1$:     PRINTC                  ;PRINT R0
010644  005301                          DEC     R1              ;DECREMENT COUNTER
010646  001407                          BEQ     2$              ;BRANCH IF=0
010650  005200                          INC     R0              ;INCREMENT CHARACTER
010652  020027  000071                  CMP     R0,#71          ;COMP CHAR TO "9"
010656  101771                          BLOS    1$              ;BRANCH IF LOWER OR SAME
010660  012700  000060                  MOV     #60,R0          ;RESET CHAR TO "0"
010664  000766                          BR      1$              ;CONTINUE
010666  000207                  2$:     RTS     PC              ;FINISHED, RETURN TO CALLER
```

```
                                        ;XXXXXXXXXX
                                        ;PT6--  BACKSPACE TEST -- A REFERENCE LINE SUCH AS IN
                                        ;       TEST PT5 IS PRINTED.  THE SECOND LINE CONSISTS
                                        ;       OF PRINTING A BACKSLASH, BACKSPACE AND FORWARD
                                        ;       SLASH COMBINATION OUT TO THE GIVEN COLUMN WIDTH.
                                        ;       THIS LINE IS THEN FOLLOWED BY THE SAME TWO REFERENCE
                                        ;       LINES AS PRINTED IN TEST PT5.
                                        ;XXXXXXXXXX

010670  000006                  PT6:    6                       ;TEST NUMBER
010672  011056                          PT7                     ;NEX" TEST
010674  104016                          PRTHDR                  ;PRINT HEADER
010676  104007                          TYPEM                   ;PRINT COLUMN # MESG
010700  014627                          HDR0
010702  013701  000652          1$:     MOV     WIDTH,R1        ;COLUMN COUNT TO R1
010706  005741                          TST     -(R1)           ;DECREMENT BY 2
010710  012700  000060                  MOV     #60,R0          ;"0" TO R0
010714  104015                  2$:     PRINTC                  ;SEND 0
010716  005301                          DEC     R1              ;DECREMENT COLUMN COUNTER
010720  001375                          BNE     2$              ;BRANCH IF NOT ZERO
010722  012700  000062                  MOV     #62,R0          ;"2" TO R0
010726  104015                          PRINTC                  ;SEND A "2"
010730  104015                          PRINTC                  ;SEND A SECOND "2"
010732  023727  000652  000204         CMP     WIDTH,#132.     ;COMPARE COLUMN COUNT
010740  001404                          BEQ     3$
010742  012700  003410                  MOV     #3410,R0        ;DELAY 1.8 SEC
010746  104010                          DELAY
010750  000407                          BR      5$
010752  012700  000063          3$:     MOV     #63,R0          ;3'S TO R0
010756  012701  000100                  MOV     #100,R1         ;64 TO CHAR COUNT
010762  104015                  4$:     PRINTC                  ;SEND CHAR
010764  005301                          DEC     R1              ;DECREMENT CHAR COUNT
010766  001375                          BNE     4$              ;CONTINUE IF NOT DONE
010770  104012                  5$:     CRLF                    ;SEND A CR,LF
010772  013701  000652                  MOV     WIDTH,R1        ;COLUMN COUNT TO R1
010776  012700  000134          6$:     MOV     #134,R0         ;BACKSLASH TO R0
011002  104015                          PRINTC                  ;SEND IT
011004  012700  000010                  MOV     #10,R0          ;BACKSPACE TO R0
011010  104015                          PRINTC                  ;SEND IT
011012  012700  000057                  MOV     #57,R0          ;FORWORD SLASH TO R0
011016  104015                          PRINTC                  ;SEND IT
011020  005301                          DEC     R1              ;END OF PAPER
011022  001365                          BNE     6$              ;BRANCH IF NO
011024  104014                          LF                      ;SEND LF
011026  104022                          CR                      ;SEND CR
011030  004737  010632                  JSR     PC,PT5AL        ;SEND REF LINE #1
011034  104012                          CRLF                    ;SEND A CR,LF
011036  012700  001750                  MOV     #1750,R0        ;DELAY 1 SEC
011042  104010                          DELAY
011044  004737  010632                  JSR     PC,PT5AL        ;SEND SECOND REF LINE
011050  104012                          CRLF                    ;SEND A CR,LF
011052  104005                          CHAIN                   ;CHAIN TO NEXT TEST
011054  000712                          BR      1$              ;REPEAT TEST
```

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

011056  000007          PT7:    7                       ;TEST NUMBER
011060  011270                  PT10                    ;NEXT TEST
011062  104016                  PRTHDR                  ;PRINT MESSAGE
011064  012703  000002  1$:     MOV     #2,R3           ;2 COUNT TO R3
011070  013701  000652  2$:     MOV     WIDTH,R1        ;NO. OF COLUMNS TO R1
011074  012700  000115  3$:     MOV     #115,R0         ;"M" TO R0
011100  104015                  PRINTC                  ;SEND IT
011102  005301                  DEC     R1              ;END OF LINE
011104  001404                  BEQ     4$              ;BRANCH IF YES
011106  004737  007562          JSR     PC,SPC          ;SEND SPACE
011112  005301                  DEC     R1              ;END OF LINE?
011114  001367                  BNE     3$              ;BRANCH IF NO
011116  022703  000002  4$:     CMP     #2,R3           ;TEST R3
011122  001003                  BNE     6$              ;BRANCH IF NOT FIRST TIME
011124  104022          5$:     CR                      ;SEND CR
011126  005303                  DEC     R3              ;DECREASE LINE COUNTER
011130  000757                  BR      2$              ;REPEAT LINE
011132  005703          6$:     TST     R3              ;THIRD TIME?
011134  001373                  BNE     5$              ;BRANCH IF NOT
011136  104012                  CRLF                    ;NEXT LINE
011140  005723                  TST     (R3)+           ;REPEAT COUNTER TO R3
011142  013701  000652  7$:     MOV     WIDTH,R1        ;COLUMN COUNT TO R1
011146  004737  007562  8$:     JSR     PC,SPC          ;SEND SPACE
011152  005301                  DEC     R1              ;DECREASE COLUMN COUNT
011154  001405                  BEQ     9$              ;BRANCH IF 0, END OF LINE
011156  012700  000100          MOV     #100,R0         ;"@" TO R0
011162  104015                  PRINTC                  ;SEND IT
011164  005301                  DEC     R1              ;DECREASE COLUMN COUNT
011166  001367                  BNE     8$              ;BRANCH IF NOT 0 (NOT END)
011170  022703  000002  9$:     CMP     #2,R3           ;END OF LINE, FIRST TIME?
011174  001003                  BNE     11$             ;BRANCH IF NOT
011176  104022          10$:    CR                      ;SEND CR
011200  005303                  DEC     R3              ;DECREASE LINE COUNTER
011202  000757                  BR      7$              ;REPEAT LINE
011204  005703          11$:    TST     R3              ;TEST IF THIRD REPEAT
011206  001373                  BNE     10$             ;BRANCH IF NOT
011210  104012                  CRLF                    ;DO NEXT LINE
011212  005723                  TST     (R3)+           ;LINE REPEAT COUNTER TO R3
011214  013701  000652  12$:    MOV     WIDTH,R1        ;COLUMN COUNT TO R1
011220  012700  000046  13$:    MOV     #46,R0          ;"&" TO R0
011224  104015                  PRINTC                  ;SEND IT
011226  005301                  DEC     R1              ;DECREASE COLUMN COUNT

011230  001404                  BEQ     14$             ;BRANCH IF END
011232  004737  007562          JSR     PC,SPC          ;SEND SPACE
```

```
 3314  011236  005301                    DEC     R1              ;DECREASE COLUMN COUNT
 3315  011240  001367                    BNE     13$             ;BRANCH IF NOT END
 3316  011242  022703  000002    14$:    CMP     #2,R3           ;TEST IF FIRST TIME
 3317  011246  001003                    BNE     16$             ;BRANCH IF =2, FIRST TIME
 3318  011250  104022            15$:    CR                      ;SENT CR
 3319  011252  005303                    DEC     R3              ;DECREASE REPEAT COUNTER
 3320  011254  000757                    BR      12$             ;PRINT LINE AGAIN
 3321  011256  005703            16$:    TST     R3              ;TEST IF END, R3=0
 3322  011260  001373                    BNE     15$             ;BRANCH IF NOT END
 3323  011262  104012                    CRLF                    ;SEND CR,LF
 3324  011264  104005                    CHAIN                   ;CHAIN TO NEXT TEST
 3325  011266  000676                    BR      1$              ;REPEAT TEST
```

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

011270  000010          PT10:   10                      ;TEST NUMBER
011272  011426                  PT11                    ;NEXT TEST
011274  104016                  PRTHDR                  ;TYPE MESSAGE
011276  012701  000036  1$:     MOV     #36,R1          ;SET R1=30
011302  012702  000170          MOV     #120.,R2        ;SET CHAR COUNT - 120
011306  012737  000036  011324  MOV     #30.,3$+2       ;SET UP DELAY VALUE
011314  012700  000110  2$:     MOV     #110,R0         ;"H" TO R0
011320  104015                  PRINTC                  ;SEND IT
011322  012700  000036  3$:     MOV     #30.,R0
011326  104010                  DELAY                   ;DELAY
011330  005301                  DEC     R1              ;DEC. COUNT OF CHARS PER LINE
011332  001426                  BEQ     6$              ;BRANCH IF 0, END OF LINE
011334  005302          4$:     DEC     R2              ;DECREMENT CHAR COUNTER
011336  001430                  BEQ     7$              ;BRANCH IF END
011340  013704  011324          MOV     3$+2,R4         ;GET OLD DELAY
011344  006204                  ASR     R4              ;CAL 1/16 OF OLD DELAY
011346  006204                  ASR     R4
011350  006204                  ASR     R4
011352  006204                  ASR     R4
011354  010405                  MOV     R4,R5           ;SAVE 1/16 IN R5
011356  006204                  ASR     R4              ;CAL 1/128 OF OLD DELAY
011360  006204                  ASR     R4
011362  006204                  ASR     R4
011364  060405                  ADD     R4,R5           ;1/16 +1/128 TO R5
011366  022702  000074          CMP     #60.,R2         ;TEST WHICH HALF OF THE 120 CHARS.
011372  003403                  BLE     5$              ;BRANCH IF LT OR EQ 60
011374  160537  011324          SUB     R5,3$+2         ;GT 51, DECREASE DELAY BY 34 MEC.
011400  000745                  BR      2$              ;GO PRINT AGAIN
011402  060537  011324  5$:     ADD     R5,3$+2         ;LT HALF WAY, ADD DELAY OF 34 MEC.
011406  000742                  BR      2$              ;GO PRINT AGAIN
011410  104012          6$:     CRLF                    ;SEND CRLF
011412  012701  000036          MOV     #36,R1          ;SET R1=30
011416  000746                  BR      4$
011420  104012          7$:     CRLF                    ;SEND CR,LF
011422  104005                  CHAIN                   ;CHAIN TO NEXT TEST
011424  000724                  BR      1$              ;REPEAT TEST
```

```
 3375                                   ;XXXXXXXXXX
 3376                                   ;
 3377                                   ;PT11-- RIBBON FEED TEST-- THIS TEST PRINTS A SINGLE COLUMN OF X'S
 3378                                   ;       (24 LINES) DOWN THE LEFT MARGIN OF THE PAGE.
 3379                                   ;       VISUALLY CHECK THE RIBBON FEED MECHANISM FOR PROPER OPERATION.
 3380                                   ;
 3381                                   ;XXXXXXXXXX
 3382
 3383
 3384  011426  000011                  PT11:   11                      ;TEST NUMBER
 3385  011430  011460                          PT12                    ;NEXT TEST
 3386  011432  104016                          PRTHDR                  ;TYPE MESSAGE
 3387  011434  012701  000030          1$:     MOV     #30,R1          ;SET R1=24(10), LINE COUNT
 3388  011440  012700  000130          2$:     MOV     #130,R0         ;SET CHAR = X
 3389  011444  104015                          PRINTC                  ;PRINT X
 3390  011446  104012                          CRLF                    ;SEND CR-LF
 3391  011450  005301                          DEC     R1              ;DECREMENT LINE COUNT
 3392  011452  001372                          BNE     2$              ;CONTINUE IF NOT DONE TEST
 3393  011454  104005                          CHAIN                   ;CHAIN TO NEXT TEST
 3394  011456  000766                          BR      1$              ;REPEAT TEST
 3395
 3396
 3397
 3398                                   ;XXXXXXXXXX
 3399                                   ;
 3400                                   ;PT12-- PRINTER BELL TEST-- THE LAST TEST IN THE
 3401                                   ;       PRINTER TEST SEQUENCE.  THIS TEST OUTPUTS
 3402                                   ;       EIGHT BELL SIGNALS TO THE PRINTER
 3403                                   ;
 3404                                   ;
 3405                                   ;XXXXXXXXXX
 3406                                   ;
 3407  011460  000012                  PT12:   12                      ;THIS TEST
 3408  011462  007374                          PTO                     ;NEXT TEST
 3409  011464  104016                          PRTHDR                  ;TYPE HEADER
 3410  011466  012701  000010          PT12A:  MOV     #10,R1          ;COUNTER TO R1
 3411  011472  012700  000007                  MOV     #7,R0           ;BELL TO R0
 3412  011476  104015                  1$:     PRINTC                  ;SEND IT
 3413  011500  005301                          DEC     R1              ;DECREMENT COUNT
 3414  011502  001375                          BNE     1$              ;BRANCH IF NOT ZERO
 3415  011504  104014                          LF
 3416  011506  012700  003720                  MOV     #3720,R0        ;DELAY 2 SEC BEFORE RESTARTING
 3417  011512  104010                          DELAY
 3418  011514  013700  000042                  MOV     @#42,R0         ;CHECK IF UNDER ACT11 OR XXDP
 3419  011520  001405                          BEQ     HERE            ;CONTINUE TEST SEQUENCE
 3420  011522  000240                          NOP                     ;A RESET WAS FORMERLY HERE
 3421  011524  004710                  LOGICAL:JSR     PC,(R0)
 3422  011526  000240                          NOP
 3423  011530  000240                          NOP
 3424  011532  000240                          NOP
 3425  011534  104005                  HERE:   CHAIN                   ;CHAIN TO NEXT TEST
 3426  011536  000753                          BR      PT12A           ;REPEAT TEST
```

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

011540  000017          PT17B:  17                      ;TEST NUMBER
011542  011540                  PT17B                   ;NEXT TEST
011544  000137  011600          JMP     PT17D           ;CONTINUE
011550  000017          PT17:   17                      ;TEST NUMBER
011552  011540                  PT17B                   ;NEXT TEST
011554  005037  012116          CLR     PASCNT          ;CLEAR PASS COUNT
011560  013704  000652          MOV     WIDTH,R4        ;INITIALIZE R4
011564  012737  000001  012114  MOV     #1,DIRTN        ;AND DIRECTION OF PRECESS
011572  104016                  PRTHDR
011574  104007                  TYPEM                   ;PRINT COLUMN # MESG
011576  014627                  HDR0
011600  012703  000041  PT17D:  MOV     #41,R3          ;SET START CHAR
011604  005237  012116          INC     PASCNT
011610  023727  012116  000031  CMP     PASCNT,#31      ;DO 31 TIMES
011616  001003                  BNE     20$             ;BRANCH IF NOT DONE
011620  012737  000001  012116  MOV     #1,PASCNT       ;START OVER
011626  012700  014542  20$:    MOV     #PASMES,R0      ;SET MESG ADDR
011632  013701  012116          MOV     PASCNT,R1       ;# TO CONVERT
011636  012702  000002          MOV     #2,R2           ;# DIGITS
011642  104023                  BTOASC                  ;CONVERT PASCNT TO ASCII
011644  013701  000652  1$:     MOV     WIDTH,R1        ;SET COLUMN COUNT
011650  010300          2$:     MOV     R3,R0           ;GET CHARACTER
011652  004737  011766          JSR     PC,CKPOS        ;TIME TO INSERT PASS # ?
011656  104015                  PRINTC                  ;SEND CHAR
011660  005301                  DEC     R1              ;DECREMENT COUNT
011662  003372                  BGT     2$              ;BRANCH IF NOT DONE
011664  004737  012034          JSR     PC,ADJR4        ;ADJUST R4 POINTER
011670  104012                  CRLF
011672  012702  000005          MOV     #5,R2           ;SET OVERPRINT COUNT
011676  013701  000652  3$:     MOV     WIDTH,R1        ;SET COLUMN COUNT
011702  010300          4$:     MOV     R3,R0           ;GET CHARACTER
011704  004737  011766          JSR     PC,CKPOS        ;TIME TO INSERT PASS # ?
011710  104015                  PRINTC                  ;SEND CHAR
011712  005301                  DEC     R1              ;DECREMENT COUNT
011714  003372                  BGT     4$              ;BRANCH IF NOT DONE
011716  104022                  CR                      ;SEND CR
011720  005302                  DEC     R2              ;DONE OVERPRINTS ?
011722  001365                  BNE     3$              ;NO. CONTINUE
011724  004737  012034          JSR     PC,ADJR4        ;ADJUST R4 POINTER
011730  104014                  LF                      ;SEND LF
011732  005203                  INC     R3              ;SET NEXT CHAR
011734  022703  000177          CMP     #177,R3         ;DONE CHAR SET ?
011740  001341                  BNE     1$              ;NO. CONTINUE
011742  004737  012034          JSR     PC,ADJR4        ;OFFSET POINTER 3 PLACES
011746  004737  012034          JSR     PC,ADJR4        ;TO RETAIN VISUAL ALIGNMENT
```

```
 3483  011752  004737  012034          JSR     PC,ADJR4        ;THROUGH END OF PASS
 3484  011756  104007                  TYPEM                   ;TYPE END OF PASS MESG
 3485  011760  014523                  ENDPAS
 3486  011762  104005                  CHAIN                   ;REPEAT TEST
 3487  011764  000705                  BR      P117D
 3488
```

```
3489
3490  011766  020401                  CKPOS:  CMP     R4,R1           ;IS IT TIME TO INSERT PASS # ?
3491  011770  001020                          BNE     1$              ;BRANCH IF NO
3492  011772  012700  000040                  MOV     #40,R0          ;PRINT A SPACE
3493  011776  104015                          PRINTC
3494  012000  113700  014542                  MOVB    PASMES,R0       ;PRINT MSG OF PASS COUNT
3495  012004  104015                          PRINTC
3496  012006  113700  014543                  MOVB    PASMES+1,R0
3497  012012  104015                          PRINTC
3498  012014  012700  000040                  MOV     #40,R0          ;PRINT A SPACE
3499  012020  104015                          PRINTC
3500  012022  162701  000003                  SUB     #3,R1           ;ADJUST R1 3 POSITIONS
3501  012026  062716  000002                  ADD     #2,(SP)         ;ADJUST RETURN PC OVER PRINTC
3502  012032  000207                  1$:     RTS     PC
3503
3504  012034  005737  012114          ADJR4:  TST     DIRTN           ;TEST DIRECTION OF PRECESS
3505  012040  001013                          BNE     1$              ;BR IF LEFT
3506  012042  005204                          INC     R4              ;INCREASE POSITION CNTR
3507  012044  020437  000652                  CMP     R4,WIDTH        ;IS R4 > WIDTH ?
3508  012050  101420                          BLOS    3$              ;BR IF NOT GREATER
3509  012052  013704  000652                  MOV     WIDTH,R4        ;CHANGE DIRECTION
3510  012056  005304                          DEC     R4              ;       TO
3511  012060  012737  000001  012114          MOV     #1,DIRTN        ;     LEFT.
3512  012066  000411                          BR      3$
3513  012070  005304                  1$:     DEC     R4              ;DECREASE POSITION CNTR
3514  012072  020427  000004                  CMP     R4,#4   ;LESS THAN 4 ?
3515  012076  002401                          BLT     2$              ;BR IF YES
3516  012100  000404                          BR      3$              ;ELSE EXIT
3517  012102  012704  000005          2$:     MOV     #5,R4           ;SET R4 TO POS 5
3518  012106  005037  012114                  CLR     DIRTN           ;CHANGE DIRECTION TO RIGHT
3519  012112  000207                  3$:     RTS     PC              ;EXIT
3520
3521  012114  000000                  DIRTN:  .WORD   0               ;DIRECTION OF PRECESS (0 LEFT)
3522
3523  012116  000000                  PASCNT: .WORD   0
```

3524

```
                                        .SBTTL  LA36 ECHO TESTS
                                ;
                                ;XXXXXXXXXX
                                ;
                                ;E020-- CHARACTER ECHO TEST-- ALL PRINTABLE AND
                                ;       NON-PRINTING CHARACTERS TYPED ON THE KEYBOARD
                                ;       ARE USED TO DRIVE THE PRINTER, ONE CHARACTER AT
                                ;       A TIME.  A "RUBOUT" WILL CAUSE THE TEST TO BE
                                ;       TERMINATED.
                                ;
                                ;XXXXXXXXXXX
                                ;
012120  000020                  E020:   20                      ;TEST NUMBER
012122  012170                          E021                    ;NEXT TEST
012124  104016                          PRTHDR                  ;TYPE HEADER
012126  104020                  1$:     READ                    ;GO WAIT FOR KEYBOARD INPUT
012130  012700  0C0036                  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
012134  104010                          DELAY
012136  022737  000177  000700          CMP     #177,TEMPCH     ;CHECK IF RUBOUT
012144  001405                          BEQ     2$              ;BRANCH IF YES
012146  104017                          PRNT                    ;NO, CHECK PRINTER READY
012150  117777  166440  166442          MOVB    @TKB,@TPB       ;READY, ECHO CHARACTER
012156  000763                          BR      1$
012160  104007                  2$:     TYPEM                   ;PRINT TERMINATION MESSAGE
012162  014772                          ECOEND
012164  104005                          CHAIN                   ;CHAIN TO NEXT TEST
012166  000757                          BR      1$              ;REPEAT TEST
                                ;
                                ;XXXXXXXXXX
                                ;
                                ;E021-- LINE ECHO TEST, FAST RATE-- THIS TEST WILL
                                ;       CAUSE THE CONTINUAL PRINTING OF "0" AT THE MAXIMUM
                                ;       RATE UNTIL EITHER ANOTHER CHARACTER IS SELECTED
                                ;       BY PRESSING A KEY ON THE KEYBOARD OR TERMINATION BY THE
                                ;       RUBOUT.
                                ;
                                ;XXXXXXXXXX
                                ;
012170  000021                  E021:   21                      ;TEST NUMBER
012172  012226                          E022                    ;NEXT TEST
012174  104016                          PRTHDR                  ;TYPE HEADER
012176  012737  000060  000662  E021A:  MOV     #60,REPT        ;CHARACTER TO BE REPEATED (0)
012204  013702  000652          1$:     MOV     WIDTH,R2        ;SET COLUMN COUNT
012210  013700  000662          2$:     MOV     REPT,R0         ;GET CHAR
012214  104015                          PRINTC                  ;PRINT CHAR
012216  005302                          DEC     R2              ;DEC COLUMN COUNT
012220  003373                          BGT     2$              ;FINISH LINE
012222  104012                          CRLF                    ;SEND A CR AND LF
012224  000767                          BR      1$
```

```
3574                                    ;XXXXXXXXXX
3575                                    ;
3576                                    ;E022-- LINE ECHO TEST, SLOW RATE-- SAME AS E021 EXCEPT
3577                                    ;       THAT A DELAY IS INTRODUCED BETWEEN CHARACTERS
3578                                    ;       TO PRODUCE A LCV ACTION
3579                                    ;
3580                                    ;XXXXXXXXXX
3581                                    ;
3582  012226  000022                    E022:   22
3583  012230  012500                            E023
3584  012232  104016                            PRTHDR                  ;TYPE HEADER
3585  012234  012737  000060  000662    E022A:  MOV     #60,REPT        ;LOAD 0 AS INITIAL CHARACTER
3586  012242  013702  000652            1$:     MOV     WIDTH,R2        ;SET COLUMN COUNT
3587  012246  013700  000662            2$:     MOV     REPT,R0         ;GET CHAR
3588  012252  104015                            PRINTC                  ;PRINT CHAR
3589  012254  005302                            DEC     R2              ;DEC COLUMN COUNT
3590  012256  001404                            BEQ     3$              ;BRANCH IF DONE LINE
3591  012260  012700  003410                    MOV     #3410,R0
3592  012264  104010                            DELAY                   ;DELAY 1.8 SEC.
3593  012266  000767                            BR      2$              ;OUTPUT NEW CHAR.
3594  012270  104012                    3$:     CRLF                    ;SEND A CR AND LF
3595  012272  000763                            BR      1$
```

```
 3596                                   ;**********
 3597                                   ;
 3598                                   ;       THIS FOLLOWING TABLE IS USED BY TEST E023
 3599                                   ;
 3600                                   ;**********
 3601
 3602  012274  052516  020114   MONIC:  .ASCII  /NUL /
 3603  012300  047523  020110           .ASCII  /SOH /
 3604  012304  052123  020130           .ASCII  /STX /
 3605  012310  052105  020130           .ASCII  /ETX /
 3606  012314  047505  020124           .ASCII  /EOT /
 3607  012320  047105  020121           .ASCII  /ENQ /
 3608  012324  041501  020113           .ASCII  /ACK /
 3609  012330  042502  020114           .ASCII  /BEL /
 3610  012334  051502  020040           .ASCII  /BS  /
 3611  012340  052110  020040           .ASCII  /HT  /
 3612  012344  043114  020040           .ASCII  /LF  /
 3613  012350  052126  020040           .ASCII  /VT  /
 3614  012354  043106  020040           .ASCII  /FF  /
 3615  012360  051103  020040           .ASCII  /CR  /
 3616  012364  047523  020040           .ASCII  /SO  /
 3617  012370  044523  020040           .ASCII  /SI  /
 3618  012374  046104  020105           .ASCII  /DLE /
 3619  012400  041504  020061           .ASCII  /DC1 /
 3620  012404  041504  020062           .ASCII  /DC2 /
 3621  012410  041504  020063           .ASCII  /DC3 /
 3622  012414  041504  020064           .ASCII  /DC4 /
 3623  012420  040516  020113           .ASCII  /NAK /
 3624  012424  054523  020116           .ASCII  /SYN /
 3625  012430  052105  020102           .ASCII  /ETB /
 3626  012434  040503  020116           .ASCII  /CAN /
 3627  012440  046505  020040           .ASCII  /EM  /
 3628  012444  052523  020102           .ASCII  /SUB /
 3629  012450  051505  020103           .ASCII  /ESC /
 3630  012454  051506  020040           .ASCII  /FS  /
 3631  012460  051507  020040           .ASCII  /GS  /
 3632  012464  051522  020040           .ASCII  /RS  /
 3633  012470  051525  020040           .ASCII  /US  /
 3634  012474  050123  020040           .ASCII  /SP  /
 3635
 3636                                   .EVEN
```

```
                                        ;XXXXXXXXXX
                                        ;
                                        ;E023-- CHARACTER CODE TEST-- ANY CHARACTER SELECTED
                                        ;       WILL BE ECHOED ALONG WITH ITS OCTAL CODE.
                                        ;       A MNEMONIC WILL BE PRINTED INSTEAD OF THE CHARACTER
                                        ;       IF IT IS A NON-PRINTING CHARACTER.
                                        ;       THE PARITY OF THE RECEIVED CODE WILL ALSO BE
                                        ;       INDICATED AS EITHER EVEN OR ODD.
                                        ;
                                        ;XXXXXXXXXX
                                        ;
012500  000023                  E023:   23                      ;TEST NUMBER
012502  013022                          E024                    ;NEXT TEST
012504  104016                          PRTHDR                  ;TYPE HEADER
012506  104020                  1$:     READ                    ;GO WAIT FOR CHARACTER
012510  012700  000036                  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
012514  104010                          DELAY
012516  023727  000700  000041          CMP     TEMPCH,#41      ;TEST IF CHAR IS PRINTABLE
012524  103015                          BHIS    3$              ;BRANCH IF IT IS
012526  004737  012662                  JSR     PC,STRLN        ;STORE CODE INTO MESSAGE
012532  113700  000700                  MOVB    TEMPCH,R0       ;GET CODE AGAIN
012536  006300                          ASL     R0              ;MULT BY 2
012540  006300                          ASL     R0              ;MULT BY 4
012542  062700  012274                  ADD     #MONIC,R0       ;ADD ADDR OF MNEMONIC TABLE
012546  004737  012740                  JSR     PC,MOVNUM       ;MOV MNEMONIC TO MESSAGE
012552  104000                  2$:     TYPE                    ;TYPE CODE AND MNEMONIC
012554  015021                          E023M                   ;ADDRESS OF MESSAGE
012556  000753                          BR      1$              ;GO WAIT FOR NEXT CHARACTER
012560  023727  000700  000177  3$:     CMP     TEMPCH,#177     ;TEST IF CHAR IS A RUBOUT
012566  001421                          BEQ     4$              ;BRANCH IF RUBOUT
012570  012701  013012                  MOV     #MG24,R1
012574  113721  000700                  MOVB    TEMPCH,(R1)+
012600  112721  000040                  MOVB    #40,(R1)+
012604  112721  000040                  MOVB    #40,(R1)+
012610  112721  000040                  MOVB    #40,(R1)+
012614  004737  012662                  JSR     PC,STRLN        ;STORE CODE INTO MESSAGE
012620  012700  013012                  MOV     #MG24,R0        ;ADDR OF CHAR INTO R0
012624  004737  012740                  JSR     PC,MOVNUM       ;MOVE CHAR INTO MESSAGE
012630  000750                          BR      2$              ;TYPE MESSAGE
012632  004737  012662          4$:     JSR     PC,STRLN        ;RUBOUT, CONVERT AND STOR CODE
012636  012700  013016                  MOV     #MG25,R0        ;ADDR. OF DEL INTO R0
012642  004737  012740                  JSR     PC,MOVNUM       ;MOVE DEL INTO MESSAGE
012646  104000                          TYPE                    ;TYPE MESSAGE
012650  015021                          E023M                   ;ADDR OF MESSAGE
012652  104007                          TYPEM
012654  014772                          ECOEND
012656  104005                          CHAIN                   ;CHAIN TO NEXT TEST
012660  000712                          BR      1$              ;REPEAT TEST
012662  012702  000003          STRLN:  MOV     #3,R2           ;COUNT OF 3 TO R2
012666  012701  015023                  MOV     #LINE5,R1       ;ADDR OF MESG TO R1
012672  062701  000003                  ADD     #3,R1           ;POINT TO LAST SPACE IN MESG
012676  013700  000704          1$:     MOV     PCHAR,R0        ;MOVE OCTAL CODE TO R0
012702  042700  177770                  BIC     #177770,R0      ;SAVE LS OCTAL CHAR
012706  062700  000060                  ADD     #60,R0          ;MAKE ASCII
012712  110041                          MOVB    R0,-(R1)        ;MOVE INTO MESG
012714  005302                          DEC     R2              ;DECREMENT CHAR COUNTER
```

```
3693  012716  001407                  BEQ     2$              ;BRANCH IF 3 MOVED
3694  012720  006237  000704          ASR     PCHAR           ;NOT THREE, SHIFT NEXT OCTAL
3695  012724  006237  000704          ASR     PCHAR           ;CHARACTER TO THE RIGHT
3696  012730  006237  000704          ASR     PCHAR           ;
3697  012734  000760                  BR      1$              ;CONVERT AND STORE NEXT CHAR
3698  012736  000207          2$:     RTS     PC              ;RETURN TO CALLER
3699  012740  012701  015027  MOVNUM: MOV     #LINE5A,R1      ;ADDR OF LINES IN R1
3700  012744  012702  000004          MOV     #4,R2           ;COUNT OF 4 TO R2
3701  012750  112021          1$:     MOVB    (R0)+,(R1)+     ;MOV 4 CHARS TO MESG AREA
3702  012752  005302                  DEC     R2              ;DECREMENT COUNTER
3703  012754  001375                  BNE     1$              ;BRANCH IF NOT ALL DONE
3704  012756  105737  000702          TSTB    PARITY          ;TEST PARITY FLAG
3705  012762  001003                  BNE     2$              ;BRANCH IF ODD PARITY
3706  012764  012700  015071          MOV     #EVEN,R0        ;SET ADDRESS FOR EVEN PARITY MESG
3707  012770  000402                  BR      3$              ;CONTINUE
3708  012772  012700  015075  2$:     MOV     #ODD,R0         ;SET ADDRESS FOR ODD PARITY MESG
3709  012776  012702  000004  3$:     MOV     #4,R2           ;COUNT OF 4 TO R2
3710  013002  112021          4$:     MOVB    (R0)+,(R1)+     ;MOVE 4 CHARS TO MESG AREA
3711  013004  005302                  DEC     R2              ;DECREMENT COUNTER
3712  013006  001375                  BNE     4$              ;BRANCH IF NOT DONE
3713  013010  000207                  RTS     PC              ;RETURN
3714
3715  013012  020040  020040  MG24:   .ASCII   /    /          ;SAVE CHARACTER CODE
3716
3717                                  .EVEN
3718
3719  013016  042504  020114  MG25:   .ASCII   /DEL /          ;MNEMONIC FOR RUBOUT
3720
3721                                  .EVEN
```

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

013022  000024                  E024:   24                      ;TEST NUMBER
013024  013570                          E025                    ;NEXT TEST
013026  104016                          PRTHDR                  ;TYPE TEST HEADER
013030  005001                  E024B:  CLR     R1              ;CLEAR CHARACTER COUNT
013032  012702  013166                  MOV     #BUFR,R2        ;ADDRESS OF BUFFER TO R2
013036  104020                  1$:     READ                    ;WAIT FOR INPUT
013040  012700  000036                  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
013044  104010                          DELAY
013046  022737  000177  000700          CMP     #177,TEMPCH     ;TEST IF RUBOUT
013054  001440                          BEQ     TERM            ;BRANCH IF RUBOUT
013056  022737  000003  000700          CMP     #3,TEMPCH       ;TEST IF CNTL-C
013064  001413                          BEQ     OUTPUT          ;BRANCH IF CNTL-C
013066  020127  000400                  CMP     R1,#256.        ;YES, CHECK IF CHAR CNT IS EQ, GT 256
013072  103361                          BHIS    1$              ;BRANCH IF YES, IGNORE CHAR
013074  113722  000700                  MOVB    TEMPCH,(R2)+    ;STORE CHAR INTO BUFFER
013100  005201                          INC     R1              ;INCREMENT CHARACTER COUNT
013102  104017                          PRNT                    ;CHECK IF PRINTER READY
013104  113777  000700  165506          MOVB    TEMPCH,@TPB     ;ECHO CHAR
013112  000751                          BR      1$              ;GO WAIT FOR NEXT CHAR
                                        ;
                                        ;SECTION TO OUTPUT CONTINOUS STRING
                                        ;
013114  020227  013166          OUTPUT: CMP     R2,#BUFR        ;CHECK IF POINTER IS AT START OF TABLE
013120  001403                          BEQ     1$              ;YES, BRANCH
013122  113722  000700                  MOVB    TEMPCH,(R2)+    ;NO, STORE ^C IN TABLE
013126  104013                          SCRLF                   ;SEND A CR LF
013130  012702  013166          1$:     MOV     #BUFR,R2        ;BUFFER ADDRESS TO R2
013134  021227  000003                  CMP     (R2),#3         ;CHECK IF FIRST CHAR IS ^C
013140  001733                          BEQ     E024B           ;YES, LOOK FOR INPUT AGAIN
013142  112200                  2$:     MOVB    (R2)+,R0        ;GET CHARACTER
013144  020027  000003                  CMP     R0,#3           ;DONE STRING?
013150  001767                          BEQ     1$              ;YES, RESTART STRING
013152  104015                          PRINTC                  ;PRINT CHAR
013154  000772                          BR      2$              ;CONTINUE
013156  104007                  TERM:   TYPEM                   ;OUTPUT TERMINATION MESSAGE
013160  014772                          ECOEND
013162  104005                          CHAIN                   ;CHAIN TO NEXT TEST
013164  000721                          BR      E024B           ;REPEAT TEST
013166  000003                  BUFR:   3                       ;INITIALIZE FIRST CHAR AS CNTL-C IN TABLE
013170  000400                          .BLKB   256.            ;256 CHARACTER BUFFER
```

```
                                        ;XXXXXXXXXXX
                                        ;
                                        ;E025-- BELL ECHO TEST-- A MESSAGE IS PRINTED AND
                                        ;       THE TEST WAITS FOR SOME PRINTABLE CHARACTER
                                        ;       TO BE SELECTED ON THE KEYBOARD (GT040).  THIS
                                        ;       TEST IS VALID ONLY IF THE PAPER WIDTH IS GT 64
                                        ;       COLUMNS.  IF LT64 COLUMNS AN ILLEGAL BELL TEST
                                        ;       MESSAGE IS PRINTED.
                                        ;
                                        ;XXXXXXXXXXX
                                        ;
013570  000025                  E025:   25                      ;TEST NUMBER
013572  012120                          E020                    ;NEXT TEST HEADER
013574  104016                          PRTHDR                  ;PRINT HEADER
013576  023727  000652  000101  1$:     CMP     WIDTH,#101      ;TEST IF COLUMN COUNTIS EQ,GT 64
013604  103427                          BLO     4$              ;BRANCH IF NOT
013606  104007                          TYPEM                   ;TYPE TEST MESG
013610  014645                          E025MA
013612  000402                          BR      3$              ;WAIT FOR CHAR
013614  104000                  2$:     TYPE                    ;TYPE TEST MESG ON TERM CHAR RCVD ON
013616  014645                          E025MA
013620  104020                  3$:     READ                    ;WAIT FOR OPERATOR RESPONSE
013622  012700  000036                  MOV     #30.,R0         ;DELAY FOR HALF DUPLEX
013626  104010                          DELAY
013630  023727  000700  000040          CMP     TEMPCH,#40      ;TEST IF PRINTABLE
013636  103770                          BLO     3$              ;BRANCH IF NON-PRINTABLE
013640  022737  000177  000700          CMP     #177,TEMPCH     ;CHECK IF CHAR IS RUBOUT
013646  001410                          BEQ     5$              ;BRANCH IF YES
013650  104017                          PRNT                    ;CHECK IF PRINTER IS READY
013652  113777  000700  164740          MOVB    TEMPCH,@TPB     ;PRINT CHAR. (BELL SHOULD SOUND)
013660  104013                          SCRLF                   ;SEND A CRLF
013662  000754                          BR      2$              ;REPEAT
013664  104007                  4$:     TYPEM                   ;TYPE ERROR MESSAGE
013666  014745                          E025MB
013670  104007                  5$:     TYPEM                   ;PRINT TERMINATION
013672  014772                          ECOEND
013674  104005                          CHAIN                   ;EXIT TO NEXT TEST
013676  000737                          BR      1$              ;REPEAT TEST
```

```
                                        ;XXXXXXXXXXXXXX
                                        ;
                                        ;CHECK SWITCH REGISTER ROUTINE.  CHECKS FOR ^G TO ALLOW
                                        ;CHANGING OF LOC 176.
                                        ;
                                        ;XXXXXXXXXXXXXX

                                        ;**************
013700                                  CHGF5:
013700  022737  000176  000714  CKSWR:   CMP     #SWREG,SR       ;SOFTWARE SW REG PRES?
013706  001132                          BNE     OUT             ;NO,GET OUT
013710  105777  164676                  TSTB    @TKS            ;YES, IS CHARACTER READY?
013714  100127                          BPL     OUT             ;IF NOT, GET OUT
013716  017737  164672  000722          MOV     @TKB,TIB        ;STORE BUFFER
013724  042737  177600  000722          BIC     #177600,TIB     ;STRIP OFF GARBAGE
013732  022737  000007  000722          CMP     #7,TIB          ;IS IT A ^G
013740  001115                          BNE     OUT             ;IF NOT GET OUT
013742  104000                          TYPE                    ;ECHO ^G
013744  015200                          CNTG
013746  005077  164646                  CLR     @TPB
013752  104000                  CNTLU:  TYPE                    ;ALLOW SWR- TO BE TYPED
013754  015206                          SWR
013756  017746  164732                  MOV     @SR,-(SP)       ;MOV CONTENTS OF SWR
013762  004737  014214                  JSR     PC,TYPOC        ;OCTAL TYPE OUT ROUTINE
013766  022600                          CMP     (%6)+,R0        ;CORRECT STACK POINTER
013770  104000                          TYPE                    ;ALLOW NEW= TO BE TYPED
013772  015216                          NEW
013774  005037  000724                  CLR     TEMPST          ;CLEAR TEMP STORAGE LOC
014000  012737  000007  000726          MOV     #7,COUNT        ;SET UP TO ACCEPT 7 CHAR
014006  105777  164600          1$:     TSTB    @TKS            ;IS CHARACTER THERE?
014012  100375                          BPL     1$              ;IF NOT,TRY AGAIN
014014  117737  164574  000722          MOVB    @TKB,TIB        ;PICKUP CHARACTER
014022  105777  164570          8$:     TSTB    @TPS            ;CHECK PRINTER STATUS
014026  100375                          BPL     8$              ;NOT READY, TRY AGAIN
014030  113777  000722  164562          MOVB    TIB,@TPB        ;PRINT IT
014036  042737  177600  000722          BIC     #177600,TIB     ;STRIP OFF GARBAGE
014044  122737  000025  000722          CMPB    #25,TIB         ;IS IT A ^U
014052  001001                          BNE     2$              ;BRANCH IF NOT
014054  000736                  3$:     BR      CNTLU           ;START OVER
014056  122737  000015  000722  2$:     CMPB    #15,TIB         ;IS IT A <CR>
014064  001005                          BNE     4$              ;BRANCH IF NOT
014066  104013                          SCRLF
014070  022737  000007  000726          CMP     #7,COUNT        ;WAS <CR> FIRST CHAR
014076  001033                          BNE     7$              ;CHANGE SWREG IF NOT FIRST <CR>
014100  122737  000060  000722  4$:     CMPB    #60,TIB         ;IS IT LESS THAN 0
014106  003004                          BGT     5$              ;GO TO ? ROUTINE IF SO
014110  122737  000067  000722          CMPB    #67,TIB         ;IS IT GREATER THAN 7
014116  002003                          BGE     6$              ;GO TO ? ROUTINE IF SO
014120  104000                  5$:     TYPE                    ;SET UP FOR ? TYPEOUT
014122  015226                          QUEST
014124  000753                          BR      3$              ;START INPUT STRING OVER
014126  006337  000724          6$:     ASL     TEMPST          ;MULTIPLY BY 10
014132  006337  000724                  ASL     TEMPST
014136  006337  000724                  ASL     TEMPST
014142  142737  000060  000722          BICB    #60,TIB         ;CLEAR OFF ASCII
014150  153737  000722  000724          BISB    TIB,TEMPST      ;MOV CHAR TO TEMPST
```

```
 3869  014156  005337  000726                 DEC     COUNT           ;ONLY WANT 6 NUMBERS AND <CR>
 3870  014162  001756                         BEQ     5$              ;IF = 7 TOO MANY NUMBERS
 3871  014164  000710                         BR      1$              ;GET NEXT CHAR
 3872  014166  013777  000724  164520  7$:    MOV     TEMPST,@SR      ;CHANGE SWR CONTENTS
 3873  014174  000207                 OUT:    RTS     PC              ;RETURN TO PROGRAM
 3874  014176  010046                 TTINTS: MOV     R0,-(SP)        ;INTERRUPT SERVICE ROUTINE
 3875  014200  010146                         MOV     R1,-(SP)        ;SAVE R0 AND R1
 3876  014202  004737  013700                 JSR     PC,CKSWR        ;GO TO SUBR TO SERVICE TTY INTERRUPT
 3877  014206  012601                         MOV     (SP)+,R1        ;RESTORE R1 AND R0
 3878  014210  012600                         MOV     (SP)+,R0
 3879  014212  000002                         RTI                     ;RETURN FROM INTERRUPT
 3880
 3881
 3882
 3883  014214  112737  000001  000730  TYPOC:  MOVB    #1,FILL         ;SET THE ZERO FILL      SWITCH
 3884  014222  112737  000006  000733          MOVB    #6,MODE+1       ;SET FOR SIX(6) DIGITS
 3885  014230  112737  000005  000734  TYPON:  MOVB    #5,CNT          ;SET THE ITERATION COUNT
 3886  014236  010346                         MOV     R3,-(SP)        ;SAVE R3
 3887  014240  010446                         MOV     R4,-(SP)        ;SAVE R4
 3888  014242  010546                         MOV     R5,-(SP)        ;SAVE R5
 3889  014244  113704  000733                 MOVB    MODE+1,R4       ;GET THE NUMBER OF DIGITS TO TYPE
 3890  014250  005404                         NEG     R4
 3891  014252  062704  000006                 ADD     #6,R4           ;SUBTRACT IT FOR MAX. ALLOWED
 3892  014256  110437  000732                 MOVB    R4,MODE         ;SAVE IT FOR USE
 3893  014262  113704  000730                 MOVB    FILL,R4         ;GET THE ZERO FILL  SWITCH
 3894  014266  016605  000010                 MOV     10(%6),R5       ;PICKUP THE INPUT NUMBER
 3895  014272  005003                         CLR     R3              ;CLEAR THE OUTPUT WORD
 3896  014274  006105                 1$:     ROL     R5              ;ROTATE MSB INTO "C"
 3897  014276  000404                         BR      3$              ;GO DO MSB
 3898  014300  006105                 2$:     ROL     R5              ;FORM THIS DIGIT
 3899  014302  006105                         ROL     R5
 3900  014304  006105                         ROL     R5
 3901  014306  010503                         MOV     R5,R3
 3902  014310  006103                 3$:     ROL     R3              ;GET LSB OF THIS DIGIT
 3903  014312  105337  000732                 DECB    MODE            ;TYPE THIS DIGIT?
 3904  014316  100015                         BPL     7$              ;BR IF NO
 3905  014320  042703  177770                 BIC     #177770,R3      ;GET RID OF JUNK
 3906  014324  001002                         BNE     4$              ;TEST FOR 0
 3907  014326  005704                         TST     R4              ;SUPPRESS THIS 0
 3908  014330  001403                         BEQ     5$              ;BR IF YES
 3909  014332  005204                 4$:     INC     R4              ;DON'T SUPPRESS ANYMORE 0'S
 3910  014334  052703  000060                 BIS     #60,R3          ;MAKE THIS DIGIT ASCII
 3911  014340  105777  164252         5$:     TSTB    @TPS            ;IS PRINTER READY FOR CHARACTER?
 3912  014344  100375                         BPL     5$              ;IF NOT, TRY AGAIN
 3913  014346  110377  164246                 MOVB    R3,@TPB         ;TYPE OUT NUMBER
 3914  014352  105337  000734         7$:     DECB    CN              ;COUNT BY 1
 3915  014356  003350                         BGT     2$              ;BR IF MORE TO DO
 3916  014360  002402                         BLT     6$              ;BR IF DONE
 3917  014362  005204                         INC     R4              ;INSURE LAST DIGIT ISN'T A BLANK
 3918  014364  000745                         BR      2$              ;GO DO THE LAST DIGIT
 3919  014366  012605                 6$:     MOV     (SP)+,R5        ;RESTORE R5
 3920  014370  012604                         MOV     (SP)+,R4        ;RESTORE R4
 3921  014372  012603                         MOV     (SP)+,R3        ;RESTORE R3
 3922  014374  000207                         RTS     PC              ;RETURN FROM INTERRUPT  PC
 3923                                 ;**************
```

```
 3924                                                   .SBTTL  MISC. DIAGNOSTIC MESSAGES
 3925                                                   ;
 3926  014376  001007  007600  055103  STARTM: .ASCII  <7><2><ACRLF><17>/CZLACFO LA36 TERM (DL11 & KL11)/<ACRLF>
 3927  014404  040514  043103  020060
 3928  014412  040514  033063  052040
 3929  014420  051105  020115  042050
 3930  014426  030514  020061  020046
 3931  014434  046113  030461  100051
 3932  014442  040514  033063  052040          .ASCII  /LA36 TERMINAL DIAGNOSTIC/<ACRLF>
 3933  014450  051105  044515  040516
 3934  014456  020114  044504  043501
 3935  014464  047516  052123  041511
 3936  014472     200
 3937  014473     104  030514  020061          .ASCIZ  /DL11 & KL11 INTERFACE/<ACRLF><12>
 3938  014500  020046  046113  030461
 3939  014506  044440  052116  051105
 3940  014514  040506  042503  005200
 3941  014522     000
 3942  014523     200  042412  042116  ENDPAS: .ASCII  <ACRLF><12>/END OF PASS  /
 3943  014530  047440  020106  040520
 3944  014536  051523  020040
 3945  014542  030060  030060  005200  PASMES: .ASCIZ  /0000/<ACRLF><12>
 3946  014550     000
 3947  014551     200  047503  051516  DL11S:  .ASCII  <ACRLF>/CONSOLE & /
 3948  014556  046117  020105  020046
 3949  014564  030060  042040  030514  DL11S1: .ASCIZ  /00 DL11'S UNDER TEST/<ACRLF><12>
 3950  014572  023461  020123  047125
 3951  014600  042504  020122  042524
 3952  014606  052123  005200     000
 3953  014613     007  100002  005017  HDRMSG: .ASCIZ  <7><2><ACRLF><17><12>/TEST #/
 3954  014620  042524  052123  021440
 3955  014626     000
 3956  014627     060  030060  041440  HDR0:   .ASCIZ  /0C0 COLUMNS/<ACRLF><12>
 3957  014634  046117  046525  051516
 3958  014642  005200     000
 3959  014645     124  050131  020105  E025MA: .ASCII  /TYPE ANY PRINTABLE CHARACTER /
 3960  014652  047101  020131  051120
 3961  014660  047111  040524  046102
 3962  014666  020105  044103  051101
 3963  014674  041501  042524  020122
 3964  014702  047101  020104  044514          .ASCIZ  /AND LISTEN FOR BELL................./
 3965  014710  052123  047105  043040
 3966  014716  051117  041040  046105
 3967  014724  027114  027056  027056
 3968  014732  027056  027056  027056
 3969  014740  027056  027056     000
 3970  014745     200  047516  020124  E025MB: .ASCIZ  <ACRLF>/NOT ENOUGH COLUMNS/<ACRLF>
 3971  014752  047105  052517  044107
 3972  014760  041440  046117  046525
 3973  014766  051516  000200
 3974  014772  042600  044103  020117  ECOEND: .ASCIZ  <ACRLF>/ECHO TEST TERMINATED/<ACRLF>
 3975  015000  042524  052123  052040
 3976  015006  051105  044515  040516
 3977  015014  042524  100104     000
 3978  015021     040     040          E023M:  .ASCII  /  /
 3979  015023     040  020040     040  LINE5:  .ASCII  /    /          ;MESG FOR TEST E024
```

```
 3980  015027     040  020040  020040  LINE5A: .ASCIZ  /        /<ACRLF>
 3981  015034  020040  100040     000
 3982  015041     200  005017  042523  MESG3:  .ASCIZ  <ACRLF><17><12>/SELECT TEST NUMBER  /
 3983  015046  042514  052103  052040
 3984  015054  051505  020124  052516
 3985  015062  041115  051105  020040
 3986  015070     000
 3987  015071     105  042526     116  EVEN:   .ASCII  /EVEN/
 3988  015075     117  042504     040  ODD:    .ASCII  /ODD /
 3989  015101     124  050131  020105  OPMSG:  .ASCIZ  /TYPE ANY CHARACTER/
 3990  015106  047101  020131  044103
 3991  015114  051101  041501  042524
 3992  015122  000122
 3993  015124  051525  020105  047523  NOSWR:  .ASCIZ  /USE SOFTWARE SWITCH REG AT MEMORY ADDR 176/<7>
 3994  015132  052106  040527  042522
 3995  015140  051440  044527  041524
 3996  015146  020110  042522  020107
 3997  015154  052101  046440  046505
 3998  015162  051117  020131  042101
 3999  015170  051104  030440  033067
 4000  015176  000007
 4001
 4002
 4003                                  ;CHGF6  **************
 4004  015200  020040  043536  000040  CNTG:   .ASCIZ  /  ^G /
 4005  015206  020040  053523  036522  SWR:    .ASCIZ  /  SWR  /
 4006  015214  000040
 4007  015216  020040  042516  036527  NEW:    .ASCIZ  /  NEW  /
 4008  015224  000040
 4009  015226  037440  000040          QUEST:  .ASCIZ  / ? /
 4010                                  ;**************
 4011          000001                          .END
```

```
ACRLF - 000200   1265#  2077  3926  3932  3937  3942  3945  3947  3949  3953  3956  3970  3974
                 3980  3982
ADD1    006756   2758  2774#
ADD2    006700   2756  2760#  2773
ADD3    006746   2761  2771#  2776
ADJR4   012034   3464  3476  3481  3482  3483  3504#
ADTFNP  004166   2167  2184#
AREAD = 104021   1287#  2692  2712  2735  2764  2794  2822
AT0     005354   1524  1587  1773  2396#
AT0X    005356   1412*  1416*  1419*  1421*  1522  2397#
AT1     005406   1416  1421  1774  2397  2412#
AT10    006032   1781  2528  2546#
AT11    006070   1782  2547  2564#
AT12    006130   1783  2565  2582#
AT13    006204   1784  2583  2606#
AT14    006264   1785  2607  2630#
AT15    006350   1786  2631  2655#
AT16    006450   1787  2656  2684#
AT17    006516   1788  2685  2704#
AT2     005440   1775  2413  2427#
AT20    006566   1789  2705  2725#
AT21    006660   1790  2726  2753#
AT22    006764   1791  2754  2784#
AT23    007072   1792  2785  2814#
AT24    007204   1793  2815  2853#
AT3     005472   1776  2428  2443#
AT4     005524   1777  2444  2458#
AT5     005614   1778  2459  2482#
AT6     005672   1779  2483  2503#
AT7     005762   1780  2504  2527#
BIT0  = 000001   1242#
BIT1  = 000002   1243#
BIT10 = 002000   1252#
BIT11 - 004000   1253#  1260
BIT12 = 010000   1254#
BIT13 = 020000   1255#  2254
BIT14 = 040000   1256#  1259
BIT15 = 100000   1257#  1562  1968
BIT2  = 000004   1244#
BIT3  = 000010   1245#
BIT4  = 000020   1246#
BIT5  = 000040   1247#
BIT6  = 000100   1248#  2465  2466  2470  2471  2489  2493  2510  2511  2515  2516  2534  2536
                 2593  2616  2641  2665  2740  2769  2800  2826  2877  2888  3123
BIT7  = 000200   1249#  1594  1621  1718
BIT8  = 000400   1250#  1571  1592  1654
BIT9  = 001000   1251#  1266
BRCTR   000664   1374#  2116*  2123*
BTOASC= 104023   1289#  1473  1510  3457
BUFR    013166   3739  3757  3761  3773#
CHAIN = 104005   1275#  2298  2307  2402  2418  2433  2449  2474  2496  2519  2539  2556  2575
                 2596  2619  2647  2669  2697  2717  2745  2772  2806  2831  2892  2941  2969
                 3019  3099  3134  3192  3256  3324  3373  3393  3425  3486  3550  3683  3771
                 3811
CHAINN  001500   1550#  1830
CHAINV  001626   1575#  1656
```

```
CHALT - 104006           1276#   1413    1422    1612
CHGF2   001046           1425#
CHGF3   001500           1549#
CHGF4   006674           2759#
CHGF5   013700           3821#
CHGF7   007204           2852#
CHLT    000736           1406#   1831
CKPOS   011766           3460    3469    3490#
CKSWR   013700           1550    3822#   3876
CNT     000734           1395#   3885*   3914*
CNTG    015200           3831    4004#
CNTLSW  000634           1362#   1449*   1551    1649*   1651    1654    1693*   1704*   1706*   1715*   2357*
CNTLU   013752           1440    3833#   3851
CNTR    000716           1387#   2863*   2869*   2882*
CNVCTR  004160           2165*   2178*   2181#
CONADD  000602           1345#   2137    2248    2319    2339    2341
CONIT   003776           1452    1717    2137#   2193
CONSET  004002           2138#   2201
CONVEC  000604           1346#   2147    2148
COUNT   000726           1392#   3841*   3855    3869*
COUNT3  000666           1375#   2196*   2203*   2259*   2327*
CR    = 104022           1288#   1921    3086    3094    3129    3145    3185    3249    3283    3300    3318    3473
CRBUF   000646           1367#
CRLF  = 104012           1280#   1916    1951    2351    2938    2964    3016    3018    3178    3187    3191    3238    5251
                         3255    3288    3305    3323    3369    3372    3390    3465    3572    3594
CTRA    000650           1368#
CURTST  000676           1379#   1526    1589    1599    1628    1725    2099*
DELAY = 104010           1278#   1659    1667    1680    1690    2122    2295    2304    2314    2350    2361    2569    2691
                         2711    2734    2763    2793    2821    2866    3171    3189    3231    3253    3348    3417    3542
                         3592    3653    3742    3798
DIGIT   004162           2170*   2173*   2176*   2177    2182#
DIRTN   012114           3445*   3504    3511*   3518*   3521#
DISPLA  000720           1389#   1435*
DISPRE  000174           1329#   1435
DLADR   000606           1347#   1489
DLCNT   000656           1371#   1491*   1498*   1507    2194    2196    2256    2259
DLNR    000610           1352#   1490
DLY     003524           1833    2035#
DL11S   014551           1512    3947#
DL11S1  014564           1508    3949#
ECOEND  014772           2297    2306    3549    3682    3770    3810    3974#
EHALT = 104002           1272#   2125
EHLT    003434           1827    1969#
EMTINT  003010           1312    1809#
EMTTAB  003064           1818    1825#
ENDPAS  014523           3485    3942#
END2    001420           1493    1506#
END2A   001406           1488    1502#
END3    001350           1492#   1505
END4    001422           1500    1503    1507#
EO20    012120           1757    3537#   3787
EO21    012170           1758    3538    3563#
EO21A   012176           2299    3566#
EO22    012226           1759    3564    3582#
EO22A   012234           2308    3585#
EO23    012500           1760    3583    3648#
```

```
E023M   015021          3663    3680    3978#
E024    013022          1761    3649    3735#
E024B   013030          2353    3738#   3763    3772
E025    013570          1762    3736    3786#
E025MA  014645          3792    3795    3959#
E025MB  014745          3808    3970#
ERR     003406          1826    1963#
ERRHLT  003442          1972#
ERROR   104001          1271#   2404    2420    2435    2451    2468    2473    2495    2513    2518    2538    2555    2574
                        2595    2622    2643    2667    2677    2696    2716    2742    2775    2802    2828    2839    2861
                        2873    2880    2885    2894
EVEN    015071          3706    3987#
FILL    000730          1393#   3883*   3893
FORWD   104024          1290#   1525    1584    1588    1598    1627    1724
FORWDA  003702          2099#   2103
FORWDB  003710          2096    2101#
FSTDL   000632          1361#   1494*   2197    2260
HDRMSG  014613          1939    3953#
HDRO    014627          1471    2928    3218    3448    3956#
HERE    011534          3419    3425#
ICTR    000660          1372#   1568*   2097*   2102*
IDEZ    007700          2996    3024#
INCHK   000710          1384#   1446*   1657*   1662*   2261    2333
LEVEL   000654          1370#   1406    1450*   1521*   1563    1580    1600*
LF      104014          1282#   1952    3098    3119    3133    3182    3248    3415    3477
LFCNT   000706          1383#   3114*   3118    3122*   3123
LINE3   010446          3116    3148#
LINE5   015023          3686    3979#
LINE5A  015027          3699    3980#
LOGICA  011524          1321    3421#
LSI11   001000          1266#   2550    2688    2708    2731    2760    2790    2818    2857
MACHER  000004          1302#   1447*   1488*   1493*   1582*   1596*   1613*   1707*   2400*   2416*   2431*   2447*
MESG3   015041          1661    3982#
MG24    013012          3667    3673    3715#
MG25    013016          3677    3719#
MODE    000732          1394#   3884*   3889    3892*   3903*
MONIC   012274          3602#   3660
MOVNUM  012740          3661    3674    3678    3699#
NEW     015216          3839    4007#
NEXT    001746          1591    1596#
NEXT1   001766          1593    1600#
NITRSW  004000          1260#   1565    1651
NOSWR   015124          3993#
NPCODE  010141          2997    3051#
NXTST   000640          1364#   1524*   1585    1587*   1590    1595*   1624*   1625    1721*   1722    2092    2094*
ODD     015075          3708    3988#
OPEN    000000          1258#   1361    1362    1363    1364    1365    1366    1367    1368    1369    1370    1371    1372
                        1373    1374    1375    1376    1378    1379    1380    1381    1382    1383    1384    1385    1387
                        1390    1391    1392    1393    1394    1395
OPMSG   015101          2865    3989#
OUT     014174          3823    3825    3829    3873#
OUTPUT  013114          3746    3757#
PARITY  000702          1381#   2214*   2219*   2222*   2224*   3704
PASCNT  012116          3443*   3450*   3451    3453*   3455    3523#
PASMES  014542          3454    3494    3496    3945#
PCHAR   000704          1382#   2212*   2213*   3688    3694*   3695*   3696*
```

```
PFAIL   003546          1451    2054#
POPSP = 005726          1261#
POPSP2= 022626          1262#   1506    1574    1642    2293    2302    2311    2352    2598    2621    2645    2676    2747
                        2774    2804    2838    2881    2895
PRGID   000644          1366#   1448*   1554    1562*   1594*   1619*   1621*   1718*   1965    1968*   2095
PRGTAB  002614          1624    1721    1741#
PRINTC= 104015          1283#   1477    1918    1923    1927    1937    1946    1950    2933    2961    2972    2976    2978
                        3003    3008    3014    3080    3093    3126    3128    3137    3139    3142    3162    3166    3167
                        3175    3181    3196    3222    3226    3227    3235    3241    3243    3245    3275    3295    3309
                        3346    3389    3412    3461    3470    3493    3495    3497    3499    3569    3588    3767
PRNT  - 104017          1285#   1668    1681    1691    1726    1890    1894    1896    3545    3751    3803
PRTHDR= 104016          1284#   2926    2951    2995    3076    3113    3158    3216    3271    3341    3386    3409    3446
                        3539    3565    3584    3650    3737    3788
PRTY4 = 000200          1264#   1358    1360
PRTY7 - 000340          1263#   1313    2462    2486    2507    2531
PSW   = 177776          1238#   2799*
PT0     007374          1595    1741    2924#   3408
PT1     007450          1742    2925    2949#
PT10    011270          1749    3270    3339#
PT11    011426          1750    3340    3384#
PT12    011460          1751    3385    3407#
PT12A   011466          3410#   3426
PT17    011550          1756    3441#
PT17B   011540          3438#   3439    3442
PT17D   011600          3440    3449#   3487
PT2     007572          1743    2950    2993#
PT3     010166          1744    2994    3074#
PT4     010306          1745    3075    3111#
PT5     010464          1746    3112    3156#
PT5AL   010632          3186    3190    3194#   3250    325-
PT6     010670          1747    3157    3214#
PT7     011056          1748    3215    3269#
QUEST   015226          3862    4009#
READ  = 104020          1286#   1663    1676    1686    3540    3651    3740    3796
READC = 104025          1291#
READ1   004300          2202    2211#
REF     010416          3117    3132    3136#
REPT    000662          1373#   2285*   2289*   2368*   2371*   3566*   3568    3585*   3587
RESTRT  003604          2062    2065#
RTNNO   000636          1363#   1520*   1578    1940    1947    2093*   2263    2274    2279    2283    2287    2291    2300
                        2309    2335    2347    2363    2366    2369
SAVR6   003602          2061*   2064#   2067
SCOPSW= 040000          1259#   1556    1963
SCOPTR  000642          1365#   1558    1560    2098*   2101*
SCRLF = 104013          1281#   1888    3760    3805    3854
SELHLT  000752          1407    1411#
SKIP    001722          1523    1592#
SPARET  003160          1847    1848    1849    1850    1851    1852    1853    1854    1855#   1856
SPBOT   000600          1340#   1423    1583    1597    1614    1708
SPC     007562          2977#   3083    3089    3278    3291    3313
SPCNT   000674          1378#   3077*   3087*   3088    3095*   3096
SPSP    007542          2956    2958    2971#
SP2     007554          2975#   3013
SR      000714          1386#   1429    1434*   1438    1463    1556    1565    1571    1576    1592    1615    1963    2254
                        2550    2688    2708    2731    2760    2790    2818    2857    3822    3835    3872*
START   001026          1332    1421#   2075
```

```
STARTM  014376  1479  3926#
STARTX  001042  1414  1417  1420  1423#
START1  000754  1333  1412#
START2  000772  1334  1415#
START3  001010  1335  1418#
STLSPV  003474  1829  2001#
STLSRV  003444  1828  1984#
STPCHV= 104004  1274#  2586  2610  2634  2659
STPPA   003512  2001*  2004#
STPRA   003462  1984*  1987#
STRDRV= 104003  1273#  2729  2757  2788  2823  2875  2886
STRLN   012662  3656  3672  3676  3685#
SWR     015206  3834  4005#
SWREG   000176  1330#  1434  1438  3822
TEMP    000712  1385#  2248*  2249*  2250  2252*  2253*  2319*  2320*  2321  2341*  2342*  2343
TEMPCH  000700  1380#  1664  1677  1687  1694  1696  1698  1700  1702  1731  1734  1738  2211*
                2212  2214  2215*  2216  2270*  2271*  2272  2277  2281  2285  2289  2315  2343*
                2344*  2345  2354  2365  2368  2371  3543  3654  3657  3665  3668  3743  3745
                3749  3752  3759  3799  3801  3804
TEMPST  000724  1391#  3840*  3864*  3865*  3866*  3868*  3872
TENPWR  004164  2169*  2171  2175  2183#
TERM    013156  2312  3744  3769#
TESTC   002552  1670  1683  1731#
TIB     000722  1390#  3826*  3827*  3828  3844*  3847  3848*  3849  3852  3857  3859  3867*  3868
TIMER   000672  1377#  1433*  2036
TKB     000614  1354#  1444  1645  1669  1682  1692  2140*  2211  2417  2713  2882  3546  3826
                3844
TKLVL   000624  1358#  1443  1988  2766  2796
TKS     000612  1353#  1445*  1643  2119  2138*  2207  2209  2401  2465*  2466  2470*  2471  2489*
                2493  2694  2714  2736*  2740*  2759*  2765*  2769*  2771*  2795*  2800*  2805*  2825*
                2826*  2829*  2859  2867  2877*  2883  2888*  2891*  3824  3842
TKVTR   000622  1357#  1441  1986  2147*  2833*
TPB     000620  1356#  1669*  1682*  1692*  1727*  1891*  1895*  1897*  2118*  2145  2146*  2448  2571*
                3546*  3752*  3804*  3832*  3847*  3913*
TPBS    004072  2145*  2153#
TPLVL   000630  1360#  2005  2612  2637
TPS     000616  1355#  2117*  2142  2143*  2234  2432  2490  2510*  2511  2515*  2516  2534*  2536
                2553  2572  2589*  2593*  2615*  2616*  2618*  2636*  2641*  2646*  2661*  2665*  2668*
                3845  3911
TPSS    004070  2142*  2152#
TPVTR   000626  1359#  2003  2148*  2149*  2671*
TTINTS  014176  1442  3874#
TTYCTL= 104011  1279#  1415  1418  1553  2265  2337
TTY1    002070  1642#  1834
TTY1B   002154  1650  1655  1657#  1728  2358
TYP     003164  1825  1879#
TYPE  = 104000  1270#  2864  3662  3679  3794  3830  3833  3838  3861
TYPEM = 104007  1277#  1478  1511  1660  1938  2065  2296  2305  2927  3217  3447  3484  3548
                3681  3769  3791  3807  3809
TYPM    003252  1832  1909#
TYPOC   014214  3836  3883#
TYPON   014230  3885#
WAITF   001772  1413*  1415*  1418*  1422*  1573  1612#  1625  1626  1722  1752  1753  1754  1755
                1763  1764  1765  1766  1767  1768  1769  1770  1771  1772  1794  1795  1796
                1797  1798  1799  1800  1801  1802  2265  2337
WIDTH   000652  1369#  1470*  2932  3078  3096  3115  3131  3159  3168  3179  3194  3219  3228
```

```
                    3239    3273    3290    3307    3444    3458    3467    3507    3509    3567    3586    3789
XCSR    000670      1376#   2197*   2198    2200    2205*   2260*   2267    2269*   2270    2317*   2318*   2321    2323
                    2325*   2326*   2329*   2331*
$AREAD  003726      1842    2116#   2126
$BTASC  004074      1844    2165#
$CR     003314      1843    1926#
$CRLF   003302      1835    1921#
$FORWD  003650      1845    2092#
$LF     003304      1837    1922#
$PRHDR  003324      1839    1936#
$PRNT   004402      1840    2234#   2235
$PRTC   004412      1838    2248#
$READ   004200      1841    2193#   2217
$READC  004272      1846    2195    2209#   2210
$SCRLF  003230      1836    1894#
.     = 015232      1298#   1300    1302    1304    1306    1308    1310    1314    1315#   1319#   1323#   1327#   1338#
                    3059#   3774#

. ABS.  015232      000

ERRORS DETECTED:  0

CZLACF.BIN,CZLACF.LST/CRF/SOL/NL:TOC-CZLACF.P11
RUN-TIME:  12  25  1 SECONDS
RUN-TIME RATIO: 142/40=3.5
CORE USED:  7K  (13 PAGES)
```