TEKTRONIX®
excellence in information display

PLOT-10

Mini Computer
HP 2114, 2116 & 2100A
User's Manual

INFORMATION DISPLAY PRODUCTS

Tektronix, Inc. • P.O. Box 500 • Beaverton, Oregon 97005 • Phone 644-0161

Document No. 062-1428-00

CUSTOMER AGREEMENT

The Tektronix PLOT-10/Mini Computer System is the sole property of Tektronix, Inc. The System, or any part thereof, may not be reproduced or used outside the Buyer's organization in any manner without the express written consent of Tektronix, Inc.

REFERENCE MATERIAL

Additional copies of this reference material may be ordered using the Document number 062-1428-00.

This program is available in the following machine-entry forms:

User's Manual and Paper Source Tape:	062-1428-01

Please place all orders through your Tektronix Application Engineer.

Copyright © 1971 by Tektronix, Inc., Beaverton, Oregon. Printed in the United States of America. All rights reserved. Contents of this publication may not be reproduced in any form without permission of the copyright owner.

U.S.A. and foreign Tektronix products covered by U.S. and foreign patents and/or patents pending.

Date: 1-1-72

C O N T E N T S

	PAGE
INTRODUCTION	1
MACHINE REQUIREMENTS	2
USE OF ACCESS LEVEL SOFTWARE	3
DETAILED DESCRIPTIONS.	4
FLOWCHARTS	10
ASSEMBLY LISTING	19

INTRODUCTION

The 4010 Access Level Software has been written to facilitate the use of the TEKTRONIX 4010 Computer Display Terminal with the Hewlett Packard 2100 series of computers. The software includes basic routines for the following:

- Plotting
- Graphic Input
- Input of any ASCII character
- Output of any ASCII character

All routines are written in assembly language. The calling sequence is compatible with FORTRAN.

MACHINE REQUIREMENTS

The Access Level Software (ALS) includes eight machine language subroutines, four of which are externally callable. The package provides the FORTRAN or assembly language programmer with a program basis that facilitates the use of the 4010 computer display terminal.

The minimum machine configuration is an HP 2100 series of computers with 4k of core, a model 12531 buffered teleprinter interface, paper tape reader, and a 4010 Computer Display Terminal with a TEKTRONIX Teletype Port Interface.

USE OF THE ACCESS LEVEL SOFTWARE

All of the routines are written in Assembly Language for use under the Basic Control System (BCS). Within this operating system installation of the Access Level Software can be accomplished by simply identifying the select code of the interface that pertains at a given site. This code must be equated to the symbol, SC, in the source program on-line number 48.

The calling sequence of all routines is compatible with and hence callable from FORTRAN. All parameters in the ALS plot package are integers.

In the event that the ALS is to be adapted to operating systems using memory protect, such as the Disk Operating System (DOS) or the Real Time Executive (RTE), it is necessary to modify the internal subroutines LCHIN and LCHOT to use the executive drivers.

DETAILED DESCRIPTIONS

This section gives the general format and a description of each routine. Further information may be obtained from the flow charts and listings.

NAME: CHOUT

PURPOSE: Output a character to the 4010.

USAGE:

FORTRAN:

```
CALL CHOUT (IDATA)
```

ASSEMBLY LANGUAGE:

```
JSB CHOUT
DEF (RETURN)
DEF IDATA
```

PARAMETERS: IDATA USASCII character to output

DESCRIPTION: CHOUT will output the character at location IDATA. CHOUT allows the FORTRAN programmer to use the full ASCII character set with the 4010.

(NOTE: After sending the USASCII characters ESC (33_8) followed by FF (14_8) to erase the screen, the user should generate a 750 ms delay. It takes 750 ms to erase the 4010 screen.)

SUBRS CALLED: LCHOT, LENTR

NAME: CHIN

PURPOSE: Receives one ASCII character from the 4010 and returns to the user with the character in the accumulator.

USAGE:

FORTRAN:

```
CALL CHOUT (IDATA)
```

ASSEMBLY LANGUAGE:

```
JSB CHIN
DEF (RETURN)
DEF IDATA
```

PARAMETERS: IDATA USASCII character received from the 4010.

DESCRIPTION: CHIN will receive any of the 128_{10} USASCII characters generated by the 4010 and store it into IDATA. CHIN allows the FORTRAN programmer to use the full ASCII character set with the 4010.

SUBRS CALLED: LCHIN, LENTR

NAME: TPLOT

PURPOSE: To perform the plotting as specified in either linear interpolate or point plot modes. Point plot is a simulated mode for the 4010.

USAGE:

FORTRAN:

```
CALL TPLOT (ITYPE,
IX, IY)
```

ASSEMBLY LANGUAGE:

```
JSB TPLOT
DEF (RETURN)
DEF ITYPE
DEF IX
DEF IY
```

PARAMETERS: ITYPE defines the type of plotting to be performed.

If: ITYPE > 0 Bright vector to (IX,IY)
ITYPE = 0 Initialize and dark vector to (IX,IY)
ITYPE < 0 Point plot at (IX,IY)

IX IX is an integer between 0-1023 to specify the horizontal coordinate on the screen.

IY IY is an integer between 0-780 specifying a vertical coordinate on the screen.

DESCRIPTION: TPLOT is the general plotting subroutine. The plot mode is determined by the value of the parameter ITYPE. The plotting position is determined by the parameters IX and IY.

Dark Vector (ITYPE = 0) If ITYPE = 0, the terminal is set to linear interpolate mode and the graphics cursor is initialized to the specified (IX,IY) position.

Bright Vector (ITYPE > 0) If ITYPE > 0, the terminal is assumed to be in linear interpolate mode and a vector is drawn from the previous (IX,IY) coordinate to the specified (IX,IY) coordinate. A call to TPLOT in dark vector mode must precede plotting in bright vector mode in order to effect proper initialization of the 4010.

Point Plot (ITYPE < 0) If ITYPE < 0, an intensified point is plotted at the specified (IX,IY) position.

The coordinate values (IX,IY) are decoded into four 5 bit bytes. The four resulting bytes are then sent to the terminal through CHOUT in this order:

High Order IY, Low Order IY, High Order IX and Low Order IX with flag bits 6 and 7 configured to Table 1. Positions taken by the 5 coordinate value bits are represented by W.

	Control Bits		Data				
High Order IY	0	1	W	W	W	W	W
Low Order IY	1	1	W	W	W	W	W
High Order IX	0	1	W	W	W	W	W
Low Order IX	1	0	W	W	W	W	W

Table 1

The point (100,200) would be decoded as:

IY IX

$200_{10} = (0011001000)_2$; $100_{10} = (0001100100)_2$

High Order IY = 0100110 High Order IX = 0100011

Low Order IY = 1101000 Low Order IX = 1000100

SUBRS CALLED: PLOT, LENTR, LCHOT

NAME: CURSI

PURPOSE: To enable the graphics cursor and read the input character together with the cursor coordinates.

USAGE:

FORTRAN:
```
CALL CURSI (ICHAR, IX, IY)
```

ASSEMBLY LANGUAGE:
```
JSB CURSI
DEF (RETURN)
DEF ICHAR
DEF IX
DEF IY
```

PARAMETERS:

IX Coordinate of the vertical crosshair when the character was typed.

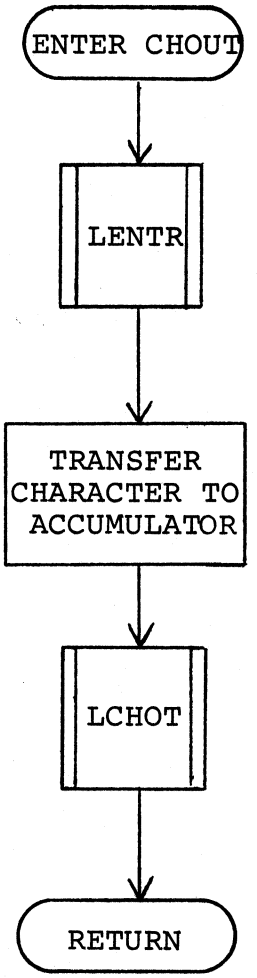
IY Coordinate of the horizontal crosshair when the character was typed.

CHAR The first ASCII character received after the cursor was enabled.

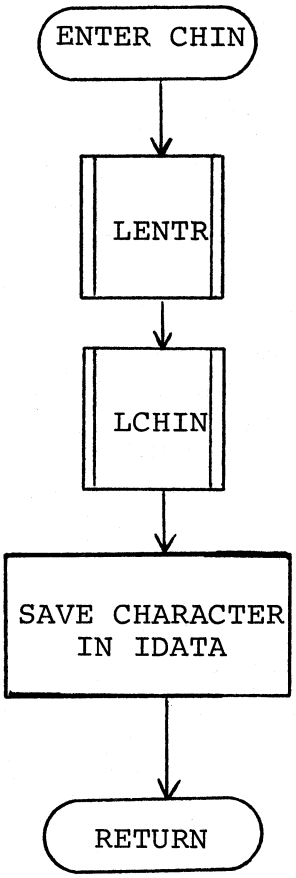
DESCRIPTION: CURSI enables the graphic cursor by outputting the USASCII characters ESC (33_8) then SUB (32_8) to the terminal. With the cursor enabled, CURSI waits for the next keyboard character to be struck which will allow CURSI to read the keyboard character and four additional character sent by the Interactive Graphics Unit. The keyboard character is returned in ICHAR. The remaining four characters are decoded and stored as the cursors present location (IX and IY coordinates). These four characters are transmitted in the following sequence: High Order IX, Low Order IX, High Order IY and Low Order IY, with the coordinate information contained in the bits 1 to 5 of each character and bits 6 and 7 set to 1,0 respectively.

SUBRS CALLED: LENTR, LCHOT, LCHIN

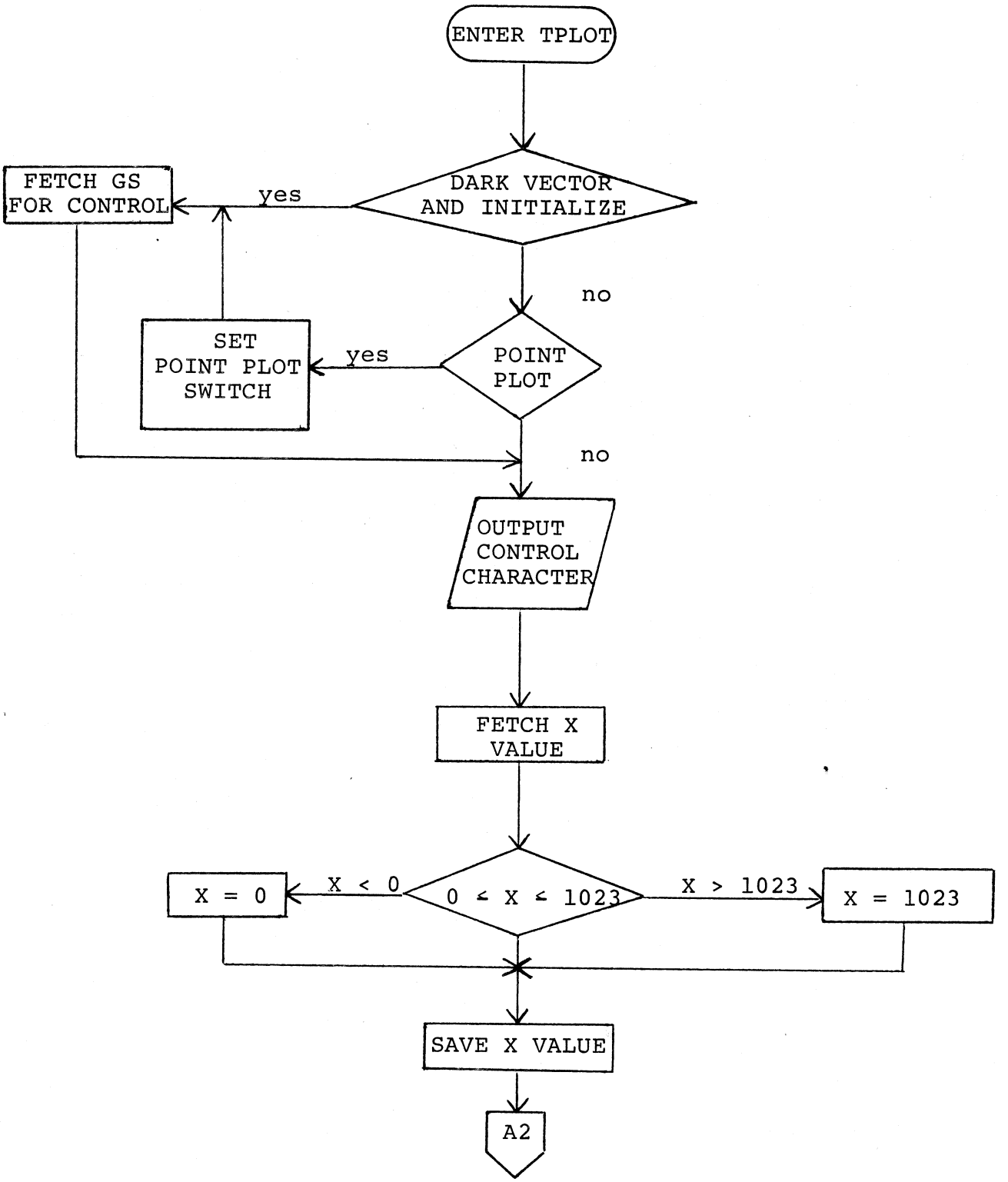
FLOW CHART: CHOUT



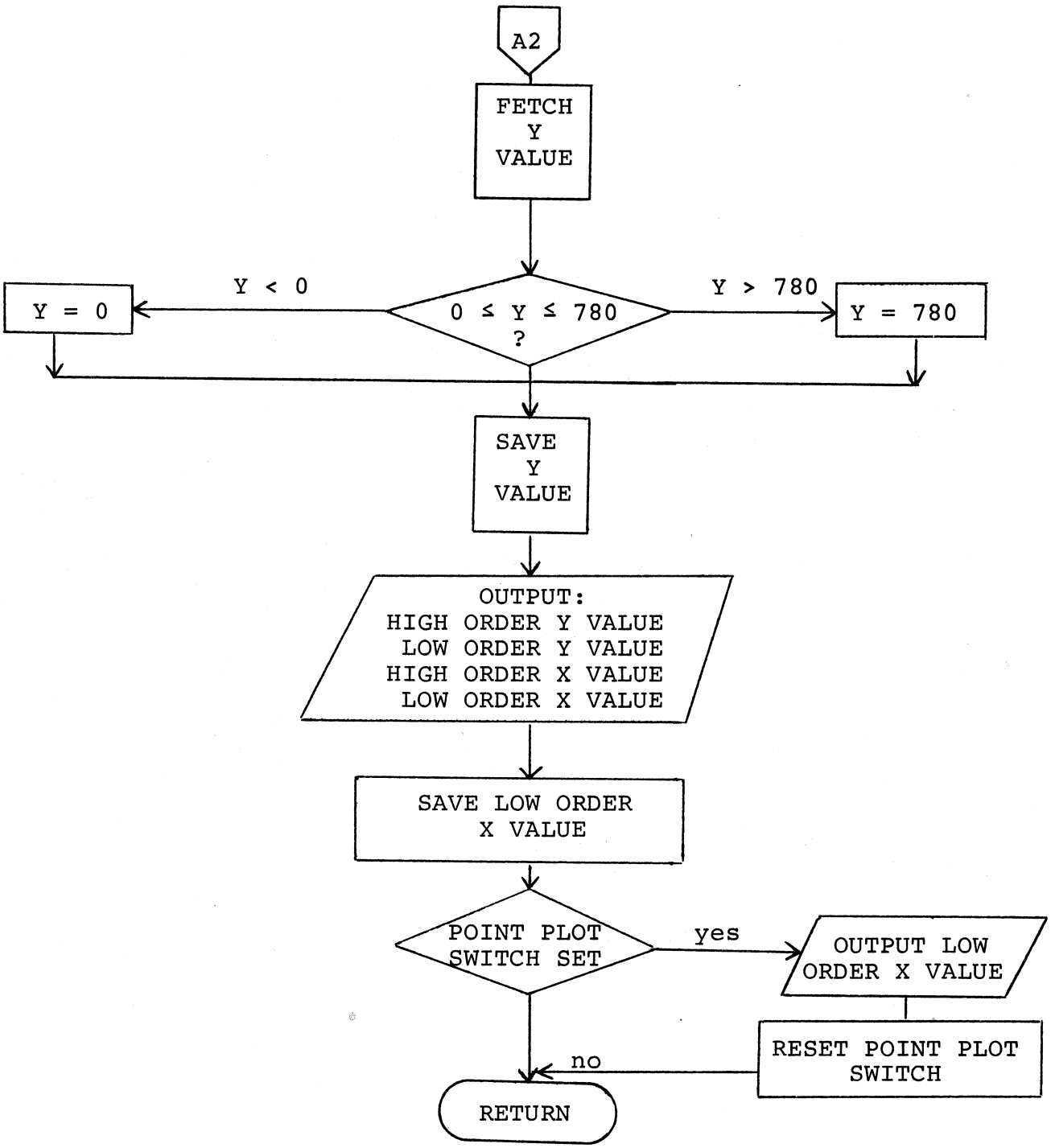
FLOW CHART: CHIN



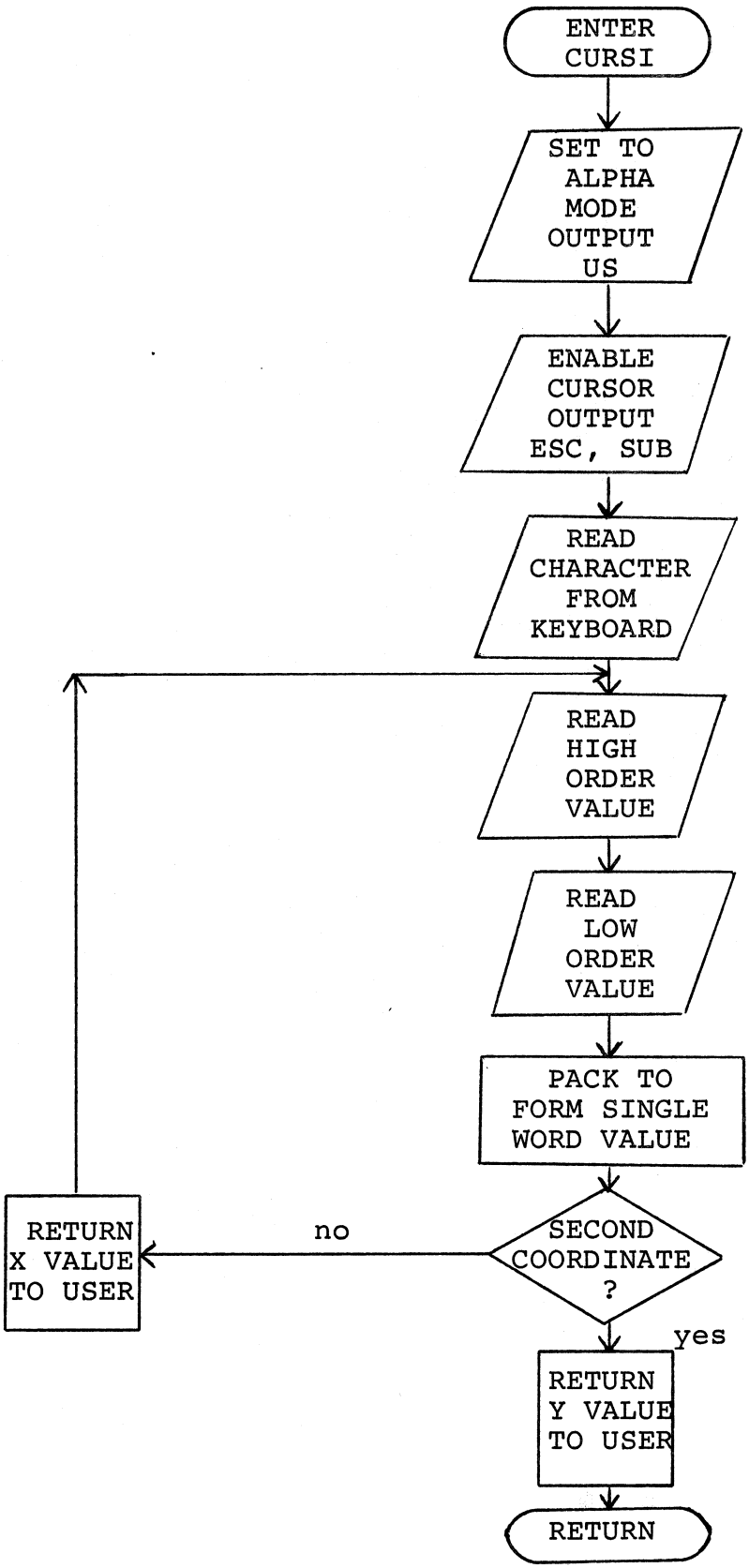
FLOW CHART: TPLOT



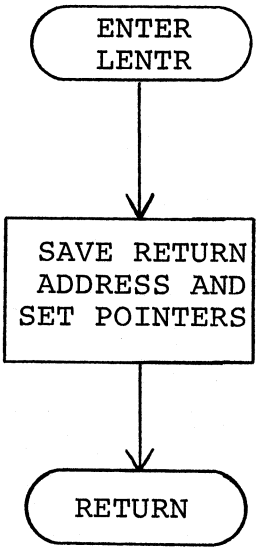
FLOW CHART: TPLOT



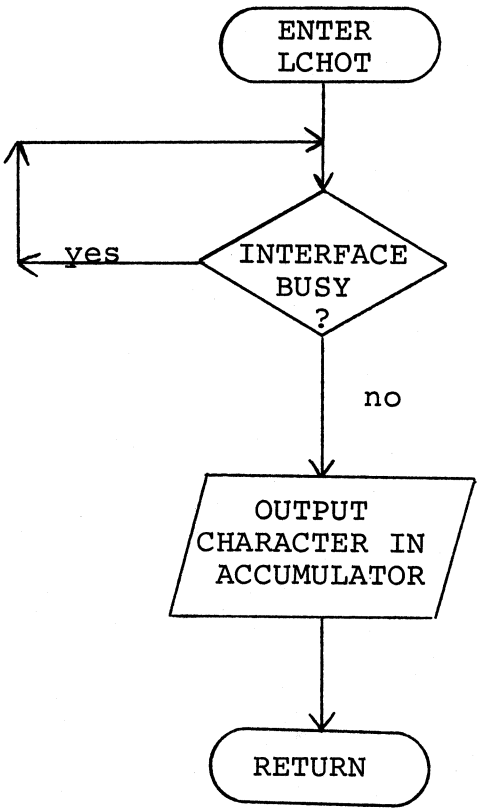
FLOW CHART: CURSI



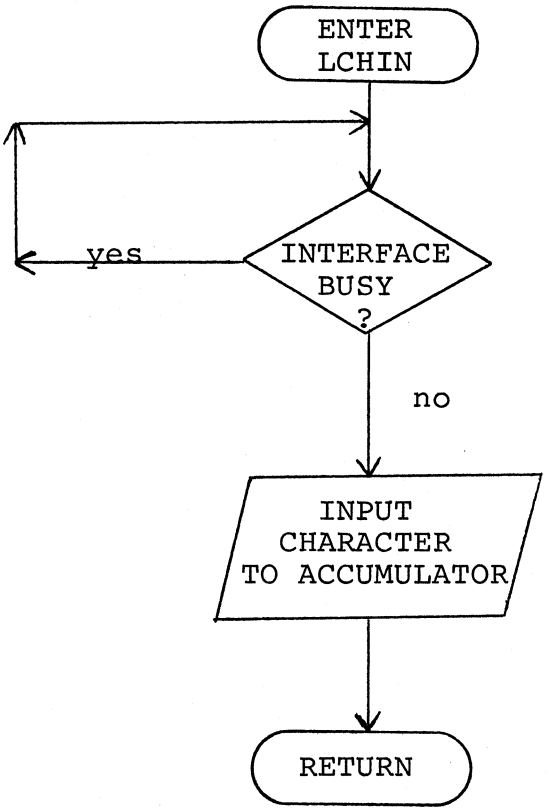
FLOW CHART: LENTR



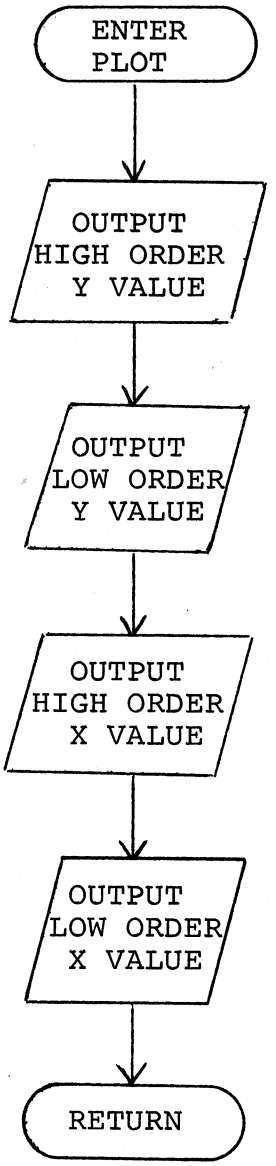
FLOW CHART: LCHOT



FLOW CHART: LCHIN



FLOW CHART: PLOT



```
ASMB,R,C,L,R                                                        00001
      NAM PLOTP                                                     00002
*                                                                   00003
*****************************************************             00004
**                                                   *            00005
**            ACCESS LEVEL SOFTWARE (ALS)            *            00006
**            PLOT PACKAGE FOR THE TEKTRONIX         *            00007
**            4010   COMPUTER DISPLAY TERMINAL       *            00008
**            AND THE HP 2100 SERIES COMPUTERS       *            00009
**                                                   *            00010
**     THE PROGRAM IS TO BE USED WITH THE            *            00011
** MODEL 12531 BUFFERED TELEPRINTER INTERFACE.       *            00012
**                                                   *            00013
**     THE OPERATING ENVIRONMENT IS INTENDED TO      *            00014
** BE RTE.                                           *            00015
**                                                   *            00016
**     FOR USE UNDER OPERATING SYSTEMS USING         *            00017
** MEMORY PROTECT (DOS, RTE) THE ROUTINES LCHIN      *            00018
** AND LCHOT WILL HAVE TO BE MODIFIED.               *            00019
**                                                   *            00020
**     INFORMATION ABOUT INTERFACE SELECT            *            00021
** CODES IS INCLUDED IN THE FOLLOWING BOX.           *            00022
**                                                   *            00023
** ALL PARAMETERS IN THE ALS PACKAGE ARE INTEGERS.   *            00024
**                                                   *            00025
**                                                   *            00026
**                  ALS VERSION 2.0                  *            00027
**                  9 NOV 1971                       *            00028
**                  ATA/N.R.C.                       *            00029
**                                                   *            00030
*****************************************************             00031
*                                                                   00032
      ENT CHIN                                                      00033
      ENT CHOUT                                                     00034
      ENT CURSI                                                     00035
      ENT IPLOT                                                     00036
*                                                                   00037
A     EQU 0           A REGISTER DEFINITION                       00038
B     EQU 1           B REGISTER DEFINITION                       00039
IMPUR EQU 0           TAGS ALTERABLE LOCATIONS                    00040
*                                                                   00041
*****************************************************             00042
**                                                   *            00043
**     THIS CARD INDICATES THE SELECT CODE OF THE    *            00044
** INTERFACE CARD AND MUST BE CHANGED ACCORDING      *            00045
** TO THE INSTALLATION.                              *            00046
**                                                   *            00047
SC    EQU 14B         SELECT CODE OF INTERFACE                    00048
**                                                   *            00049
**                                                   *            00050
*****************************************************             00051
*                                                                   00052
*                                                                   00053
US    OCT 37          CHAR TO SET ALPHA MODE                      00054
GS    OCT 35          CHAR TO SET VECTOR MODE                     00055
ESC   OCT 33          ESC+SUB ENABLE GRAPHICS CURSOR              00056
SUB   OCT 32          CHAR TO ENABLE GRAPHICS CURSOR              00057
*                                                                   00058
M2    DEC -2                                                      00059
B17   OCT 17                                                       00060
B37   OCT 37                                                       00061
B40   OCT 40                                                       00062
```

```
HOB     OCT 40                                                  00063
LXB     OCT 100                                                 00064
LYB     OCT 140                                                 00065
B377    OCT 377                                                 00066
BIT15   OCT 100000                                              00067
TYPB    OCT 120000      TTY OUTPUT BITS                         00068
TYRB    OCT 160000      TTY INPUT BITS                          00069
ARGAD   BSS 1                                                   00070
RETAD   BSS 1                                                   00071
X       BSS 1                                                   00072
Y       BSS 1                                                   00073
SAVEX   BSS 1           SAVE LOW ORDER X BITS                   00074
PPSW    OCT 0           POINT PLOT MODE SWITCH                  00075
```

```
      HED CHOUT                                                          00076
*****************************************************************      00077
**                                                             *      00078
**      CHOUT(IDATA)                                           *      00079
**                                                             *      00080
**      CALLING SEQUENCE                                       *      00081
**          JSB CHOUT                                          *      00082
**          DEF <RETURN>                                       *      00083
**          DEF IDATA                                          *      00084
**                                                             *      00085
**      THIS ROUTINE OUTPUTS THE RIGHT EIGHT BITS              *      00086
** OF IDATA AS A SINGLE CHARACTER.                             *      00087
**      CHOUT ALLOWS THE FORTRAN PROGRAMMER                    *      00088
** TO USE THE FULL ASCII CHARACTER SET                         *      00089
** WITH THE  4010.                                             *      00090
**                                                             *      00091
**      CHOUT CALLS THE INTERNAL CHARACTER OUTPUT              *      00092
** ROUTINE, LCHOT, AS WELL AS LENTR.                           *      00093
**                                                             *      00094
**                                                             *      00095
*****************************************************************      00096
*                                                                    00097
*                                                                    00098
CHOUT ABS IMP JR                                                     00099
      JSB LENTR      SET UP THE POINTERS                             00100
      LDA ARGAD,I    GET IDATA                                       00101
      JSB LCHOT        AND OUTPUT IT                                 00102
      JMP RETAD,I    RETURN                                          00103
```

```
      HED CHIN
*****************************************************
**                                                  *
**      CHIN(IDATA)                                 *
**                                                  *
**     CALLING SEQUENCE                             *
**         JSB CHIN                                 *
**         DEF <RETURN>                             *
**         DEF IDATA                                *
**                                                  *
**      THIS ROUTINE INPUTS A CHARACTER FROM THE    *
** TERMINAL AND STORES IT INTO IDATA.               *
**      IF CHIN IS CALLED AFTER A CHARACTER HAS     *
** BEEN STARTED, A TRASH CHARACTER MAY BE STORED.   *
**      THIS IS A HARDWARE LIMITATION.              *
**                                                  *
**      CHIN ALLOWS THE FORTRAN PROGRAMMER TO USE   *
** THE FULL FORTRAN CHARACTER SET WITH THE  4010 .  *
**                                                  *
**      CHIN CALLS THE INTERNAL CHARACTER           *
** INPUT ROUTINE, LCHIN, AS WELL AS LENTR.          *
**                                                  *
**                                                  *
*****************************************************
*
*
CHIN  ABS IMPUR
      JSB LENTR      SET UP THE POINTERS
      JSB LCHIN      GET AN INPUT CHARACTER
      STA ARG1,I     STORE IT INTO IDATA
      JMP RETAD,I    RETURN
```

```
      HED TPLOT                                                          00135
**********************************************************************  00136
**                                                                  *  00137
**     TPLOT(I,IX,IY)                                               *  00138
**                                                                  *  00139
**     CALLING SEQUENCE                                             *  00140
**         JSB TPLOT                                                *  00141
**         DEF <RETURN>                                             *  00142
**         DEF I                                                    *  00143
**         DEF IX                                                   *  00144
**         DEF IY                                                   *  00145
**                                                                  *  00146
**     THIS ROUTINE DEALS WITH FOUR CASES:                          *  00147
**                                                                  *  00148
**         CASE 1.  I = 0                                           *  00149
**                                                                  *  00150
**     CALLING CASE 1 PUTS THE  4010  INTO LINEAR                   *  00151
** INTERPOLATE MODE AND PLOTS A NON-INTENSIFIED                     *  00152
** VECTOR TO THE (X,Y) COORDINATE POSITION.                         *  00153
** THE USER IS THEN LEFT IN LINEAR INTERPOLATE                      *  00154
** MODE, AND ADDITIONAL INTENSIFIED VECTORS MAY                     *  00155
** BE DRAWN BY CALLING CASE 2.                                      *  00156
** WHILE IN LINEAR INTERPOLATE MODE, THE USER                       *  00157
** MAY EFFECT A DARK VECTOR AT ANY TIME BY CALLING                  *  00158
** CASE 1.                                                          *  00159
**                                                                  *  00160
**         CASE 2.  I > 0                                           *  00161
**                                                                  *  00162
**     CASE 2 IS CALLED TO DRAW AN                                  *  00163
** INTENSIFIED VECTOR TO THE (X,Y) COORDINATE                       *  00164
** POSITION SPECIFIED BY THE PARAMETERS.                            *  00165
** PRIOR TO CALLING THIS ROUTINE, THE  4010                         *  00166
** MUST HAVE BEEN PLACED INTO LINEAR INTERPOLATE                    *  00167
** MODE.                                                            *  00168
**                                                                  *  00169
**         CASE 3.  I < 0                                           *  00170
**                                                                  *  00171
**     CASE 3 SETS THE TERMINAL INTO SIMULATED                      *  00172
** POINT PLOT MODE AND THEN PLOTS A POINT                           *  00173
** AS DETERMINED BY X AND Y.                                        *  00174
**                                                                  *  00175
**     TPLOT CALLS THE INTERNAL ROUTINE, PLOT,                      *  00176
** TO OUTPUT THE (X,Y) COORDINATE INFORMATION                       *  00177
**     TPLOT ALSO CALLS LENTR AND LCHOT.                            *  00178
**                                                                  *  00179
**                                                                  *  00180
**                                                                  *  00181
**********************************************************************  00182
*                                                                       00183
*                                                                       00184
TPLOT ABS IMPUR                                                         00185
      JSB LENTR      SET UP POINTERS                                    00186
      LDA ARGAD,I    GET THE FIRST ARGUEMENT                            00187
      ISZ ARGAD      ADVANCE ARGUEMENT POINTER                          00188
      SZA,RSS        CHECK FOR CASE 1, I = 0                            00189
      JMP DARK                                                          00190
      SSA            SKIP IF BRIGHT VECTOR                              00191
      JMP POINT      CASE 3, I < 0                                      00192
      JMP BRIGHT     CASE 2, I > 0                                      00193
```

```
      HED TPLOT  CASE 1                                          00194
*                                                                00195
*                                                                00196
DARK  EQU *                                                      00197
      LDA GS          GET THE CODE FOR VECTOR MODE               00198
      JMP POIFE         AND ENTER POINT PLOT                     00199
```

```
      HED TPLOT  CASE 3                                          00200
*                                                                00201
*                                                                00202
POINT EQU *                                                      00203
      LDA GS          GET THE CODE FOR POINT PLOT                00204
      STA PPSW        SET POINT PLOT SWITCH                      00205
POIFE EQU *                                                      00206
      JSB LCHOT         AND OUTPUT IT                            00207
POIBR EQU *                                                      00208
      LDA ARGAD,I     GET X                                      00209
      STA X             AND SAVE IT LOCALLY                      00210
      ISZ ARGAD       ADVANCE TO NEXT ARGUEMENT                  00211
      LDA ARGAD,I     GET Y                                      00212
      STA Y                                                      00213
      JSB PLOT        NOW PLOT THIS POINT                        00214
      LDA SAVEX       LOAD LOW ORDER X                           00215
      LDB PPSW        LOAD POINT PLOT SWITCH                     00216
      SZB             SKIP IF NOT POINT PLOT MODE                00217
      JSB LCHOT       OUTPUT LOW ORDER X                         00218
      CLA                                                        00219
      STA PPSW        RESET POINT PLOT SWITCH                    00220
      JMP RETAD,I     RETURN                                     00221
```

```
      HED CURSI                                                          00222
**********************************************************************  00223
**                                                                  *  00224
**      CURSI(ICHAR,IX,IY)                                          *  00225
**                                                                  *  00226
**     CALLING SEQUENCE                                             *  00227
**         JSB CURSI                                                *  00228
**         DEF <RETURN>                                             *  00229
**         DEF ICHAR                                                *  00230
**         DEF X                                                    *  00231
**         DEF Y                                                    *  00232
**                                                                  *  00233
**      THIS ROUTINE SETS THE TERMINAL INTO ALPHA                   *  00234
** MODE, ENABLES THE GRAPHIC CURSER, AND WAITS                      *  00235
** FOR FIVE CHARACTERS TO BE RETURNED.  THE FIRST                   *  00236
** IS PLACED INTO ICHAR AND THE REMAINING FOUR ARE                  *  00237
** CONVERTED TO X AND Y COORDINATES.                                *  00238
**                                                                  *  00239
**      WHEN CURSI IS CALLED, THE FULL SCREEN                       *  00240
** CROSS HAIR CURSOR WILL APPEAR, AND THE                           *  00241
** PROGRAM WILL THEN WAIT FOR THE  4010  TO                         *  00242
** TRANSMIT 5 BYTES OF INFORMATION.  THE USER                       *  00243
** CAUSES THIS TO HAPPEN BY FIRST POSITIONING                       *  00244
** THE CURSOR TO THE DESIRED POSITION (USING                        *  00245
** THE JOYSTICK ADJACENT TO THE TERMINAL) AND                       *  00246
** THEN DEPRESSING A KEY.  THE  4010  SENDS THE                     *  00247
** ASCII VALUE OF THE DEPRESSED KEY TOGETHER                        *  00248
** WITH CURSOR COORDINATE INFORMATION BACK TO                       *  00249
** THE COMPUTER.                                                    *  00250
**                                                                  *  00251
** CURSI CALLS LENTR, LOHOT, AND LOHIN.                             *  00252
**                                                                  *  00253
**                                                                  *  00254
**********************************************************************  00255
*                                                                        00256
*                                                                        00257
CURSI  ABS IMPUR                                                        00258
       JSB LENTR       SET UP THE POINTERS                              00259
       LDA US          GET THE CODE FOR ALPHA MODE                      00260
       JSB LOHOT         AND OUTPUT IT                                  00261
       LDA ESC         GET THE CODE TO ENABLE GRAPHIC INPUT             00262
       JSB LOHOT         AND OUTPUT IT                                  00263
       LDA SUB                                                          00264
       JSB LOHOT                                                        00265
*                                                                       00266
       JSB LOHIN       WAIT FOR THE FIRST INPUT                         00267
       STA ARGAD,I     SAVE THE CHARACTER                               00268
       ISZ ARGAD       ADVANCE POINTER TO NEXT ARGUEMENT                00269
*                                                                       00270
       JSB LOHIN       GET THE NEXT INPUT CHARACTER                     00271
       AND B37         MASK IT TO ONLY FIVE BITS                        00272
       ALF,RAL         POSITION FIVE BITS OVER                          00273
       STA X             AND SAVE THEM IN X                             00274
       JSB LOHIN       GET THE NEXT CHARACTER                           00275
       AND B37           AND MASK TO FIVE BITS                          00276
       IOR X           PLUG IN MSB                                      00277
       STA ARGAD,I     AND RETURN X VALUE TO CALLER                     00278
       ISZ ARGAD       => Y ARGUEMENT                                   00279
       JSB LOHIN       FETCH THE NEXT CHARACTER                         00280
       AND B37           AND MASK IT TO FIVE BITS                       00281
       ALF,RAL         MOVE FIVE PLACES TO THE LEFT                     00282
       STA Y             AND SAVE THEM IN Y                             00283
```

```
JSB LCHIN       GET THE LAST CHARACTER                          00284
AND B37           AND MASK IT OFF TO ONLY FIVE BITS             00285
IOR Y           PLUG IN MSB                                     00286
STA APGAD,I     AND RETURN VALUE TO CALLER                      00287
JMP RETAD,I     RETURN                                          00288
```

```
      HED PLOT
*************************************************************
**                                                         *
**     PLOT                                                *
**                                                         *
**     THIS ROUTINE, FOR LOCAL USE ONLY,                   *
** CONVERTS THE X AND Y COORDINATES INTO                   *
** FOUR CHARACTERS AND OUTPUTS THEM                        *
** TO THE TERMINAL.                                        *
**                                                         *
**     PLOT IS USED BY TPLOT.                              *
**     PLOT CALLS LCHOT.                                   *
**                                                         *
**                                                         *
*************************************************************
*
*
PLOT  ABS IMPUR
      LDA Y          GET THE Y VALUE
      ALF,ALF        DISCARD THE RIGHTMOST
      ALF,RAR          FIVE BITS
      AND B37        MASK TO ONLY FIVE BITS
      IOR HOB        PLUG IN REQUIRED BITS
      JSB LCHOT        AND OUTPUT THE CHARACTER
      LDA Y          GET BACK THE Y VALUE
      AND B37          AND MASK TO ONLY FIVE BITS
      IOR LYB        PLUG IN REQUIRED BITS
      JSB LCHOT        AND OUTPUT THE CHARACTER
      LDA X          NOW GET THE X VALUE
      ALF,ALF        DISCARD THE RIGHTMOST
      ALF,RAR          FIVE BITS
      AND B37
      IOR HOB        PLUG IN THE REQUIRED BITS
      JSB LCHOT        AND OUTPUT THE CHARACTER
      LDA X          GET BACK THE X VALUE
      AND B37          AND LEAVE ONLY LOW ORDER FIVE
      IOR LXB        PLUG IN THE REQUIRED BITS
      STA SAVEX      SAVE LOW ORDER X BITS
      JSB LCHOT        AND OUTPUT THE CHARACTER
      JMP PLOT,I     RETURN
```

00289
00290
00291
00292
00293
00294
00295
00296
00297
00298
00299
00300
00301
00302
00303
00304
00305
00306
00307
00308
00309
00310
00311
00312
00313
00314
00315
00316
00317
00318
00319
00320
00321
00322
00323
00324
00325
00326
00327
00328

```
      HED LCHIN                                                    00329
*************************************************************  00330
**                                                           *  00331
**     LCHIN                                                 *  00332
**                                                           *  00333
**     THIS ROUTINE, FOR LOCAL USE ONLY, INPUTS              *  00334
** A CHARACTER FROM THE TERMINAL AND LEAVES                  *  00335
** IT IN THE A REGISTER.                                     *  00336
**     SINCE A BCS DRIVER MAY BE RUNNING ON AN               *  00337
** INTERRUPT BASIS WHEN WE ENTER, A WAIT LOOP                *  00338
** HAS BEEN INSERTED TO WAIT FOR A CONSTANT                  *  00339
** FLAG (MEANING THAT THE DRIVER HAS FINISHED                *  00340
** AND HAS CLEARED THE CONTROL F.F.)                         *  00341
**                                                           *  00342
**     LCHIN IS CALLED BY CHIN AND CURSI.                    *  00343
**                                                           *  00344
**                                                           *  00345
*************************************************************  00346
*                                                               00347
*                                                               00348
LCHIN ABS IMPUR                                                 00349
LCHIP EQU *                                                     00350
      SFS SC          WE ARE NOT ALLOWED TO                     00351
      JMP LCHIP         TOUCH IF BEING USED                     00352
      LDA *           TIME WASTER                               00353
      LDA *                                                     00354
      SFS SC          CHECK ONCE MORE TO MAKE SURE              00355
      JMP LCHIP                                                 00356
      LDA TYPB        GET THE TTY INPUT BITS                    00357
      OTA SC            AND OUTPUT TO THE INTERFACE             00358
      STC SC,C        START THE INPUT                           00359
      CLC SC          PREVENT AN INTERRUPT                      00360
      SFS SC          WAIT FOR SOME ACTION                      00361
      JMP *-1                                                   00362
      LIA SC          GET THE CHARACTER                         00363
      AND B377          AND MASK IT OFF                         00364
      JMP LCHIN,I     RETURN                                    00365
```

```
      HED LCHOT
**************************************************
**                                              *
**     LCHOT                                    *
**                                              *
**     THIS ROUTINE, FOR LOCAL USE ONLY, OUTPUTS *
** THE CHARACTER IN THE A REGISTER TO THE       *
** TERMINAL.  A IS MASKED TO EIGHT BITS.        *
** THE B REGISTER IS DESTROYED.                 *
**     SINCE A BCS DRIVER MAY BE RUNNING ON AN  *
** INTERRUPT BASIS WHEN WE ENTER, A WAIT LOOP   *
** HAS BEEN INSERTED TO WAIT FOR A CONSTANT     *
** FLAG (MEANING THAT THE DRIVER HAS FINISHED   *
** AND HAS CLEARED THE CONTROL F.F.)            *
**                                              *
**                                              *
**************************************************
*
*
LCHOT ABS IMPUR
      AND B377      MASK THE CHARACTER TO SIZE
LCHOP EQU *
      SFS SC        WE ARE NOT ALLOWED TO
      JMP LCHOP       TOUCH IF BEING USED
      LDB *         TIME WASTER
      LDB *
      SFS SC        CHECK ONCE MORE.
      JMP LCHOP
      LDB TYPO      GET THE TTY OUTPUT BITS
      OTB SC          AND TELL INTERFACE BOARD
      OTA SC        PUT THE CHARACTER OUT
      STC SC,C      START THE OUTPUT
      CLC SC        PREVENT AN INTERRUPT
      SFS SC        WAIT FOR THE OUTPUT
      JMP *-1         TO FINISH
      JMP LCHOT,I   RETURN
```

00366
00367
00368
00369
00370
00371
00372
00373
00374
00375
00376
00377
00378
00379
00380
00381
00382
00383
00384
00385
00386
00387
00388
00389
00390
00391
00392
00393
00394
00395
00396
00397
00398
00399
00400
00401

```
      HED LENTR                                                          00402
*********************************************************************** 00403
**                                                                     * 00404
*     LENTR                                                           * 00405
*                                                                     * 00406
**    THIS ROUTINE IS CALLED TO SAVE THE RETURN                        * 00407
** ADDRESS AND TO SET UP A POINTER TO THE                             * 00408
* ARGUEMENT LIST.                                                     * 00409
**                                                                    * 00410
**    A CALL TO LENTR ( JSP LENTR ) MUST                              * 00411
* BE THE FIRST INSTRUCTION FOLLOWING THE ENTRY                        * 00412
* POINT OF EACH EXTERNALLY CALLABLE ROUTINE.                          * 00413
**                                                                    * 00414
*                                                                     * 00415
*********************************************************************** 00416
*                                                                     00417
*                                                                     00418
ENTR  ABS IMPUR                                                       00419
      LDA LENTR      A = ADDRESS OF OUR CALLER + 2                    00420
      ADA M2         SUBTRACT OFF THE TWO                             00421
      LDA A,I        FIND WHERE HE WAS CALLED FROM                    00422
      IOR BIT15      PLUG IN THE INDIRECT BIT                         00423
      STA RETAD      SAVE PTR TO RETURN ADDRESS                       00424
      INA            ADVANCE POINTER TO ARGUEMENT LIST                00425
      STA ARGAD        AND SAVE FOR THE ROUTINE                       00426
      JMP LENTR,I    RETURN                                           00427
      END                                                             00428
```

READER'S COMMENT FORM

Your comments about this publication may be helpful to us.

If you wish to comment, please use the space provided below, giving specific page and paragraph reference.

Please do not use this form to ask technical questions about the equipment or to make requests for copies of publications. Instead, make such inquiries to your Tektronix Application Engineer.

Reply requested

Yes ☐

No ☐

Name _____

Job Title _____

Address _____

_____ Zip _____

Document No. 062-1428-00

YOUR COMMENTS PLEASE

If you have any comments on this publication, please write them on the reverse side of this sheet.

Your comments will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of Tektronix.

Note: Please direct any requests for copies of publications, or for assistance in using your Tektronix equipment to your Tektronix Application Engineer.

fold *fold*

FIRST CLASS
PERMIT NO. 61
BEAVERTON, OREGON

BUSINESS REPLY MAIL
No postage necessary if mailed in the United States

Postage will be paid by

TEKTRONIX, INC.
P.O. BOX 500
BEAVERTON, OREGON 97005
U.S.A.

ATTN: TEKTRONIX USER'S LIBRARY

fold *fold*

TEKTRONIX, INC.
P.O. BOX 500
BEAVERTON, OREGON 97005
U.S.A.

ATTN: TEKTRONIX USER'S LIBRARY