

HP-75 PLOTTER ROM

External Reference Specification

Bob Miller  
Ext. 2672

Version 1.0

<Company Confidential>

11:15 AM FRI., 12 NOV., 1982

Table of Contents

1	Introduction . . . . .	1
2	About the HP-75 Plotter ROM . . . . .	2
2.1	What It Is . . . . .	2
2.2	RAM Required . . . . .	2
3	Command Set . . . . .	4
3.1	Definitions . . . . .	4
3.1.1	GRAPHIC LIMITS . . . . .	4
3.1.2	DEFAULT GRAPHICS SCALE . . . . .	4
3.1.3	GRAPHICS UNITS (GUs) . . . . .	5
3.1.4	PLOTTING BOUNDARIES . . . . .	5
3.1.5	PLOTTING AREA . . . . .	5
3.1.6	USER UNITS (UUs) . . . . .	5
3.2	PLOTTER IS . . . . .	6
3.3	RATIO . . . . .	8
3.4	GCLEAR . . . . .	9
3.5	LIMIT . . . . .	10
3.6	SCALE . . . . .	11
3.7	SHOW . . . . .	12
3.8	MSCALE . . . . .	13
3.9	LOCATE . . . . .	14
3.10	CLIP . . . . .	15
3.11	UNCLIP . . . . .	16
3.12	SETGU . . . . .	17
3.13	SETUU . . . . .	17
3.14	PEN . . . . .	18
3.15	PENUP . . . . .	19
3.16	LINETYPE . . . . .	20
3.17	Optional Pen Control Parameter . . . . .	21
3.18	PLOT . . . . .	22
3.19	IPLOT . . . . .	23
3.20	RPLOT . . . . .	24
3.21	MOVE . . . . .	25
3.22	DRAW . . . . .	26
3.23	IMOVE . . . . .	27
3.24	IDRAW . . . . .	28
3.25	PDIR . . . . .	29
3.26	XAXIS and YAXIS . . . . .	30
3.27	AXES . . . . .	31
3.28	FRAME . . . . .	32
3.29	GRID . . . . .	33
3.30	LABEL . . . . .	34
3.31	LORG . . . . .	35
3.32	LDIR . . . . .	36
3.33	Csize . . . . .	37
3.34	Labeling Axes . . . . .	38

3.35	FXD . . . . .	39
3.36	LAXES . . . . .	40
3.37	LGRID . . . . .	41
3.38	DIGITIZE . . . . .	42
3.39	CURSOR . . . . .	43
3.40	WHERE . . . . .	44
3.41	VELOCITY . . . . .	45

Introduction	CHAPTER 1
--------------	-----------

The HP-75 Plotter ROM will be the foundation of CVD's graphic strategy for the HP-75. This ROM will provide the graphic primitives (i.e. MOVE, DRAW, etc.) necessary for the development of a Graphic Presentation PAC and any other Graphics PAC written by HP or an outside software house.

The Plotter ROM will be an 8K ROM containing 39 microcoded primitives. The ROM is essentially cloned from the HP-85 Printer/Plotter ROM in that, in the majority of cases, the primitives will function EXACTLY the same as the HP-85 PP ROM. In those cases where the commands are different the differences and the reasons for those differences will be pointed out. There is only 1 new command, VELOCITY, which will allow the user to slow the pen down. This command is helpful for producing consistent line quality on transparencies.

The benefits of cloning the HP-85 PP ROM are: 1) a quick ROM development cycle; 2) continuity of command capability and format for our customers and 3) ease of program porting from the 85 to the 75. The disadvantage of this course is that we temporarily limit functionality to an existing command set. Currently the strategy is to incorporate the plotter ROM primitives into a 32K Graphic Presentation ROM. Development on this ROM will start once the primitives have been completed. It may be possible to include some advanced primitives in this 32K ROM (i.e., an array plot command, a rectangle command, a circle and arc command, etc.).

About the HP-75 Plotter ROM	CHAPTER 2
-----------------------------	-----------

## 2.1 What It Is

The HP-75 Plotter ROM is just that--a PLOTTER ROM. It does not drive a raster device, nor does it drive a printer. It transmits HPGL (Hewlett Packard Graphic Language) commands to the device on the loop which the user has specified with the PLOTTER IS statement (similar to the PRINTER IS statement).

Unlike the PRINTER IS statement, only one device on the loop may be specified as the active plotter by the PLOTTER IS statement. This is required by the interaction between the plotter and the 75. The 75 must interrogate the plotter to find the location of its scaling points P1 and P2;



The scaling points P1,P2 have a different default depending on the plotter module, plus they may be manually set by controls on the plotter. Thus what makes sense to one plotter may be garbage to another.

## 2.2 RAM Required

As with other Plotter ROMs (the 85 and 41) a block of RAM is required to keep track of the plotter scaling points, pen location, scaling factors, etc. This is accomplished on the HP-75 by creating a system file, ".plotfil", at cold and warm start. The file will consume about 370 bytes of user RAM. If this file does not exist (i.e., there was not room to create

HP-75 Plotter ROM --- Company Confidential

the file at warm start) every plotter command will generate  
the error message: "Plotter Data File Missing".

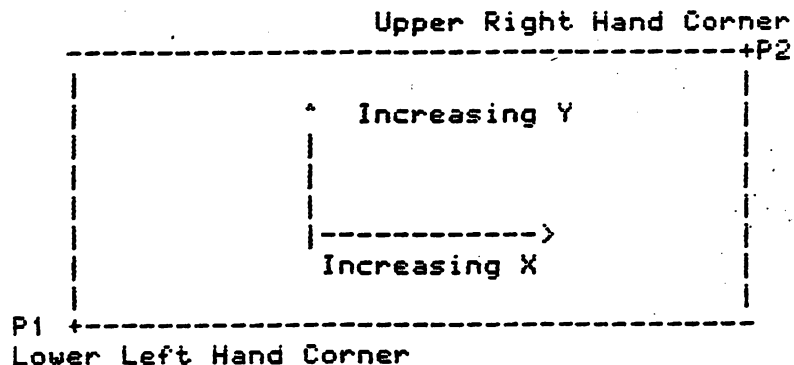
11:15 AM FRI., 12 NOV., 1982

Command Set	CHAPTER 3
-------------	-----------

### 3.1 Definitions

#### 3.1.1 GRAPHIC LIMITS

The GRAPHIC LIMITS specify the area of the plotter's platen in which the pen is free to move. The GRAPHIC LIMITS may be the physically imposed, hardware-set limits or they may be limits set manually or programmatically. For the HP-75 Plotter ROM the GRAPHIC LIMITS will be the area defined by the P1 and P2 scaling points of the plotter.



#### 3.1.2 DEFAULT GRAPHICS SCALE

Regardless of the size of the GRAPHIC LIMITS, the DEFAULT GRAPHICS SCALE is always calculated the same way. The short side is scaled from 0 to 100 and the long side is scaled from 0 to 100 times the ratio of the longest side to the shortest side.

The default scale is provided so that users can begin plotting on a known scale as soon as the PLOTTER IS statement is executed. Also the default scale provides easy access to plotting on a percentage basis regardless of the size of the GRAPHIC LIMITS.

### 3.1.3 GRAPHICS UNITS (GUs)

The units provided by the DEFAULT GRAPHICS SCALE are called the GRAPHICS UNITS.

### 3.1.4 PLOTTING BOUNDARIES

The default PLOTTING BOUNDARIES are equal to the GRAPHIC LIMITS. However, it is possible (using LOCATE) to specify a subset of the GRAPHIC LIMITS; the PLOTTING BOUNDARIES. PLOTTING BOUNDARIES are particularly useful when you wish to save some space in the GRAPHIC LIMITS for labels. ONLY labels may be plotted outside of the PLOTTING BOUNDARIES, but the labels must still fall within the GRAPHIC LIMITS.

### 3.1.5 PLOTTING AREA

The area of the platten that is currently active is called the Plotting Area. This area may be the entire GRAPHIC LIMITS or a subset of those limits created by a LOCATE or CLIP statement.

### 3.1.6 USER UNITS (UUs)

It is possible to change the scale of the area specified by the PLOTTING BOUNDARIES using the SCALE statement. The units of the area thus specified are called the USER UNITS.



### 3.2 PLOTTER IS

PLOTTER IS <device specifier> on.\*

The PLOTTER IS statement specifies the target device for all graphic statements provided by the Plotter ROM. Only 1 device may be the active plotter in the loop. Until a device has been identified by the PLOTTER IS command all other plotting commands will respond with the "NO PLOTTER" error message.

This statement sets the default plotting conditions:

1. Activates the plotting device specified.
2. Reads the scaling points, P1,P2 from the plotting device and makes them the endpoints of the GRAPHIC LIMITS.
3. Scales the plotting area to the default scale (shortest side from 0 to 100; longest side from 0 to  $RATIO*100$  if horizontal, or from 0 to  $100/RATIO$  if vertical). NOTE: RATIO here refers to the Plotter Module command RATIO and means:  $(total\ X\ units)/(total\ Y\ units)$ .
4. Sets the UUs equal to the GUs and specifies the user units mode (SETUU).
5. Defaults the plotting boundaries (set by CLIP and LOCATE) to the graphic limits.
6. Selects pen one (PEN 1).
7. Selects line type one and repeat length four (LINETYPE 1,4).
8. Selects standard character size (CSIZE 3, .6).
9. Selects label origin one (LORG 1).
10. Sets label direction as left-to-right (LDIR 0).
11. Clears the character count from any previous LABEL statement.

HP-75 Plotter ROM --- Company Confidential

12. Clears any error conditions.

11:15 AM FRI., 12 NOV., 1982

### 3.3 RATIO

The RATIO function returns a value equal to the ratio of the physical dimensions of the GRAPHIC LIMITS, that is, the units in the X dimension divided by the units in the Y dimension.

3.4 GCLEAR

On plotting devices with roll paper, the GCLEAR statement advances the paper a full page. On fixed paper plotters (e.g., the 7470A), the GCLEAR statement is ignored.

### 3.5 LIMIT

LIMIT [x min, x max, y min, y max]

The LIMIT statement enables the user to define new GRAPHIC LIMITS as long as the specified new GRAPHIC LIMITS falls within the limits of the hardware.

The first two parameters specify the logical left and right limits while the second two parameters specify the logical lower and upper limits of the new GRAPHIC LIMITS.

The parameters of the LIMIT statement are ALWAYS expressed in millimeters.

Executing the LIMIT statement without parameters enables the user to digitize two opposite corners to describe the new GRAPHIC LIMITS.

7470 0, 272, 0, 191

### 3.6 SCALE

SCALE x min, x max, y min, y max

The first two parameters specify the values represented by the logical left and right PLOTTING BOUNDARIES. The last two parameters specify the values represented by the logical lower and upper PLOTTING BOUNDARIES.

The SCALE statement ALWAYS maps units onto the area specified by the LOCATE statement or onto the default PLOTTING BOUNDARIES if no LOCATE statement has been executed.

The SCALE statement automatically sets the active units to UUs mode.

3.7 SHOW

SHOW x min, x max, y min, y max

The SHOW statement is an easy means of providing UUs of equal scale, that is, one length of measure in the X direction will equal one length of measure in the Y direction (isotropic scaling). The parameters specify the minimum acceptable units in either direction and the scale is then stretched or shrunk proportionally in the X and Y directions so that the aspect ratio of the units will equal 1.

The SHOW statement automatically sets the active units to UUs mode.

### 3.8 MSCALE

MSCALE x-offset, y-offset

The MSCALE statement sets millimeters as UUs and defines the origin. MSCALE is useful when correspondence with physically measurable objects is desirable.

The origin (0,0) in millimeter space, is offset from the logical minimum corner of the current PLOTTING BOUNDARIES by the specified amounts in millimeters. The specified reference point need not be inside the GRAPHIC LIMITS.

The MSCALE statement automatically sets the active units to UUs mode.



### 3.9 LOCATE

LOCATE [x min, x max, y min, y max]

The LOCATE statement enables the user to define PLOTTING BOUNDARIES smaller than or equal to the GRAPHIC LIMITS.

The parameters of the LOCATE statement are expressed in GUs. It is convenient to think of the LOCATE parameters as defining PLOTTING BOUNDARIES on a percentage basis of the short side of the GRAPHIC LIMITS.

The first two parameters specify the logical left and right boundaries and the last two parameters specify the logical lower and upper boundaries.

Executing the LOCATE statement without parameters enables the user to digitize two opposite corners (normally the lower-left and upper-right hand corners) of the new PLOTTING BOUNDARIES.

It is important to note that LOCATE does not change the active units mode.

### 3.10 CLIP

CLIP [x min, x max, y min, y max]

The CLIP statement redefines the PLOTTING BOUNDARIES. The parameters are interpreted according to the currently active units (either UUs or GUs). The first two parameters define the left and right PLOTTING BOUNDARIES while the second two parameters specify the lower and upper PLOTTING BOUNDARIES.

Executing CLIP without parameters allows the user to digitize two opposite corners (normally the lower-left and upper-right hand corners) of the new PLOTTING BOUNDARIES.

It is important to note that SCALE statements executed after CLIP will still be superimposed on the area set by the last LOCATE statement (or on the default PLOTTING BOUNDARIES which are equal to the GRAPHIC LIMITS) if no LOCATE statements have been executed. Also, CLIP does not change the active units mode.

3.11 UNCLIP

The UNCLIP statement resets the PLOTTING BOUNDARIES equal to the GRAPHIC LIMITS. UNCLIP does not change the active units mode. Nor does UNCLIP change the UU scale factors.

3.12 SETGU

SETGU sets graphic units (GUs) as the current units mode.

3.13 SETUU

SETUU sets user units (UUs) as the current units mode.

3.14 PEN

PEN <pen number>

The PEN statement enables the user to select any pen or to stable all pens.

Non-integer pen number parameters are rounded up or down to the nearest integer value. Negative as well as positive pen values are sent to the plotter. The handling of the pen numbers sent is plotter dependent but no error will be reported to the user.

PEN 0 will return all pens to their stalls.

NOTE: The implementation of this command differs from that of the HP-85 PP ROM. The 85 ROM would only allow pen numbers 0 through 4 to be sent and in fact would do a mod 5 to get such a number.

3.15 PENUP

The PENUP statement lifts the pen.

### 3.16 LINETYPE

LINETYPE <type number> [, <length>]

The LINETYPE statement selects one of several solid or dashed line types (the exact number is plotter dependent) and a repeat factor for each type. Non-integer type numbers will be rounded up or down to the nearest integer. Length parameters may be real numbers. Negative as well as positive parameters are sent to the plotter. The plotter's reaction to the parameters is plotter dependent but no error will be reported to the user.

Type number 0 and 1 produce a solid line.

### 3.17 Optional Pen Control Parameter

Three plotting statements provide optional pen control: PLOT, IPLIT and RPLIT. The optional pen control parameter enables users to control the pen's up or down position and whether the up or down pen change is made before or after the pen moves. The pen control parameter for these statements is interpreted as follows:

Parameter	Pen Control
odd	pen down
even	pen up
positive	pen change after motion
negative	pen change before motion

The default value for the optional pen control parameter is +1. This means that the pen is lowered after the pen is moved.



3.18 PLOT

PLOT x-coord, y-coord [ , pen control ]

This statement moves the pen to the absolute x,y coordinates in current units using the current pen number and line type. The optional pen control parameter specifies the up or down pen movement and defaults to +1 (the pen moves and then drops).

Plotted lines drawn in UUs will not extend beyond the PLOTTING BOUNDARIES. Plotted lines drawn in GUs can be drawn outside of the PLOTTING BOUNDARIES but not beyond the GRAPHIC LIMITS.

3.19 I PLOT

I PLOT x-increment, y-increment [ , pen control ]

The I PLOT (incremental plot) provides incremental plotting capability with pen control. The origin (point 0,0) is assumed to be the pen's position before the statement is executed. The pen moves or draws the amount specified in the x & y direction from the current pen location.

Parameters are interpreted according to current units.

The optional pen control parameters are the same as for PLOT.

The direction of incremental pen movement can be rotated with the PDIR statement.

3.20 RPLLOT

RPLLOT x-relative, y-relative [, pen control ]

The RPLLOT (relative plot) statement provides relative plotting capability with pen control. The origin is assumed to be the last pen position determined by a statement other than RPLLOT, specifically:

PLOT	DRAW	FRAME	AXES	LAXES
IPLOT	MOVE	GRID	XAXIS	LGRID
LABEL	IMOVE	IDRAW	YAXIS	

The RPLLOT statement interprets x & y coordinates according to current units relative to a local origin.

The direction of RPLLOT can be rotated about the local origin with the PDIR statement.

The optional pen control parameter directs the up and down movement of the pen and is interpreted the same as for PLOT.

3.21 MOVE

MOVE x-coordinate, y-coordinate

The MOVE statement lifts the pen and moves the pen to the specified x,y coordinate position in current units and leaves the pen up. The pen's movement is restricted only by the GRAPHIC LIMITS.

3.22 DRAW

DRAW x-coordinate, y-coordinate

The DRAW statement drops the pen and draws a line to the specified x,y coordinate position in current units.

3.23 IMOVE

IMOVE x-increment, y-increment

The IMOVE (incremental move) statement provides incremental moving capability. The current pen position is treated as the origin and the pen is moved the specified units in the x and y direction according to the current units.

3.24 IDRAW

IDRAW x-increment, y-increment

The IDRAW (incremental draw) statement provides incremental drawing capability. The current pen position is treated as the origin, the pen is dropped and a line is drawn the specified units in the x and y direction according to the current units.

### 3.25 PDIR

PDIR angle or PDIR run, rise

The PDIR (plot direction) statement sets the angle of axes rotation for relative plotting (RPLLOT) and incremental plotting (IPLLOT, IMOVE and IDRAW) in the current angular units mode (DEG, RAD, or GRAD). If the plot direction is specified with run/rise parameters, the parameters are interpreted in the current units.

The angle of axes rotation can be specified in either of 2 ways:

- As the counterclockwise angle, in current angle units (DEG, RAD, or GRAD) between the new x-axis and the horizontal axis.
- The rotation parameter can alternately be specified as the run and rise units (in current units) of a vector drawn in the desired direction.



3.26 XAXIS and YAXIS

XAXIS y-intercept [, tic-spacing [,xmin, xmax] ]

YAXIS x-intercept [, tic-spacing [,xmin, xmax] ]

The XAXIS statement generates a horizontal axis at the specified y-intercept value. The YAXIS statement generates a vertical axis at the specified x-intercept value. An intercept must be specified, the remaining parameters are optional.

The x and y tic-spacing parameters are interpreted in current units. Tics are placed on the axis at the specified interval.

The sign of the tic-spacing parameter determines whether tics will be drawn in the direction of increasing magnitude (positive) or decreasing magnitude (negative).

The min and max parameters enable a shorter axis to be drawn within the current plotting area.

3.27 AXES

```
AXES [x tic-spacing, y tic-spacing [ , x-intersection,  
      y intersection [ , x major count , y major count  
      [ , major tic size ] ] ] ]
```

The AXES statement draws a pair of axes with optional (linearly spaced) major and minor tic-marks. The axes intersect at (x intersection, y intersection); the coordinates are interpreted in current units. The axes are drawn across the entire current plotting area (bounded by the PLOTTING BOUNDARIES if in UUs modes; and by the GRAPHIC LIMITS if in GUs mode. The intersection point may lie outside the plotting area, in which case only that portion of the axes within the plotting area is drawn.

The x and y tic-spacing specify, in current units, the distance between tic-marks on each axis. The absolute value is taken of negative values; zero signifies "no tics at all."

The optional x and y major count parameters are positive integers which specify the number of tic intervals between major tics. For example, an x major count of 2 means that every other tic on the x-axis is a major tic. Tic marks are drawn so that a major tic falls at the axis intersection point (whether visible or not); tic marks need not fall on the plotting boundaries. The default values for the count parameters are 1,1 (all tics are major tics).

The optional tic size parameter specifies the length of a major tic, end-to-end in GUs. Minor tics are always half the size of major tics. The default length of a major tic is 2 GUs.

When AXES is executed without parameters, the system will automatically draw a pair of axes that intersect at the lower-left corner of the plotting area. The axes will be drawn in the current scale with approximately 10 evenly spaced tics on each axis.

3.28 FRAME

The FRAME statement draws a box around the current plotting area defined by LIMIT, LOCATE or CLIP.

The box is drawn using the current pen and linetype around the current plotting area. The pen is positioned at the lower left corner of the frame after the frame is complete.

### 3.29 GRID

```
GRID [x tic-spacing, y tic-spacing [ , x-intersection  
      , y intersection [ , x grid spacing, y grid spacing  
      [, tic size ] ] ] ]
```

The parameters are the same as for the AXES statement, except that the last parameter specifies the length of the minor tics only. The grid lines (major tics) are drawn completely across the plotting area and a cross tic is drawn at the intersection of the tics. The grid lines are spaced symmetrically about the intersection of the x axis with the y axis. The tic size parameter specifies the length of the axes tics in GUs; the default size is 2 GUs. The default value for the x,y intersection is the lower left corner of the plotting area.

The grid line and tic marks are drawn using the current line type and pen number.

3.30 LABEL

LABEL [label list]

or

LABEL USING image specifier [; label list]

or

LABEL USING statement number [; label list]

The LABEL and LABEL USING statements are very similar to the PRINT and PRINT USING statement except that the label list is directed to the current plotter instead of a printer or alphanumeric display.

The position and rotation of a label are determined by the current pen position and the LORG (label origin) and LDIR (label direction) statements. The size, aspect ratio and slant of the characters in the label are determined by the CSIZE (character size) statement.

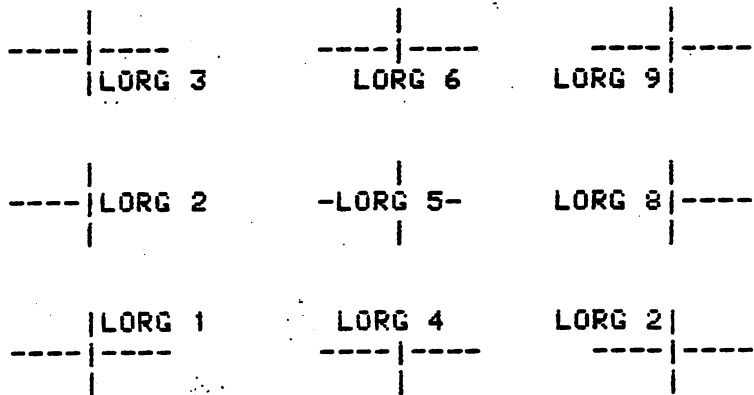
The characters are drawn in GUs mode (regardless of the currently active mode) and are clipped only at the GRAPHIC LIMITS. Whether labels can or cannot be reflected depends upon the capabilities of the active plotter. On plotters that can reflect labels the user must exchange the order of the LIMIT statement parameters to effect the reflection.

3.31 LORG

LOGR label position

The label origin statement sets the label origin that determines where subsequent labels will be placed relative to the current pen position. The label position parameter can have a value between 1 and 9.

In the following figure, each cross marks the initial position of the pen for the corresponding label at each of the LOGR positions. LOGR positions are calculated on the size of an upper case letter.



3.32 LDIR

LDIR angle or LDIR run, rise

The angle can be specified in current angular units (DEG, RAD or GRAD). The value of the angle is the counterclockwise angle between the letter direction and the horizontal x axis. Or the angle can be specified as the number of current units of run (horizontal distance) along the x axis and the number of current units of rise (vertical distance) above the x axis.

### 3.33 CSIZE

CSIZE height [ , aspect ratio [ , slant ] ]

The CSIZE statement specifies the height, aspect ratio and slant of the characters used in labels.

A character is composed of both a symbol and the space that surrounds the symbol and separates it from other characters. Thus the actual height of the symbol will be smaller than the height parameter due to the space around it. The character space height is twice the character's height and the character space width is 1.5 times the character width.

The height parameter specifies the height, in GUs, of the entire character space. The default height is approximately 3 GUs after the PLOTTER IS or LIMIT statement is executed.

The width is specified indirectly with the aspect ratio parameter. The aspect ratio is the ratio of the width of the character, not including the space around it, to the height. The default aspect ratio is .6.

The slant parameter specifies the angle from the vertical, in current angular units (DEG, RAD or GRAD).



### 3.34 Labeling Axes

Two statements provide automatic labeling of axes: LAXES, and LGRID. Axes and grid labels are drawn using the current CSIZE specifications in the format determined by the FXD statement. The LORG and LDIR statements do not affect LAXES and LGRID labels.

Axes labels are always placed outside of the current plotting area (described by LOCATE or CLIP). It is most important to allow space for axes labels between the plotting area and the GRAPHIC LIMITS. If not enough space is allowed labels may be clipped or not appear at all.

3.35 FXD

FXD number of digits [, number of digits]

The FXD (fixed) statement establishes a format for axes and grid labeling with LAXES and LGRID. The FXD parameters must be a number between 0 and 7. They specify the number of digits to the right of the decimal point in the axes labels. If only one parameter is given, both axes are fixed in the specified format; if two parameters are given, the first fixes the X axis format and the second fixes the Y axis format. Number labels that do not fit the fixed format are forced to exponential notation in the format "MDE" (as expressed in an IMAGE statement). The default labeling format is FXD 0.

3.36 LAXES

LAXES [x tic-spacing, y tic-spacing [ , x-intersection,  
y-intersection [ , x major count, y major count  
[ , major tic size ] ] ]

The LAXES (label axes) statement draws a pair of axes and labels them at the same time. The parameters are interpreted the same as for the AXES statement. Labels are placed outside the plotting boundaries, within the GRAPHIC LIMITS, on each major tic. With positive tic-spacing, labels for the corresponding axis are perpendicular to that axis. With negative tic-spacing, labels are parallel to the axis.

3.37 LGRID

LGRID [x tic-spacing, y tic-spacing [ , x-intersection,  
y-intersection [ , x grid-spacing, y grid-spacing  
[ , tic size ] ] ] ]

The parameters are interpreted the same as for the AXES, LAXES  
and GRID statements. Labels appear on each grid line.

3.38 DIGITIZE

DIGITIZE x variable, y variable [, pen status variable]

DIGITIZE pauses the program and allows user to reposition the cursor by moving the pen. When the user pushes the [ENTER] button on the plotter the coordinate values of the pen position are assigned to the x,y variables. The coordinates are returned in the current units (GU or UU) form. If the optional third parameter is present, the pen status (0=up 1=down) will also be returned.

3.39 CURSOR

CURSOR x variable, y variable [, pen status variable] .

The CURSOR statement returns the cursor coordinate location and pen status information to the specified variables. Coordinate values are returned in the current active units scale (GU or UU). This statement is useful because the the plotter maintains a physical pen location and a logical cursor location. CURSOR does not pause the program and does not require the user to press the [ENTER] button on the plotter.

3.40 WHERE

WHERE x variable, y variable [, pen status variable]

The WHERE statement assigns the coordinate values of the current pen position (last plotted or moved-to point) to the specified variables.

### 3.41 VELOCITY

#### VELOCITY [pen velocity]

The VELOCITY instruction is used to set pen velocity to a speed other than the default velocity of 38.1 cm/s and to change the acceleration from its default value of 2 g's (980 cm/s<sup>2</sup>). A velocity of 10 cm/s is the recommended velocity for overhead transparencies.

A VELOCITY command with no parameters resets the pen velocity and acceleration to their default values. Valid parameters for velocity are between 0 and 38.1. A PLOTTER IS statement restores the default pen velocity and acceleration.