

Hewlett-Packard

System 45B

Desktop Computer



Graphics ROM Programming

HEWLETT  PACKARD

Warranty Statement

Hewlett-Packard products are warranted against defects in materials and workmanship. For Hewlett-Packard Desktop Computer Division products sold in the U.S.A. and Canada, this warranty applies for ninety (90) days from date of delivery.* Hewlett-Packard will, at its option, repair or replace equipment which proves to be defective during the warranty period. This warranty includes labor, parts, and surface travel costs, if any. Equipment returned to Hewlett-Packard for repair must be shipped freight prepaid. Repairs necessitated by misuse of the equipment, or by hardware, software, or interfacing not provided by Hewlett-Packard are not covered by this warranty.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR CONSEQUENTIAL DAMAGES.

*For other countries, contact your local Sales and Service Office to determine warranty terms.

Graphics ROM Programming



HP System 45B Desktop Computer

Hewlett-Packard Desktop Computer Division
3404 East Harmony Road, Fort Collins, Colorado 80525
(For World-wide Sales and Service Offices see back of manual.)
Copyright by Hewlett-Packard Company 1979

Printing History

Periodically, this manual is updated. Each new edition of this manual incorporates all material updated since the previous edition. Each new or revised page is indicated by a revision date.

The date on the back cover changes only when each new edition is published.

First Printing...January 1, 1979

Second Printing...May 2, 1979

Table of Contents

Chapter 1: General Information

Overview	1
Key Words	1
For the Beginning Graphics Programmer	2
For the Advanced Graphics Programmer	2
Definition of Terms	2
Plotting Space	2
Unit-of-Measure Mode	3
Line Generation	4
About the Examples	4
The Appendices	6
Syntax Guidelines	6
What to Expect	6
Summary	7

Chapter 2: Simple Plotting Operations

Plotting Boundaries	10
Hard Clip Area	10
Soft Clip Area	10
Plotting Coordinates	10
GRAPHICS and EXIT GRAPHICS Statements	12
Example	13
PEN Statement	14
Jaggies	14
Example	15
LINE TYPE Statement	16
Examples	17
PLOT Statement	20
Example	21
PLOTTER IS Statement	22
Statement Parameters	22
CRT	22
9872A	22
Incremental	23
Default Conditions	24
Example	25

PLOTTER...IS ON or IS OFF Statements	26
Example	27
GCLEAR Statement	28
Example	29
FRAME Statement	30
Example	30
DUMP GRAPHICS Statement	32
Example	33
PENUP Statement	34
Example	35
Summary	36

Chapter 3: Positioning and Scaling Plots

Overview	38
Current Units	39
Positioning	39
Scaling	40
Offsets	40
Other Statements	41
LIMIT Statement	41
Example	43
CLIP Statement	44
Example	45
UNCLIP Statement	48
Example	49
LOCATE Statement	50
Example	51
Scaling the Plotting Area	52
SCALE Statement	53
Examples	54
SHOW Statement	56
Example	57
MSCALE Statement	58
Example	59
More About Plotting Units	60
SETGU and SETUU Statements	61
Example	61
RATIO Function	62
Example	63
Summary	64

Chapter 4: Plotting Enhancements

Overview	66
Enhancing the Plotting Area	66
AXES Statement	67
Example	68
GRID Statement	70
Example	70
MOVE Statement	72
Example	72
DRAW Statement	74
Example	74
RPLOT Statement	76
Example	77
IPLOT Statement	78
Example	79
RPLOT vs IPLOT	80
Rotating the Plot	81
PDIR Statement	81
Example	82
Summary	84

Chapter 5: Labeling Statements

Overview	86
LABEL and LABEL USING Statements	87
Example	88
LETTER Statement	90
Erasing Characters	91
The Graphicswriter	91
Example	92
CSIZE Statement	94
Examples	95
LDIR Statement	99
Example	100
LORG Statement	102
Example	102
Example	104
Example	106
Summary	107

Chapter 6: Digitizing and Cursor Control

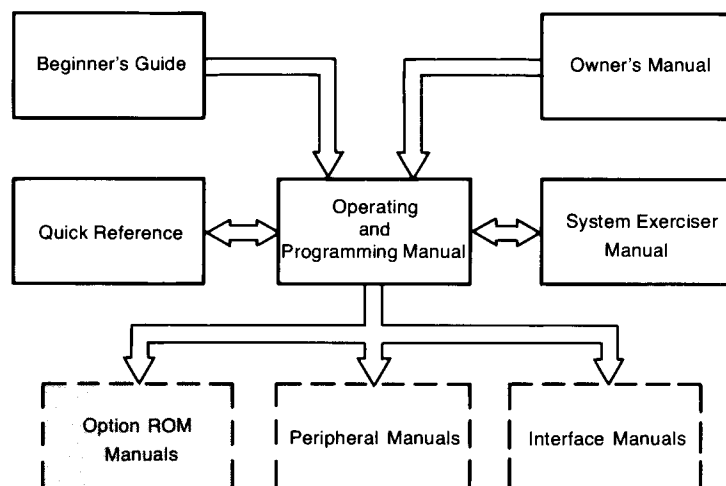
Pen vs Cursor Concept	110
POINTER Statement	111
Example	112
CURSOR Statement	114
Example	114
DIGITIZE Statement	116
Digitizing from the CRT	116
Digitizing with the 9872A	117
Digitizing with an Incremental Plotter	117
Example	118
WHERE Statement	120
Example	120
Summary	122

Chapter 7: Advanced Graphic Techniques

GLOAD and GSTORE Statements	124
Example	125
More About CRT Arrays	127
Example	128
Reflective Plotting	130
Example	132
More About the HP 9872A Plotter	134
Example	135
Curve Fitting	137
Example	137
Menu Selection	138
Examples	139
Some Interesting Examples	144
SIN(X)/X Example	144
Football Field Example	146
LOG/LIN Chart Example	148
PLOT Example	149
FRAME Example	151
LIMIT Example	152
CLIP Example	153
Histogram Using Clip Statement	154
Plotting Outside the Hard Clip Area	155
UNCLIP Example	156
LOCATE Example	157

SHOW and SCALE Examples	158
Metric Scale Example	162
GDU and UDU Example	163
AXES Examples	164
GRID Example	166
MOVE Example	167
DRAW Example	168
LABEL Example	169
LETTER Example	170
CSIZE Example	171
Character Set Example	172
Slant and Angle Example	174
LDIR Example	175
LORG Examples	176
POINTER Examples	178
CURSOR Example	180
DIGITIZE Examples	181
Appendix A: Glossary	183
Appendix B: Syntax Summary	185
Appendix C: Graphics Device Information	193
Index	199
Sales and Service Offices	204
Tear-Out Reference Card	

Manual Map



General Information

Overview

Your HP System 45B Graphics ROM provides CRT graphic solutions and/or hard copy illustrations of problems solved by the computer. In addition, graphics provides other valuable capabilities such as labeling plots, axes, tick marks on the axes, and digitizing from a plot or graphic display.

Before attempting to use your System 45B and this manual, you should be familiar with the basic operation of your System 45B. Refer to your System 45B Operating and Programming manual if you need this information.

Information contained in this manual is specifically directed to the CRT as the graphics display device. Most information contained in this manual is applicable to other external display or plotting units, such as the HP 9872A Plotter. Some of the more common differences between a graphics statement's effect on the CRT versus the 9872A plotter are mentioned in the discussion concerning the statement.

The simulated CRT graphic drawings used to produce this manual were actually drawn on the HP 9872A Plotter. Therefore, the simulations in this manual may appear slightly different from the results you obtain on the CRT.

Chapter 1

For the Beginning Graphics Programmer

Graphics is the solution to a problem which is provided by means of graphs or diagrams rather than by a printed list of numbers.

Example programs are provided throughout this manual to give you some hands-on experience and to enhance the explanation of various statements. Each of the forty-one graphics statements for your System 45B is explained at its appropriate place in this manual.

By the end of this manual, you should be able to plot, label and digitize the various solutions to the problems and equations in your programs.

For the Advanced Graphics Programmer

If you have programming experience using graphics, you can either use the summary pages at the beginning of each chapter or you may go directly to the tear-out reference card in the back of this manual.

Plotting Space

Plotting devices have physically-imposed **hard clip** limits which restrict pen movement (e.g., CRT screen size and plotter platen size). Default hard clip limits and the resulting plotting area size depend upon which plotting device is being used. As seen later, hard clip limits can be made smaller under program control (e.g., to fit a piece of paper), or on some plotters, manually. Whether this area is the default or has been set to a specified size, it is still referred to as the area within the hard clip limits.

Another set of limits that further restrict pen movement is the soft clip limits. Soft clip limits are defaulted to hard clip limits, but can be altered under program control. **Soft clip** limits restrict pen movement when the User Defined Units (UDU) mode is set and not when the Graphic Display Units (GDU) mode is set. As seen later, this capability is very useful, since you may wish to label outside of the plotting area.

Unit-of-Measure Modes

Your System 45B allows you to define or declare dimensions for its X,Y coordinate system. This allows you to access an X,Y coordinate by using your coordinate system, even though your System 45B is converting and computing your coordinate system onto its coordinate system.

There are three unit-of-measure modes that can be selected under program control: **Graphic Display Units (GDU)**, **User Defined Units (UDU)**, and **Metric Units**.

One **GDU** is defined as being one percent of the length of the shortest side of the space bounded by the current hard-clip limits. Thus, the short side is 100 GDU's long and the length of the long side is 100 times the ratio of the long to short dimensions. This unit mode allows plotter space access on a percent of full scale basis.

UDUs are the units defined by your application. Statements which operate in the UDU mode can have values for the parameters directly in user units. Several statements are provided to allow you to define the units of measure for a particular application.

The unit of measure in the **Metric Unit** mode is the millimetre. This mode is implemented as a special case of UDUs, such that plotting statements executed in this mode put a line on the plotting area physically scaled to millimetres. This mode is useful where correspondence with physically measurable objects is desirable, as in drafting applications. In some devices such as the CRT and the thermal printer, metric units are only approximate.

The term **current units** refers to the last unit of measure mode that was set.

Most of the statements explained in this manual interpret their parameters according to the last unit-of-measure mode set (or current units); other statements interpret their parameters in one of the three modes (GDU, UDU or Metric).

Line Generation

Line Generation refers to the process of producing a line on the CRT, which is similar to drawing a line with a pen on an external plotter. Unlike the external plotter, the CRT has no actual pen. The CRT does have a point referred to as “the pen”, which when moved, produces a line if line generation is turned on (pen down). If line generation is turned off (pen up) no line is produced, but the point moves.

About the Examples

While all of the graphics statements in this manual are keyboard executable, perhaps the best way to familiarize yourself with the statements is to use the example programs. For your convenience, there are two types of examples which appear throughout the manual.

The first type of example is a “continuing example”. This example “grows” as each programming syntax is explained. The continuing example is used with most programming statements. It allows you to develop a complex graphics program by first typing in the initial program lines, and then adding each statement as it is explained. This provides you with a unified approach to each statement, as the effects of each statement can be seen as it is executed.

This “continuing example” is shown next. You should notice that there are gaps in the sequence of the line numbers. These gaps are intentional. Their function is to allow you to use the continuing example with a minimal amount of inconvenience in typing the program.

```

1    REM *** This is the continuing example (CONT)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
40  GRAPHICS ! Enables the CRT
90 Start_plot: !
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140  READ X_plot_point,Y_plot_point ! Obtains data coordinates
●150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280  READ Label$ ! READS data
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
480 Copies: !
500 END

```

Any syntax statements which consistently occur in the example, end with a zero as the last digit (such as 10, 20, 30). If a syntax statement appears which is not consistently used in the continuing example, its last digit is a five (as in 25, 35, 45). This way, line numbers which end in a five can be deleted as soon as possible after you have run the program. The line numbers ending in zero should be retained. This way you never have to type more than one or two additional program lines to the continuing example.

Not all of the statements immediately affect the plot in the continuing example. Therefore, a second type of example is used to demonstrate the explained syntax in a specific application. These applications are chosen to highlight the statement, rather than a specific graphics application. These examples can be found in Chapter 7, Additional Graphics Techniques. If you choose to alternate between the examples, you may wish to store the continuing example on tape, to eliminate the re-typing of it.

With all of the examples which appear in your System 45B Graphics manual, a ● appears to the left of the line number which uses the statement being explained.

The Appendices

Appendix A contains a glossary of terms which are used in this manual.

Appendix B contains the summary of the graphics programming statements.

Appendix C contains various items, such as the equipment supplied, graphics specifications for the CRT, and installation information for the HP 9872A Plotter.

Syntax Guidelines

The following general conventions apply to the statements which appear in this manual.

`Dot Matrix` All items appearing in Dot Matrix must be entered as shown.

[] All items in brackets are optional and explained in the accompanying text.

Customer Comments Card

Your comments are an important part in the creation of our manuals. In order to make it easier for you to let us know how you like your manuals we have included postage paid comments cards in the back of the manual.

If the comments card is missing, please address your comments to:

Hewlett-Packard Company
Desktop Computer Division
3404 East Harmony Road
Fort Collins, Colorado 80525 U.S.A.
Attn. Customer Documentation

Summary

Your System 45B can be used to draw plots of data and digitize data from plots. The plot is displayed on a graphics display device, such as the CRT. When the plot is displayed on the CRT, it is plotted on a 560 x 455 dot matrix with the plotted information being stored in a 32k byte R/W memory.

There are three units of measure for your System 45B Graphics system –

- Graphic Display Units (GDU s) where the shorter side of the plotting area is always equal to 100; the longer side of the plotting area is always equal to the shorter side times its ratio.
- User Defined Units (UDU s) where you can set limits on the plotting area for both the short and the long side.
- Metric units where the plotting area is physically scaled to millimetres. The CRT on the System 45B approximates the metric units rather than being exact. (20cm is actually 18.4 cm).

Simple Plotting Operations

page 12	• GRAPHICS (enables plotting to be viewed on CRT)
page 12	• EXIT GRAPHICS (returns CRT to alphanumeric raster)
page 14	• PEN (selects a pen to draw/plot/label with)
page 16	• LINE TYPE (selects one of 10 lines)
page 20	• PLOT (plots a line)
page 22	• PLOTTER IS (enables/specifies a plotting device)
page 26	• PLOTTER IS ON (enables a specified plotting device)
page 26	• PLOTTER IS OFF (disables a specified plotting device)
page 28	• GCLEAR (clears CRT & graphics R/W memory)
page 30	• FRAME (draws a border around current soft clip area)
page 32	• DUMP GRAPHICS (thermal printer printout of graphics R/W memory)
page 34	• PENUP (raises the pen)

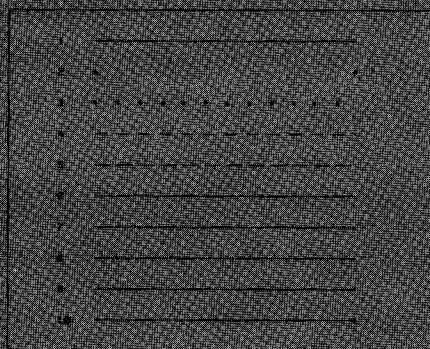
Terms

- Hard Clip Area** – The physical limits of the plotting device, beyond which no line can be drawn.
- Soft Clip Area** – The limits of the plotting device which restricts pen movement for lines drawn in UDUs.
- Clipping Area** – The area which restricts the pen movement whenever lines are drawn in UDUs.
- Plotting Coordinates** – The X,Y coordinate pair which specifies a plotting point.
- UDUs** – User Defined Units. Defined by the program to whatever X and Y units of measure which are convenient.
- GDU's** – Graphic Display Units. An X,Y reference system which at default defines the CRT to extend from X minimum = 0 to X maximum = 123.127753304, Y minimum = 0 to Y maximum = 100.

Default Conditions Set by PLOTTER IS

Graphics R/W memory is cleared
 Soft clip area = hard clip limits
 UDUs = GDUs
 UDUs are current units
 Pen #1
 Line type 1
 Characters
 Size = 3.3 GDU's
 Aspect ratio = 9/15
 Slant = 0°
 Label origin 1
 Label direction horizontal
 (left to right)

Line Types



Chapter 2

Plotting Boundaries

All plotting takes place within some type of boundary on the plotting device, whether the plotting device is the CRT or an external plotter. There are two boundaries which you are concerned with on your System 45B Graphics system –

Hard Clip Area

The **hard clip area** can be set to any size, up to and including the physical limits of the CRT's graphics raster. This is an area equal to 20 cm horizontally and 16.25 cm vertically on the CRT. On most plotters, the hard clip area is approximately equal to the platen size.

Soft Clip Area

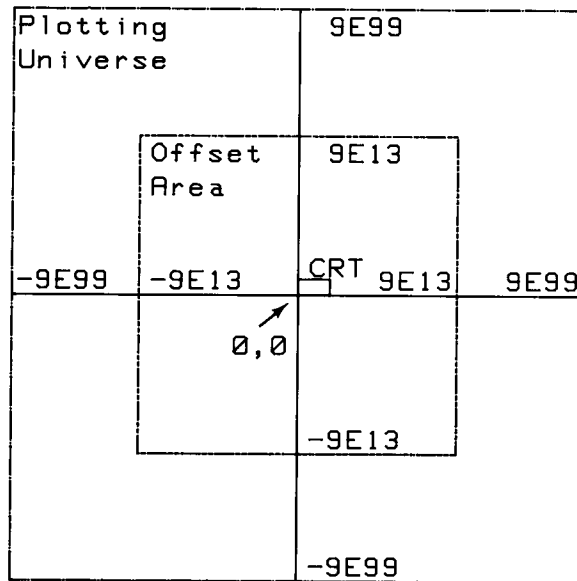
The **soft clip area** (also called the **clipping area**) is the boundary which actually limits the plotting (i.e. lines cannot be drawn outside of the soft clip area). The soft clip area can be set to any size, up to and including the hard clip limits of the plotting device.

Both of the soft and hard clip areas can be altered by various graphics statements which are explained in this manual. However, for simple plotting operations they are not needed. The statements which alter the soft and hard clip areas are explained in Chapter 3.

Plotting Coordinates

All points within the soft clip and hard clip areas can be referenced by means of a coordinate system. This coordinate system can be specified by various graphics statements which are explained in this manual. However, for simple plotting operations which are described in this chapter the following information should be sufficient.

Your System 45B uses a coordinate system scaled as shown –



The acceptable values for processing coordinates are from $-9E99$ to $+9E99$ along both the X and Y axes.

As you can see, the CRT is positioned at the intersection of the X and Y axes (0,0). However, you can offset the CRT to anywhere from $-9E13$ to $+9E13$ along both axes.

The coordinate system which is in effect at power-on is referred to as the “default” system. The coordinate system is a simple X(horizontal), Y(vertical) coordinate system with the following values –

	min X unit	min Y unit	max X units	max Y units
●CRT	0	0	123.127753304	100
●9872A	0	0	152	100
●Incremental	0	0	152	100

In effect, this means that unless you change the plotting coordinate system by means of a scaling statement (Chapter 3), the only points which can be plotted on the CRT are those which occur between 0 and 123.127753304 on the X axis, and 0 and 100 along the Y axis. Keep in mind that while only the coordinates which lie between these limits are plotted, data which exceeds these coordinates can still be calculated and processed by your System 45B.

GRAPHICS and EXIT GRAPHICS Statements

The `GRAPHICS` and `EXIT GRAPHICS` statements turn the CRT graphics raster on and off respectively. If you are plotting on the CRT and want to see what you are plotting, `GRAPHICS` must be executed. You can plot on the CRT while the alphanumeric raster is on and then switch on the graphics raster to see what you have plotted.

Syntax:

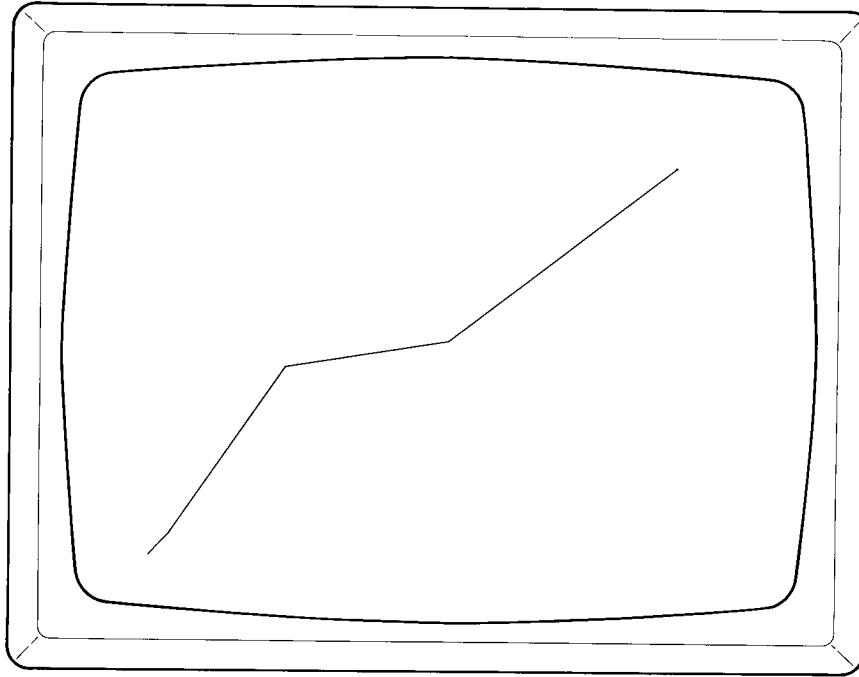
```
GRAPHICS
```

or

```
EXIT GRAPHICS
```

The graphics raster remains on the CRT until `EXIT GRAPHICS` is executed, an error occurs, or either a typing key or the recall key is pressed on the keyboard. If you accidentally get out of graphics, type `G R A P H I C S` `EXIT`, to get back into it.

External plotters do not use the Graphics R/W memory. Executing a `GRAPHICS` statement when a device other than the CRT is the plotter results in a random pattern being displayed as the CRT memory contents.

Example

```

1    REM *** This is the continuing example (CONT1)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
●40  GRAPHICS ! Enables the CRT
90 Start_plot: !
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140  READ X_plot_point,Y_plot_point ! Obtains data coordinates
150  PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280  READ Label$ ! READs data
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
480 Copies: !
●485 EXIT GRAPHICS ! Returns to alphanumeric raster
500 END

```

PEN Statement

The `PEN` statement allows you to select a specific pen. On the HP 9872A Plotter, this could be any of four different pens.

Syntax:

```
PEN pen number
```

The range of the pen number parameter is minus one (-1) through plus four ($+4$). Non-integer pen number parameters are rounded (up or down) to the nearest integer value.

Pen number zero (0) selects a blank pen. On the 9872A Plotter all pens are returned to their holders. When using a 9872A Plotter, it is a good idea to end all programs with a `PEN 0`. This helps keep the pens from drying out.

On external plotters that have more than one pen, the absolute value of the pen parameter is used to select one of four pens (possibly of different colors).

On the CRT, any positive pen number draws a line. A negative pen number selects an “eraser”. A line redrawn with a negative pen number is erased along with the intersecting points of any intersecting lines.

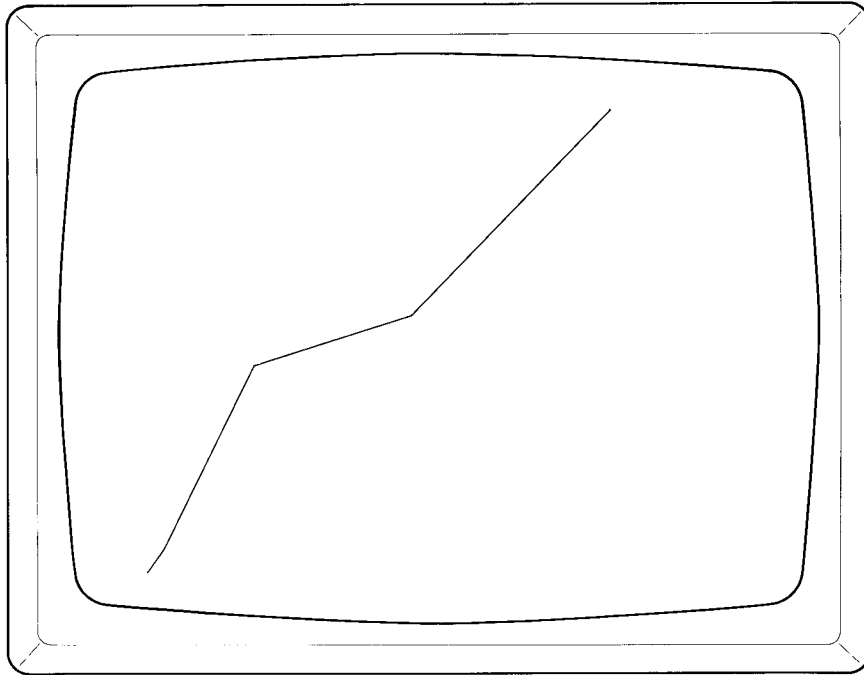
Jaggies

You may notice an occurrence called “Jaggies”. Jaggies are caused by the turning on of adjacent phosphor dots which do not lie in the same direction as the line being plotted. This becomes evident when you attempt to erase a line which has jaggies.

The proper way to erase a line using `PEN -1` is to re-draw or re-plot it from end point to end point in the same direction which it was drawn. This ensures that all of the jaggies are erased.

Executing a `PEN` statement with a negative pen number (the eraser) resets the `LINE TYPE` to type one, (refer to the next section of this manual).

Example



```

1    REM *** This is the continuing example (CONT2)
10 Set_up: !
●15  PEN 1 ! Selects PEN #1
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
40  GRAPHICS ! Enables the CRT
90 Start_plot: !
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280 READ Label$ ! READs data
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
480 Copies: !
500 END

```

LINE TYPE Statement

The `LINE TYPE` statement selects one of several solid or dashed line types and defines the length of the repeat pattern for the dashed or dotted lines.

Syntax:

```
LINE TYPE identifier number [, length]
```

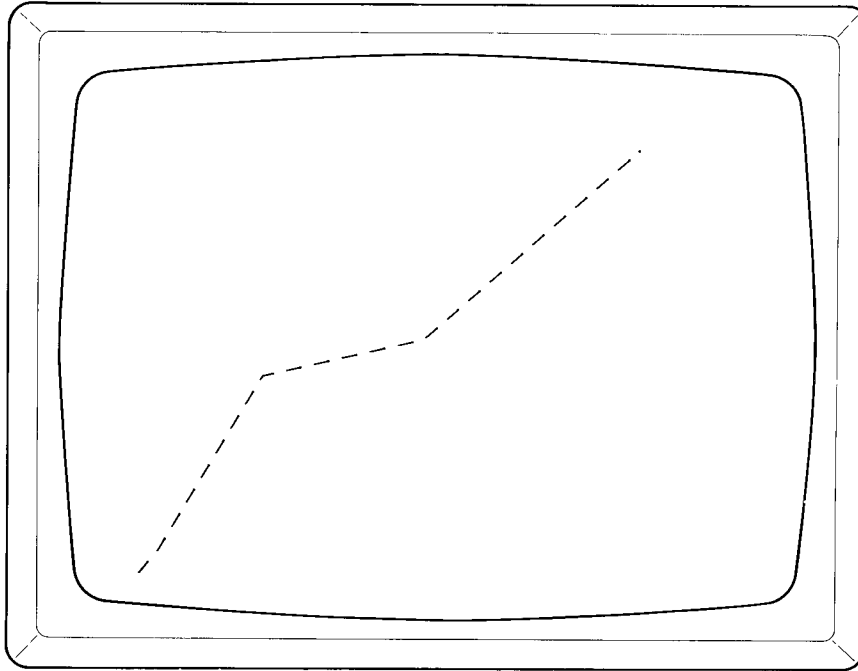
The identifier number parameter (integer value 1 through 10) selects one of the ten line types for plotting (see the next example). The default line type is type one.

The length parameter specifies the approximate length of the dashed line repeat pattern, in GDU's. The default length is 4 GDUs.

NOTE

The line type patterns generated by external plotters are slightly different from the line types drawn on the CRT.

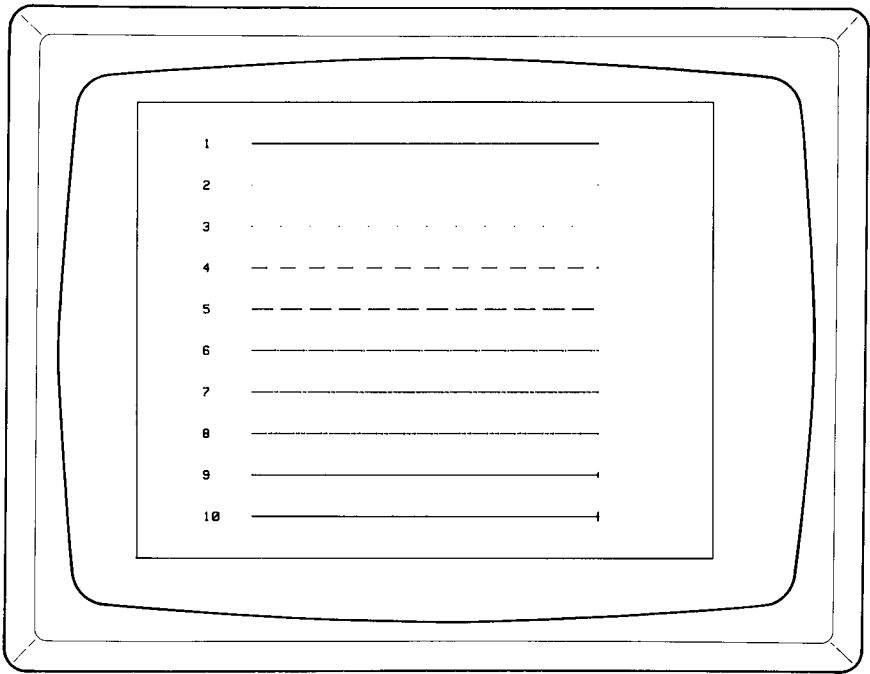
Examples



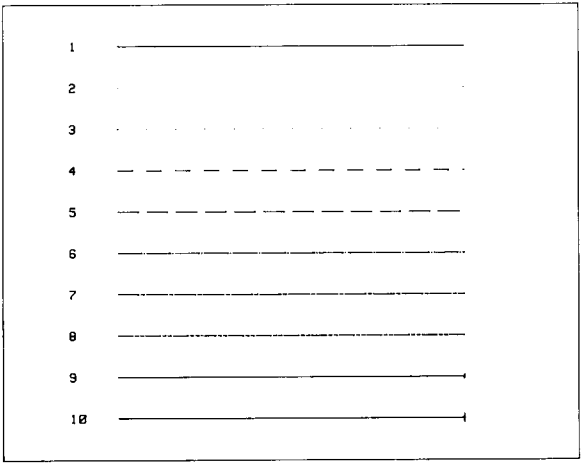
```

1    REM *** This is the continuing example (CONT3)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
40  GRAPHICS ! Enables the CRT
90 Start_plot: !
●120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140  READ X_plot_point,Y_plot_point ! Obtains data coordinates
150  PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160  NEXT Loop_index ! Repeats loop
170 Enhancements: !
●180 LINE TYPE 6 ! Specifies dashed LINE TYPE
●220 LINE TYPE 1 ! Specifies solid LINE TYPE
240 Label_axes: !
250  DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260  FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280  READ Label$ ! READs data
300  NEXT X_label ! Repeats loop
320  FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350  NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
480 Copies: !
500 END

```



CRT Line Types



9872A Line Types

```

1  REM ** This example demonstrates the LINE TYPE statement (LINE)
10 PLOTTER IS "GRAPHICS"      ! Specifies the CRT as the plotter
20 GRAPHICS                   ! Enables the CRT
30 FRAME                      ! Draws a box around the SOFT CLIP area
40 SCALE 0,10,-11,0          ! SCALES the LOCATE area to 10x11
50 LONG 2                     ! Centers LABELs on the LINE TYPE
51 REM ** LABEL and DRAW available LINE TYPEs
60 FOR Line=1 TO 10           ! Starts the loop
70 Labels: MOVE 1,-Line       ! MOVES to LABEL position
●80     LINE TYPE 1           ! Selects solid LINE TYPE for LABEL
90     LABEL Line             ! LABELs the number of the LINE TYPE
●100 Lines:LINE TYPE Line     ! Specifies LINE TYPE for line to be drawn
110     PENUP                 ! Raises the PEN
120     PLOT 2,-Line,-0       ! Moves to origin of line and lowers the PEN
130     DRAW 8,-Line          ! DRAWS a horizontal line
140 NEXT Line                 ! Repeats the loop
150 END

```

PLOT Statement

The `PLOT` statement provides data plotting with an optional pen control parameter. This statement moves the pen to the X,Y coordinate in current units using the current pen number and line type.

Syntax:

```
PLOT X parameter, Y parameter [, pen control]
```

The X and Y parameters are interpreted according to the current units mode.

The optional pen control parameter specifies the up or down pen movement, and defaults to plus one (the pen moves and then drops). The pen control parameter is interpreted as follows.

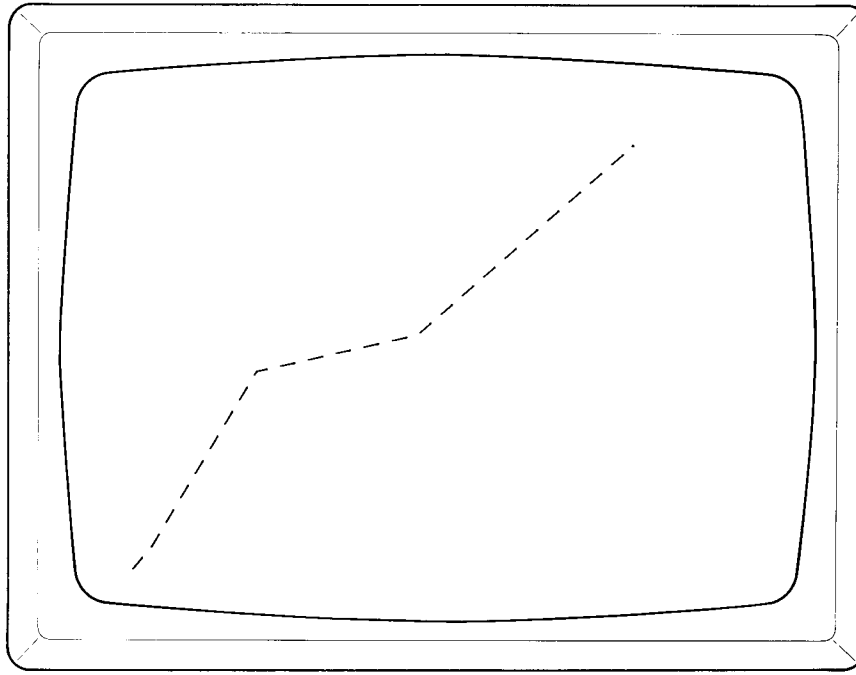
Odd =	Drop pen (pen down or start line generation)
Even =	Lift pen (pen up or stop line generation)
Positive =	Pen change after motion
Negative =	Pen change before motion

For example,

+1 or no parameter	Change pen location, then drop pen
2 or 0	Change pen location, then lift pen
-2	Lift pen, then change pen location
-1	Drop pen, then change pen location

This pen control parameter is specified in the `PLOT`, `IPLOT` and `RPLLOT` statements. Once you have set the pen control parameter, it remains in effect until it is changed by a default condition, or by another pen control parameter, a `PENUP` or a `MOVE`.

Example



```

1      REM *** This is the continuing example (CONT4)
10 Set_up:  !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
40  GRAPHICS ! Enables the CRT
90 Start_plot:  !
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140  READ X_plot_point,Y_plot_point ! Obtains data coordinates
●150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements:  !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
220 LINE TYPE 1 ! Specifies solid LINE TYPE
240 Label_axes:  !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280  READ Label$ ! READs data
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350 NEXT Y_label ! Repeats loop
360 Label_text:  !
370 DEG ! Change to degrees mode
480 Copies:  !
500  END

```

PLOTTER IS Statement

A `PLOTTER IS` statement is used to specify the select code and the type of plotting device to which successive plotter operations are directed.

Syntax:

```
PLOTTER IS [select code [, HP-IB device address],] plotter identifier string [, step
size [, number of pens, pen offset [, incremental plotter identifier] ] ]
```

Statement Parameters

The select code of the CRT (when used as a plotter) is 13 and is used for most examples in this manual.

The plotter identifier string (string variable or constant) specifies the type of plotting device, and therefore, the proper I/O operations which drive the plotter using the specified select code. External plotting devices (e.g. HP 9872A Plotter) set to a select code other than 13 can be connected to the computer. The plotter identifier string should specify the type of plotting device. The three plotter identifier strings and default select codes are:

Identifier String	Select Code
"GRAPHICS"	13
"9872A"	7,5
"INCREMENTAL"	5

CRT

The plotter identifier string, "GRAPHICS", specifies the CRT as the display device.

9872A

The plotter identifier string "9872A", can specify either the HP 9872A, 7245A or 7225A as the plotter which must be connected to the computer via the HP 98034A HP-IB Interface. Refer to Appendix C and the 98034A Installation and Service Manual for hookup information.

Incremental

The last four optional parameters of the `PLOTTER IS` statement provide a means for you to describe the features of the incremental plotter to the computer, so that the plotter can be driven properly. If you are not going to use an incremental plotter, do not use the last four optional parameters.

The default values for these parameters are:

Parameter	Default Value
step size	.254mm
number of pens	1
pen offset	0mm
incremental plotter identifier	1

The step size parameter must be specified if the other three parameters are to be specified. The assigned value of the step size parameter is plotting device dependent and must be obtained from the plotter's documentation.

If one of the last three parameters is to be specified, they all must be specified.

The number of pens parameter is used to tell the computer the number of available pens on the plotter.

The pen offset is used to specify the center-to-center distance between pens on multi-pen devices. If the plotter has only one pen, zero offset should be specified.

The plotter identifier parameter is used to specify one of several known incremental plotters that operate with the computer and the 98040A Interface. This parameter can also be used to tell the computer whether the offset is in the X or the Y direction. For more information regarding these four parameters refer to HP 98040A Incremental Plotter Interface Installation and Service manual.

Default Conditions

The `PLOTTER IS` statement sets the following default conditions when executed.

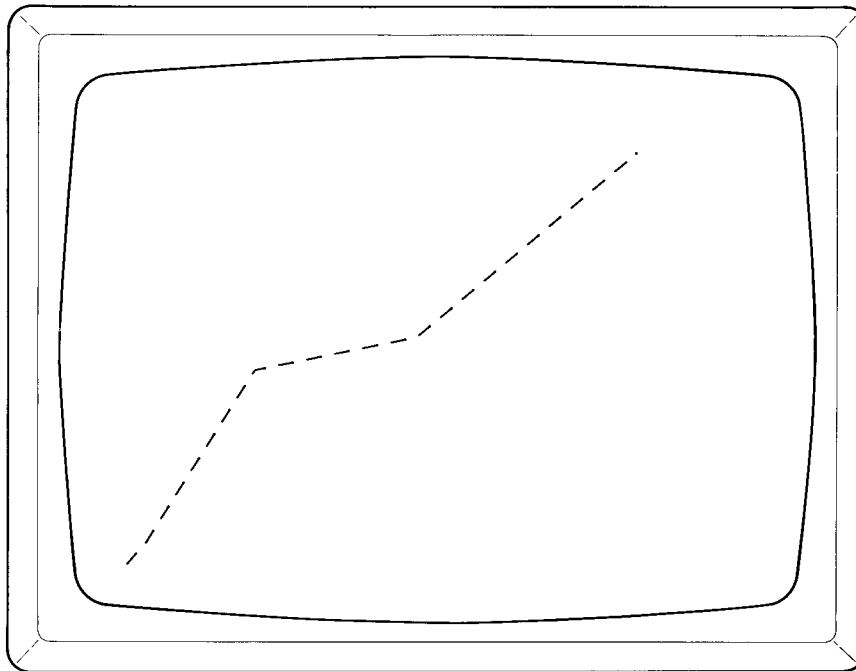
1. Activates the plotting device specified.
2. Sets UDU's (User Defined Units) as current units.
3. Sets UDU's (User Defined Units) equal to GDU's.
4. Reads hardware-set hard clip limits from the plotting device.
5. Defaults the soft clip area to the hard clip limits.
6. Clears the graphics display.
7. Selects pen one.
8. Selects line type one.
9. Selects the standard character size.
10. Selects label origin one.
11. Sets label direction as left to right.
12. Clears any error conditions.
13. Clears character count from any previous `LABEL` statement.

The default conditions can be altered as necessary by executing one or more of the plotter statements with the proper parameters from the keyboard or from within a program.

Note

Executing the `LIMIT` statement sets all of the previous default conditions except default condition number 6.

Example



```

1      REM *** This is the continuing example (CONT5)
10 Set up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
●30  PLOTTER IS 13,"GRAPHICS"           ! Specifies the CRT as the plotter
40  GRAPHICS                           ! Enables the CRT
90 Start_plot: !
120 LINE TYPE 4                        ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5              ! Begins loop to PLOT 5 points
140  READ X_plot_point,Y_plot_point    ! Obtains data coordinates
150  PLOT X_plot_point,Y_plot_point    ! PLOTS data coordinates
160 NEXT Loop_index                    ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6                        ! Specifies dashed LINE TYPE
220 LINE TYPE 1                        ! Specifies solid LINE TYPE
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12       ! Initializes loop
280  READ Label$                       ! READs data
300 NEXT X_label                       ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10       ! Initializes loop
350 NEXT Y_label                       ! Repeats loop
360 Label_text: !
370 DEG                               ! Change to degrees mode
480 Copies: !
500  END

```

PLOTTER. . .IS ON or IS OFF Statements

The `PLOTTER. . .IS ON` or `IS OFF` statement activates (ON) or deactivates (OFF) the plotting device specified by the select code. When activated, a plotting device responds to all of the plotting statements and when deactivated the plotting device does not respond. Executing a `PLOTTER. . .IS ON` for one select code automatically deactivates all other plotters. If all plotters are deactivated, the plotting statements are executed but are not directed to any plotter.

Syntax:

```
PLOTTER select code [ , HP-IB device address] IS ON
```

This statement activates the specified plotting device and deactivates all others. That is, all plotting statements are directed only to the specified device. This statement DOES NOT set any of the default conditions described under the `PLOTTER IS` statement.

Syntax:

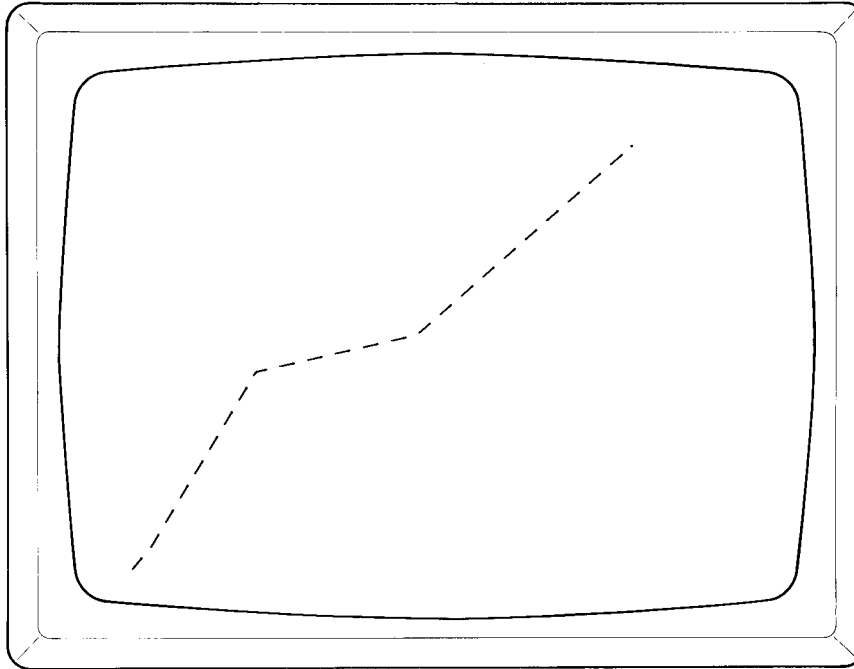
```
PLOTTER select code [ , HP-IB device address] IS OFF
```

This statement deactivates the plotter set to the specified select code. All plotting statements are executed but the specified plotter does not respond.

NOTE

If you do not have an HP-IB interface connected to the computer, an `ERROR 111` is generated by lines 5 and 15.

Example



```

1    REM *** This is the continuing example (CONT6)
●5    PLOTTER 7,5 IS ON                ! Directs plotting to the 9872A
10 Set up: !
●15    PLOTTER 7,5 IS OFF              ! Directs plotting away from the 9872A
20    DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30    PLOTTER IS 13,"GRAPHICS"         ! Specifies the CRT as the plotter
40    GRAPHICS                         ! Enables the CRT
90 Start_plot: !
120    LINE TYPE 4                     ! Specifies dashed LINE TYPE
130    FOR Loop_index=1 TO 5           ! Begins loop to PLOT 5 points
140        READ X_plot_point,Y_plot_point ! Obtains data coordinates
150        PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160    NEXT Loop_index                 ! Repeats loop
170 Enhancements: !
180    LINE TYPE 6                     ! Specifies dashed LINE TYPE
220    LINE TYPE 1                     ! Specifies solid LINE TYPE
240 Label_axes: !
250    DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260    FOR X_label=0 TO 144 STEP 12     ! Initializes loop
280        READ Label$                 ! READs data
300    NEXT X_label                     ! Repeats loop
320    FOR Y_label=0 TO 115 STEP 10     ! Initializes loop
350    NEXT Y_label                     ! Repeats loop
360 Label_text: !
370    DEG                             ! Change to degrees mode
480 Copies: !
500    END

```

GCLEAR Statement

The `GCLEAR` statement clears the graphics memory and the CRT screen of any previously plotted data. The `GCLEAR` statement does not reset or set the default parameters for any other statement.

Syntax:

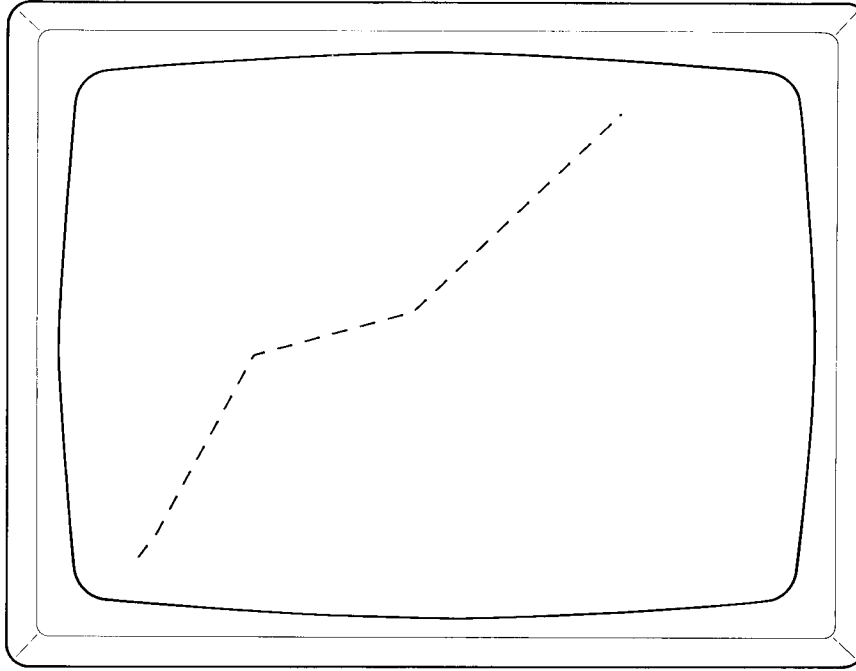
```
GCLEAR [paper advance]
```

The `PLOTTER IS` statement should normally be used between plots, rather than `GCLEAR`, since `GCLEAR` does not restore the default conditions.

On plotting devices with roll paper, the optional parameter allows you to advance the paper by a specified amount of millimetres.

On fixed paper plotters (e.g., HP 9872A) and on the CRT, the paper advance parameter is ignored.

Example



```

1      REM *** This is the continuing example (CONT?)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS"           ! Specifies the CRT as the plotter
40  GRAPHICS                           ! Enables the CRT
90 Start_plot: !
120 LINE TYPE 4                        ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5              ! Begins loop to PLOT 5 points
140  READ X_plot_point,Y_plot_point    ! Obtains data coordinates
150  PLOT X_plot_point,Y_plot_point    ! PLOTS data coordinates
160 NEXT Loop_index                    ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6                        ! Specifies dashed LINE TYPE
220 LINE TYPE 1                        ! Specifies solid LINE TYPE
240 Label_axes: !
250 DATA I,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12       ! Initializes loop
280  READ Label$                       ! READs data
300 NEXT X_label                       ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10       ! Initializes loop
350 NEXT Y_label                       ! Repeats loop
360 Label_text: !
370 DEG                                ! Change to degrees mode
480 Copies: !
● 485 GCLEAR                           ! Clears CRT & Graphics memory
500  END

```

FRAME Statement

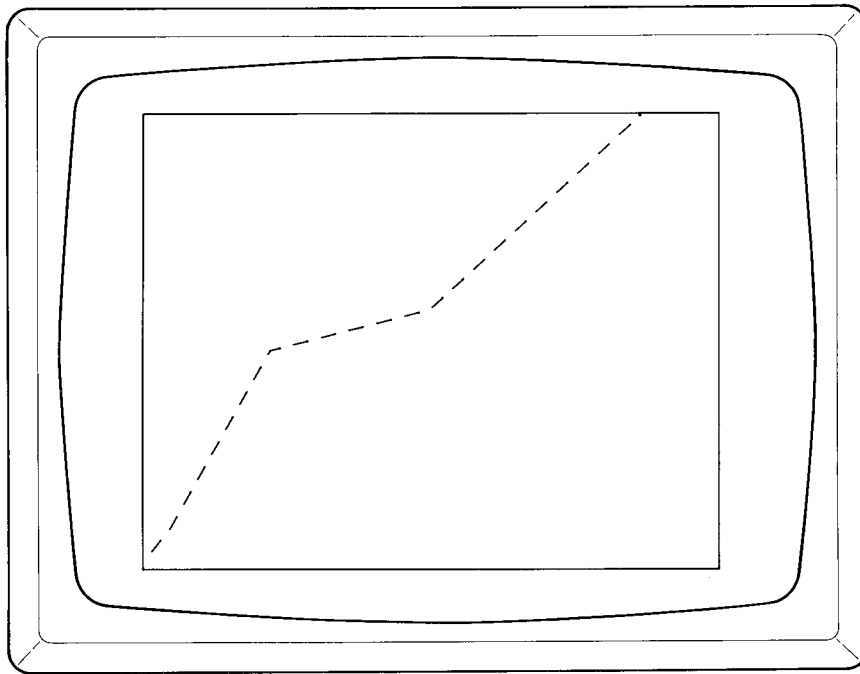
The `FRAME` statement draws a box around the current clipping area (the soft clip limits).

Syntax:

```
FRAME
```

The box is drawn using the current pen and line type around the current clipping area. The pen is positioned at the lower left corner of the frame after the operation is complete.

Example



```

1      REM *** This is the continuing example (CONT8)
10 Set_up:  !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS"           ! Specifies the CRT as the plotter
40  GRAPHICS                           ! Enables the CRT
90 Start_plot:  !
●100 FRAME                             ! Boxes the SOFT CLIP area
120 LINE TYPE 4                        ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5              ! Begins loop to PLOT 5 points
140  READ X_plot_point,Y_plot_point    ! Obtains data coordinates
150  PLOT X_plot_point,Y_plot_point    ! PLOTS data coordinates
160 NEXT Loop_index                    ! Repeats loop
170 Enhancements:  !
180 LINE TYPE 6                        ! Specifies dashed LINE TYPE
220 LINE TYPE 1                        ! Specifies solid LINE TYPE
240 Label_axes:  !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12       ! Initializes loop
280  READ Label#                       ! READs data
300 NEXT X_label                       ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10       ! Initializes loop
350 NEXT Y_label                       ! Repeats loop
360 Label_text:  !
370 DEG                               ! Change to degrees mode
480 Copies:  !
500  END

```

DUMP GRAPHICS Statement

The `DUMP GRAPHICS` statement allows a graphic display on the CRT to be copied onto the internal printer. You must have the optional thermal printer installed to execute this statement.

Syntax:

```
DUMP GRAPHICS [lower-bound, [upper-bound] ]
```

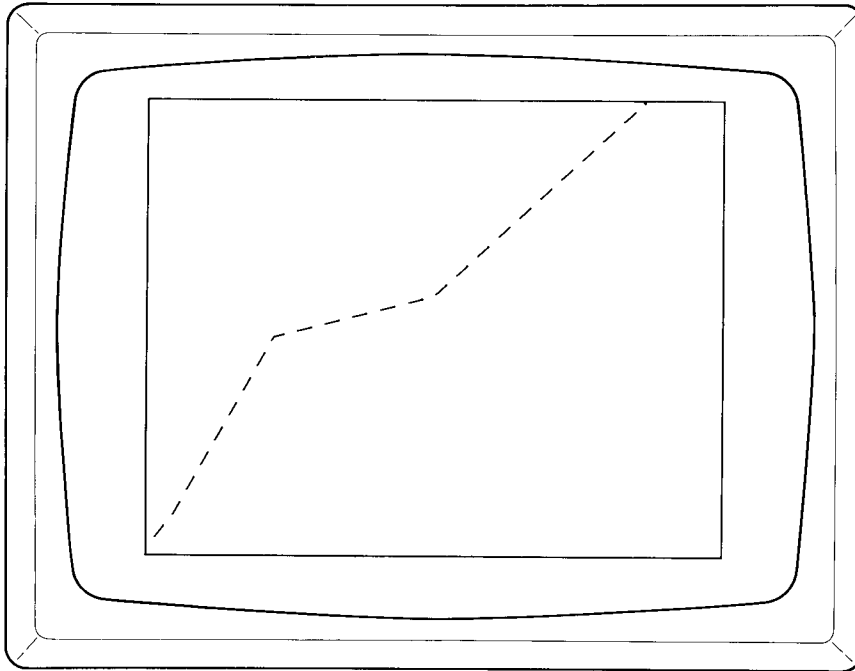
This statement can be executed by the program or from the keyboard while the graphics raster is off or on.

The two optional parameters allow you to copy any horizontal area displayed on the CRT by specifying in current units the upper and lower bounds of the area to be copied.

Without the second parameter, the top of the CRT is used as the upper bound.

Without the first and second parameters, the entire CRT display is copied onto the internal printer.

Example



```

1    REM *** This is the continuing example (CONT9)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
220 LINE TYPE 1 ! Specifies solid LINE TYPE
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280 READ Label$ ! READs data
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
480 Copies: !
● 490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

PENUP Statement

The `PENUP` statement lifts the pen.

Syntax:

```
PENUP
```

On plotting devices with actual pens (e.g. 9872A Plotter) this statement lifts the pen. The pen can then be moved without drawing a line. This allows you to avoid generating an ink blot on the paper after a plot has been finished or during a lengthy calculation.

On the CRT this statement turns off line generation. If the pen is moved (with the pen up) its new location is not apparent on the CRT, but the pen's coordinate values can be found by executing the `WHERE` statement (refer to Chapter 6).

The pen up or down status can be automatically controlled by using the `DRAW` and `MOVE` statements or the `PLOT`, `RPLLOT` and `IPLOT` statements with the optional pen control parameter.

Summary

The second chapter covered –

- simple plotting operations
- plotting boundaries
- hard and soft clip areas
- plotting coordinate system
- GRAPHICS
- EXIT GRAPHICS
- PEN
- LINE TYPE
- PLOT
- PLOTTER IS
- PLOTTER. . . IS ON
- PLOTTER. . . IS OFF
- GCLEAR
- FRAME
- DUMP GRAPHICS
- PENUP

In this chapter you learned how to enable the CRT, as well as other plotting devices, how to plot data based on an X-Y coordinate reference system, how to get a hard copy printout on System 45B's thermal printer and how to clear the graphic's memory.

Information concerning positioning and scaling of your plotting area is contained in the next chapter.

Positioning and Scaling Plots

page 42	•LIMIT (sets hard clip limits)
page 44	•CLIP (sets soft clip limits)
page 46	•UNCLIP (changes soft clip limits to hard clip limits)
page 50	•LOCATE (defines points to be scaled)
page 53	•SCALE (makes UDU's, which you can specify, the current units)
page 56	•SHOW (causes specified LOCATE area to have X dimensions = Y dimensions)
page 57	•MSCALE (makes millimetres the current units)
page 61	•SETGU (makes UDU's the current units)
page 61	•SETUU (makes UDU's the current units)
page 62	•RATIO (determines ratio of longer plotting side to shorter plotting side)

Terms

- **Hard clip limits** - The physical limits of the plotting device, beyond which no further movement.
- **Soft clip limits** - The limits of the plotting device which restricts pen movement for basic and digitized UDU's.
- **Current units** - The current X-Y reference which is in effect, may be either GDU's, UDU's or Metric (mm).

Statement	Effect on Hard Clip Limits	Effect on Soft Clip Limits	Effect on Scaling Points
PLLOTTER IE	Reads the hardware P1 and P2	Sets to hardware P1 and P2	Sets to hardware P1 and P2
LIMIT	Sets boundaries to those specified or digitized by the LIMIT statement	Sets to hard clip limits or digitized boundaries of the LIMIT statement	Sets to hard clip limits defined by the LIMIT statement
LOCATE	No effect	Sets to specified or digitized boundaries of the LOCATE statement	Sets to the points specified or digitized by the LOCATE statement
CLIP	No effect	Sets to boundaries specified or digitized by the CLIP statement	No effect
UNCLIP	No effect	Resets to hard clip limits	No effect

Summary of Scaling Statements

Scaling statements affect the area defined by the LOCATE rectangle.

Scaling statements set UDU's as the current units.

SCALE defines the UDU's.

SHOW causes specified area to have 1 unit of X = 1 unit of Y.

MSCALE specifies millimetres as UDU's.

Chapter 3

Overview

Basically, there are two things which you may wish to do with a plot –

- position the plot on the graphics device
- scale the plotting area to some dimensions

Positioning the plot is actually done with three statements –

- `LIMIT` (which defines the hard clip area)
- `CLIP` (which defines the soft clip area)
- `UNCLIP` (which changes the soft clip area to equal the hard clip area)

Scaling the plot is done with four statements –

- `LOCATE` (which defines the area that is being scaled)
- `SCALE` (which defines the X-Y coordinates along dimensions that you specify)
- `MSCALE` (which defines the X-Y coordinates to metric units (millimetres))
- `SHOW` (which defines one unit of X to equal one unit of Y)

Current Units

Before you can start positioning the plot, you have to know something about how your plotting device's X-Y coordinate system is referenced, or what its **current units** are. The term current units refers to whether the X-Y coordinate system in effect is in Graphic Display Units (GDU) or User Defined Units (UDU).

GDUs sets the shorter of the two axes (X or Y) equal to 100 units. The number of units in the longer axis then becomes equal to the ratio of the longer axis divided by the shorter axis times 100. The end result of this is that the shorter axis is equal to 100 units, while the longer axis is a value $= > 100$.

UDUs are defined by you when you use the `SCALE` statement or when you use the `MSCALE` statement. The `SCALE` statement allows you to select any numeric units for the axes, while the `MSCALE` statement defines the X and Y axes to be equal to millimetres.

Positioning

Positioning allows you to designate an area on the plotting device where the plot appears. This can be very useful for a number of considerations, among them are –

- leaving space for labeling operations
- placing several related plots on the same display or piece of paper
- aesthetic reasons

As previously mentioned, all plotting takes place within the **soft clip limits**. The default on the CRT sets the soft clip limits equal to the **hard clip limits**.

On some plotting devices, such as the HP9872A Plotter, you can manually change the hard clip limits by using the `P1` and `P2` keys. The hard clip limits can also be changed under program control by using the `LIMIT` statement. Remember, the hard clip limits can never exceed the physical limits of the plotting device.

However, the soft clip limits can be set less than, equal to or greater than the hard clip limits. This can be done by the `CLIP` statement. If your plotted coordinates exceed the physical limits of the plotting device and are within the soft clip limits, the System 45B can still compute your data. Once the plotted data returns to a point within the soft and hard clip limits, it can be plotted.

If you wish to change the soft clip limits back equal to the hard clip limits, you can use the `UNCLIP` statement.

Scaling

Scaling is the process of changing the X-Y coordinate reference system values. This is done when the X-Y reference system which is currently in effect is no longer applicable to your application (why plot on a 123x100 area when you need an area of 5x12?).

Scaling is accomplished by first defining which area of the plotting device is to be scaled. The defining can be done either by using the default condition of the hard clip area, or by using the `LOCATE` statement to specify an area.

The scaling is done with two statements –

- `SCALE`
- `MSCALE`

`SCALE` allows you to specify values which define the number of units into which X and Y axes are to be divided.

`MSCALE` divides the plotting area into metric units. On the CRT, this division is not exact due to the dot matrix of the CRT. This results in a 20cm value actually being equal to 18.4cm. However on the 9872A plotter, the metric values are exact.

The `SHOW` statement is similar to a scaling statement where $X_{min} = Y_{min}$ and $X_{max} = Y_{max}$. A figure is then plotted into the `LOCATE` rectangle, showing only as much as fits into the rectangle.

Offsets

The `LIMIT` and `MSCALE` statements also allow an offset distance in millimetres to be specified. The offset allows you to locate the plotting area away from the lower left hand corner of your CRT or 9872A plotter.

Other Statements

Some of the other statements which may prove useful in these applications are –

- **SETGU** (which makes GDUs the current units.)
- **SETUU** (which makes UDUs the current units.)
- **RATIO** (which computes the GDU ratio of the length of the X axis divided by the length of the Y axis.)

LIMIT Statement

The **LIMIT** statement allows you to define the “plotting area” as a subset of the physical plotting area by setting the hard clip limits. In effect you are describing the size and location of a piece of paper or a specified area of the CRT to the computer. The computer is NOT allowed to move the pen outside the specified area without executing a new **LIMIT** statement for the area.

This statement overrides any previously set or defaulted hard or soft clip values. If the parameters are specified outside the physical limits of the plotting device, **ERROR 113** is generated.

If the **LIMIT** statement is not executed in a program, the hard clip limits are automatically read from the specified plotting device. On the HP 9872A Plotter, the default hard clip area is somewhat smaller than the physical plotting area. The plotting area is scaled in GDUs when the **LIMIT** statement is executed. Nothing can be labeled or drawn outside the currently specified hard clip limits.

Syntax:

```
LIMIT [X min, X max, Y min, Y max]
```

The first two parameters normally specify the left and right limits and the second two parameters normally specify the lower and upper limits of the plotting area.

42 Positioning and Scaling Plots

Because you are describing a physical piece of paper or plotting area to the computer, the units must be actual units of measure. For this syntax the units of measure are always millimetres and normally the origin (point 0,0) is at the lower-left physical limits of the plotting device. The default upper-right limit values for the CRT and thermal printer are X max = 184.47, Y max = 149.8. Other plotting devices have their own default upper limit values.

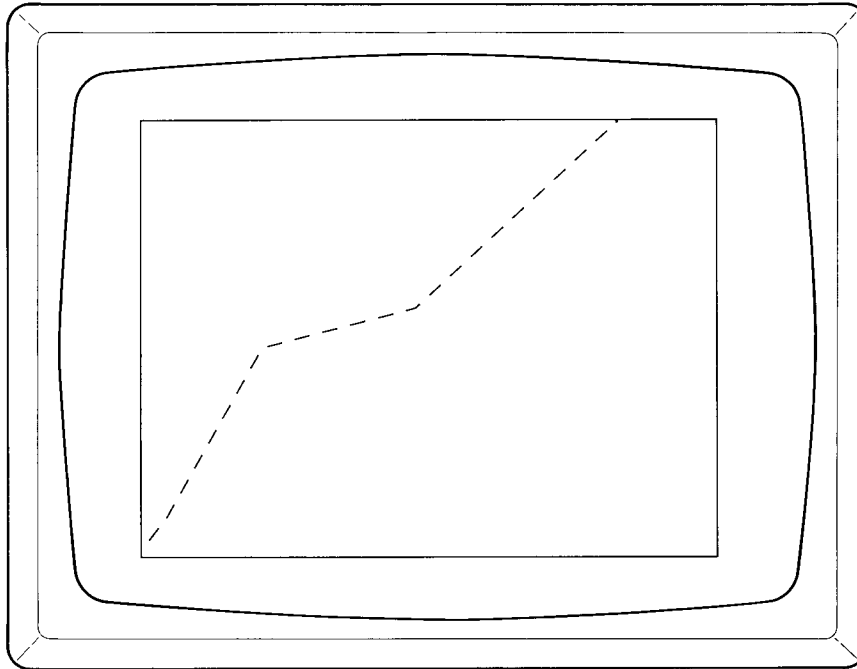
In the normal sequence of parameters, the first two parameters can be exchanged to produce a reflected plot across the Y axes (i.e. LIMIT X max, X min, Y min, Y max). Likewise the last two parameters can be exchanged to produce a reflected plot across the X axes (i.e. LIMIT X min, X max, Y max, Y min). Both the first two parameters and second two parameters can be exchanged, at the same time, to produce a reflected plot across the origin (i.e. LIMIT X max, X min, Y max, Y min). Chapter 7 contains more information on reflected plots.

Executing the LIMIT statement without parameters allows you to digitize two opposite corners to describe the plotting area. For further details about inputting parameters by digitizing, refer to the DIGITIZE Statement in Chapter 6.

The LIMIT statement is the software equivalent of manually setting the graph limits (keys and) on a 9872A Plotter.

When the HP 9872A is specified in the PLOTTER IS statement and the LIMIT statement is executed without parameters, the pen moves to the current and coordinates. This is so you can position the pen at your desired and coordinates which are input as the LIMIT X min, Y min, and X max, Y max points when the key on the plotter is pressed. These coordinates become the default hard clip limits for all subsequent programs until a new LIMIT statement is executed.

Example



```

1      REM *** This is the continuing example (CONT11)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
●50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
220 LINE TYPE 1 ! Specifies solid LINE TYPE
240 Label_axes: !
250 DATA I,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280 READ Label$ ! READs data
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

CLIP Statement

The `CLIP` statement defines the soft clip boundaries. This allows the soft clip boundaries to be moved from their default location set by the `PLOTTER IS`, `LIMIT` or `LOCATE` statements. Soft clipping affects lines plotted in UDU's, but has no effect on lines plotted in GDU's or as labels.

Syntax:

```
CLIP [X min, X max, Y min, Y max]
```

The parameters are interpreted according to current units of measure.

The first two parameters specify the left and right boundaries and the last two parameters specify the lower and upper boundaries of the clipping area.

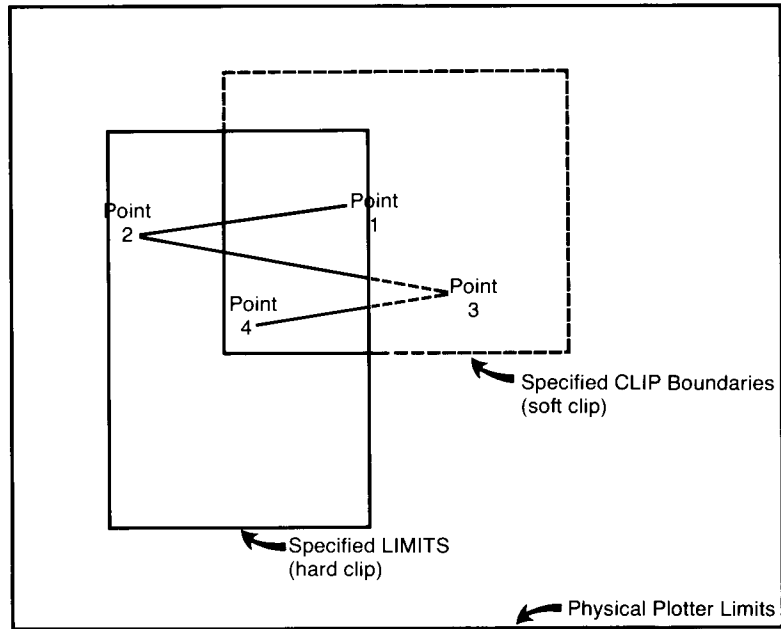
Lines plotted from inside this area to points outside the clip boundary extend no farther than the boundary. Lines plotted outside the `CLIP` boundaries are not drawn (unless drawn in GDUs).

Executing the `CLIP` statement without parameters allows you to specify the area by digitizing two opposite corners (normally the lower-left and upper-right corners).

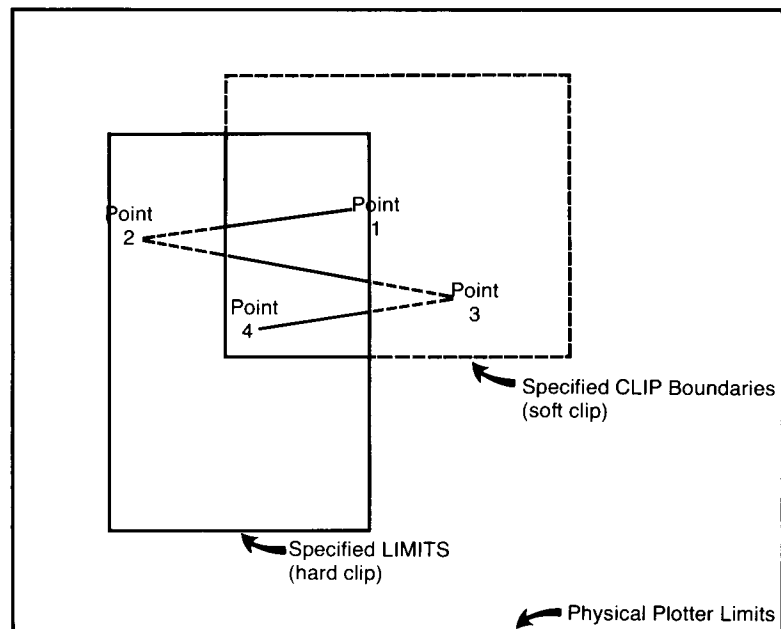
The soft clip limits can be turned off (by `UNCLIP`) or may be set beyond the hard clip limit. In either case and in the case of plotting commands in GDUs only, the hard clip limit (set by the `LIMIT` statement or defaulted by the `PLOTTER IS` statement) is used as the soft clip area.

Executing `PLOTTER IS`, `LIMIT`, `LOCATE` or `UNCLIP` reinstates various default values including the soft clip boundaries.

When `CLIP` is used to set the soft clip boundaries such that the soft clip rectangle is partially in and partially out of the hard clip limits, UDU plotting is clipped at the hard clip limits.



Plotting in GDUs



Plotting in UDUs

These two figures show the clipping action to expect when drawing a line from point 1, to point 2, to point 3 and then to point 4. The rectangles represent clipping boundaries or limits for the associated statements (LIMIT and CLIP).

Notice that the line drawn in the first figure (plotted in GDU's) is clipped only by the hard clip limits, and the line drawn in the second figure (plotted in UDU's) is clipped at the soft (CLIP) boundaries.

Another feature of CLIP is that you can specify a soft clip area greater than the hard clip area. The System 45B calculates the plots outside the hard clip area, and the plotted lines re-enter the soft clip area at the appropriate points.

In effect, the hard clip area then becomes a "window", while the soft clip area is the known plotting area. If you wished, you could set the hard clip limits to be the acceptable parameters of a test, while the soft clip limits could be the results of the test. Any line types which were plotted could then represent satisfactory test results.

UNCLIP Statement

The UNCLIP statement sets the soft clip boundaries equal to the hard clip limits. This allows you to draw lines anywhere in the plotting area defined by the LIMIT statement while in UDUs. It is primarily used whenever a LOCATE or CLIP statement has changed the soft clip area from the default hard clip area.

Syntax:

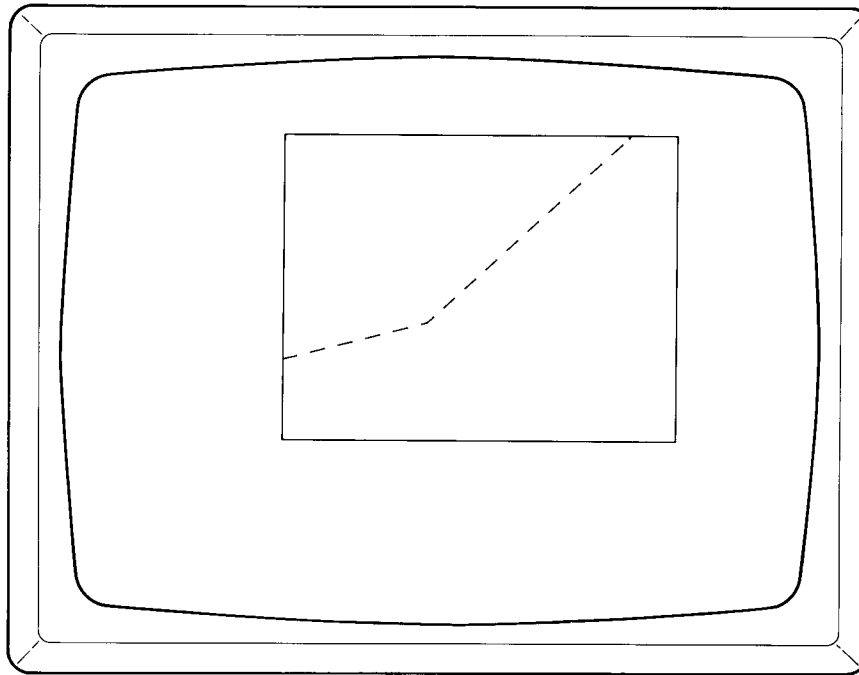
```
UNCLIP
```

Clipping is set by executing PLOTTER IS, LIMIT, LOCATE, or CLIP statements. SCALE and SHOW do not affect the clipping limits.

NOTE

Executing an UNCLIP statement **does not** override the scaled area defined by a LOCATE statement.

Example



```

1      REM *** This is the continuing example (CONT13)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
● 210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280 READ Label$ ! READs data
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

LOCATE Statement

The `LOCATE` statement allows you to define an area which the `SCALE` statement dimensions or the `SHOW` statement fills. The `LOCATE` statement also defines and resets the default soft clip boundaries.

Syntax:

```
LOCATE [X min, X max, Y min, Y max]
```

The first two parameters specify the left and right boundary limits and the last two parameters specify the lower and upper boundary limits.

The units are **always** in Graphic Display Units (GDU) with the origin (GDU point 0,0) normally at the lower-left corner of the plotting area (or as defined by the `LIMIT` statement).

The length of a GDU is equal to $1/100$ (one percent) of the length of the shorter side of the plotting area. Therefore, the length of the shorter side is always 100 GDU's. The length of the longer side of the plotting area is something greater than 100 depending on the plotting area aspect ratio. The aspect ratio can be found using the `RATIO` function as shown later.

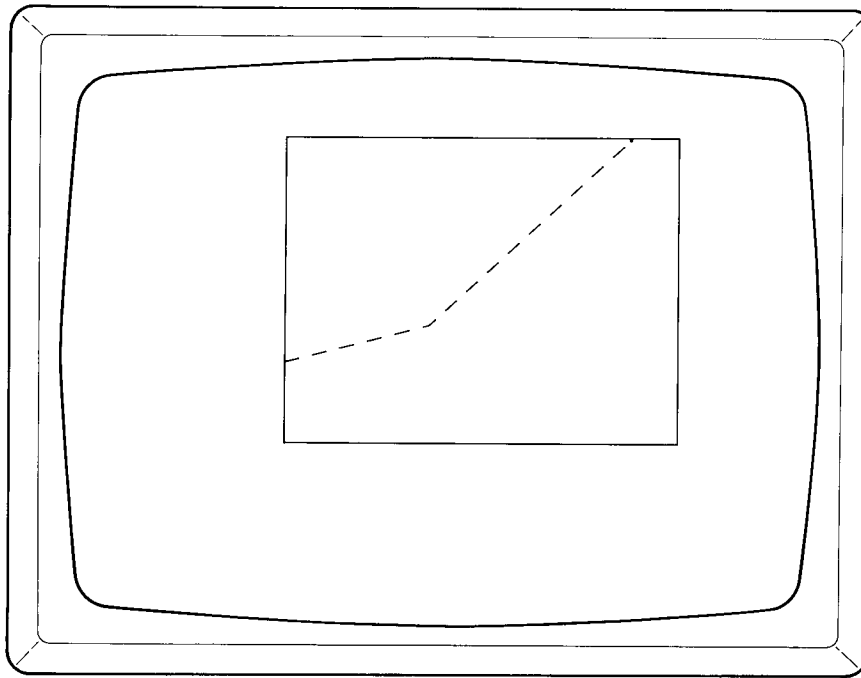
Executing the `LOCATE` statement without parameters allows you to specify the area by digitizing two opposite corners (the lower-left and upper-right corners).

All lines drawn while in the UDUs mode and after executing the `LOCATE` statement do not extend beyond the soft clip boundaries. The pen can be positioned between the soft clip boundaries and the hard clip limits to draw labels, but plotted lines do not extend into this area. Plotting in GDUs, you can always draw lines anywhere within the hard clip area.

The default physical length of the shorter side of all plotting devices is 100 GDUs but the default length of the longer side is device dependent. For the CRT, the default length of the longer side is the length/height aspect ratio times the height of 100 GDUs ($559 \div 454 \times 100 = 123.127753304$).

The `SCALE` and `SHOW` statements assign user definable units of measure (UDU's) to the area specified by the `LOCATE` statement. The `SCALE` and `SHOW` statements act on this area in two very different ways as can be seen in the next two sections of this manual.

Example



```

1    REM *** This is the continuing example (CONT14)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140  READ X_plot_point,Y_plot_point ! Obtains data coordinates
150  PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280  READ Label$ ! READs data
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

Statement	Effect on Hard Clip Limits	Effect on Soft Clip Limits	Effect on Scaling Points
PLOTTER IS	Reads the hardware P1 and P2.	Sets to hardware P1 and P2.	Sets to hardware P1 and P2.
LIMIT	Sets boundaries to area specified or digitized by LIMIT statement.	Sets to hard clip limits or digitized boundaries of the LIMIT statement.	Sets to hard clip limits defined by LIMIT statement.
LOCATE	No effect	Sets to specified or digitized boundaries of the LOCATE statement.	Sets to the points specified or digitized by the LOCATE statement.
CLIP	No effect	Sets to boundaries specified or digitized by the CLIP statement.	No effect
UNCLIP	No effect	Resets to hard clip limits.	No effect

Scaling the Plotting Area

Once you have specified the plotting area to be scaled, you may wish to dimension, define, or scale it in units which are suitable to your particular application.

The following statements can be used to define the located area –

- SCALE
- SHOW
- MSCALE

SCALE Statement

The `SCALE` statement defines minimum and maximum values of X and Y for the plotting area as specified by the `LOCATE` statement. This allows you to specify your own units for plotting.

Syntax:

```
SCALE X min, X max, Y min, Y max
```

This statement automatically sets the current units to the User Defined Units (UDU's) mode.

The first two parameters specify the values represented by the left and right boundary of the area specified by the `LOCATE` statement. The last two parameters specify the values represented by the lower and upper boundary in a similar manner.

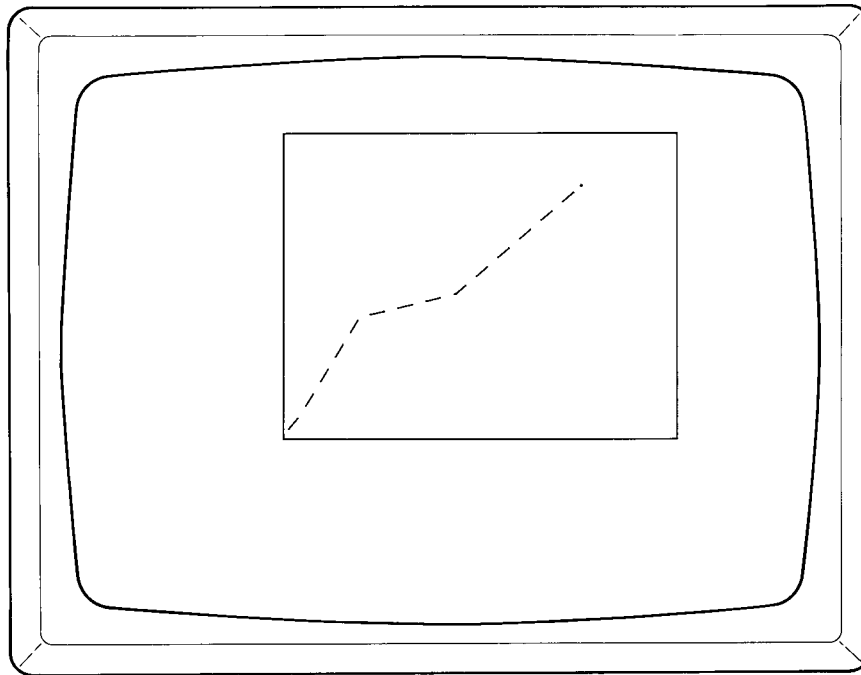
For example, the first two parameters of the `SCALE` statement could specify the left edge of the plotting area as -20 and the right edge as 30 . This has the effect of dividing the horizontal plotting distance into 50 units ($30 - (-20) = 50$). The last two parameters could specify different values and therefore a different scale for the vertical direction. These units can be used to represent distance, volume, time or whatever units your specific problem requires. The scaling factors for the X and Y directions are completely independent of each other. Thus, plots are stretched or shrunk independently in the X and Y direction to fit the plotting area (anisotropic scaling). Note that this is not the case with the `SHOW` statement.

For example, if you wanted to plot the average annual rainfall at a weather station for a 10 year period, the `SCALE` statement might look like this:

```
SCALE 1966, 1976, 0, 20
```

where the left edge represents the year 1966 and the right edge represents 1976. Rainfall would be plotted in the Y direction in units of depth (e.g. inches). This allows the data to be plotted in years and inches (e.g. 1976, 7) directly on the plotter.

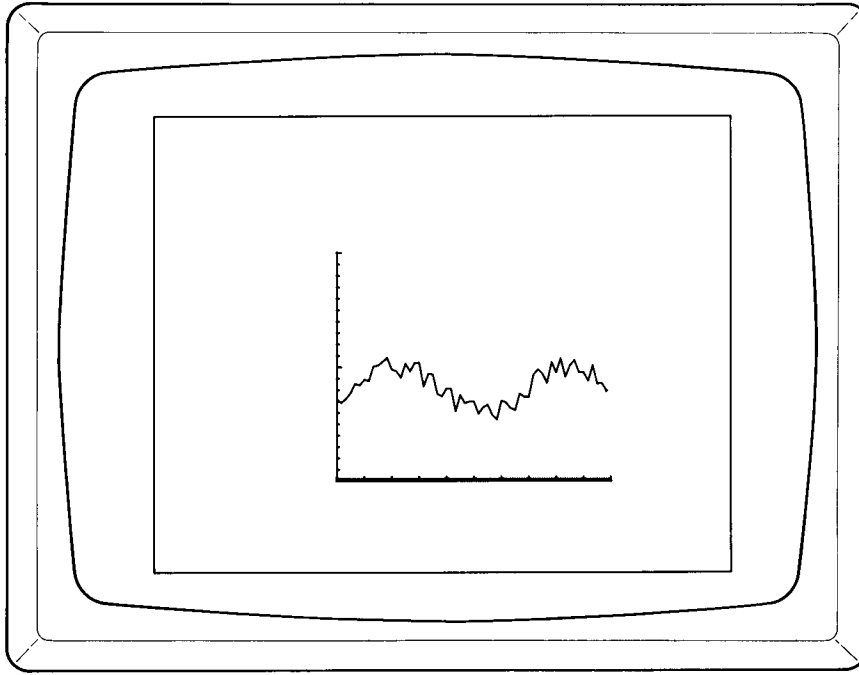
Examples



```

1    REM *** This is the continuing example (CONT15)
10 Set up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
●80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140  READ X_plot_point,Y_plot_point ! Obtains data coordinates
150  PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160  NEXT Loop_index ! Repeats loop
170 Enhancements: !
180  LINE TYPE 6 ! Specifies dashed LINE TYPE
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220  LINE TYPE 1 ! Specifies solid LINE TYPE
240 Label_axes: !
250  DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260  FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280  READ Label$ ! READs data
300  NEXT X_label ! Repeats loop
320  FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350  NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
480 Copies: !
490  DUMP GRAPHICS ! Hard copy of CRT display
500  END

```



```

1  REM ** This is the SCALE example (SCALE)
2  REM ** In this example, the X axis is SCALED to represent the
3  REM ** years from 1966 to 1976. The Y axis is SCALED to represent
4  REM ** from 0 to 20 inches of rainfall.
5  REM *****
10  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20  GRAPHICS                ! Enables the CRT
30  FRAME                   ! Draws a FRAME around the SOFT CLIP area
40  LOCATE 40,100,20,70     ! LOCATE an area within the SOFT CLIP area
●50  SCALE 1966,1976,0,20    ! SCALES the LOCATE area to 10x20
60  AXES 1/12,1,1966,0,12,10 ! Draws AXES for the plot
61  REM ** The next section PLOTS using random data
62  REM ** lines 80,90,100 generate and PLOT the random data
63  REM *****
70  MOVE 1966,0             ! MOVE to lower left corner of plot
80  FOR Random_data=1966 TO 1976 STEP 1/6
90      PLOT Random_data,RND*2+7+2*SIN(Random_data)
100  NEXT Random_data
110  END

```

SHOW Statement

The `SHOW` statement displays an area within the `LOCATE` rectangle where one unit of X equals one unit of Y.

Syntax:

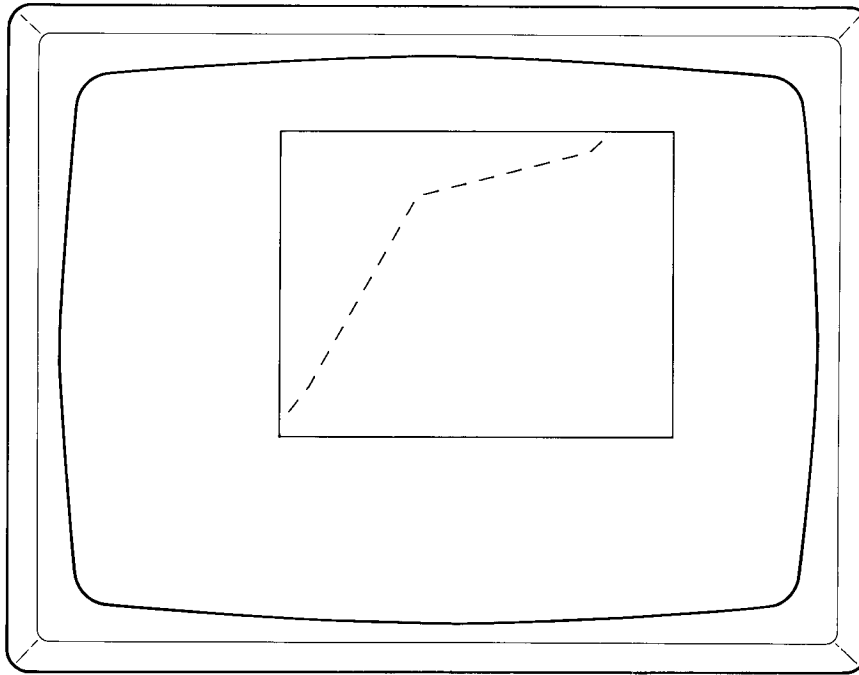
```
SHOW X min, X max, Y min, Y max
```

The first two parameters specify the acceptable bounds for the X direction and the last two parameters specify the acceptable bounds for the Y direction.

The `SHOW` statement defines an area that is stretched or shrunk proportionately in the X and Y directions so that the specified area fits into the plotting area defined by default or by the `LOCATE` statement. The aspect ratio of the units is forced to be equal to one (1) by the `SHOW` statement. That is, one unit of measure in the X direction is equal to one unit of measure in the Y direction (isotropic scaling).

The `SHOW` statement would not normally be used when you want to plot in two different units of measure, as in the previous example for the `SCALE` statement where inches of rainfall and years were used as units of measure. But `SHOW` should be used when you want to plot something like a map where one mile along the X axis should equal one mile along the Y axis.

Example



```

1      REM *** This is the continuing example (CONT16)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALEs the LOCATE area to 144x120
85  SHOW 0,80,0,60 ! SHOW's area as 1 unit of X= 1 unit of Y
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280 READ Label$ ! READs data
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

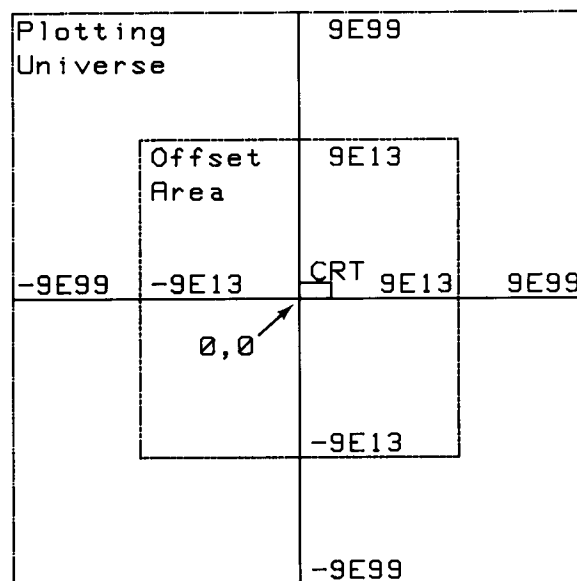
MSCALE Statement

The `MSCALE` statement sets millimetres as user units and defines the origin. This mode is very useful where correspondence with physically measurable objects is desirable, as in drafting and mapping applications. On the CRT and the thermal printer, metric units are only approximate (i.e. .033 cm per dot).

Syntax:

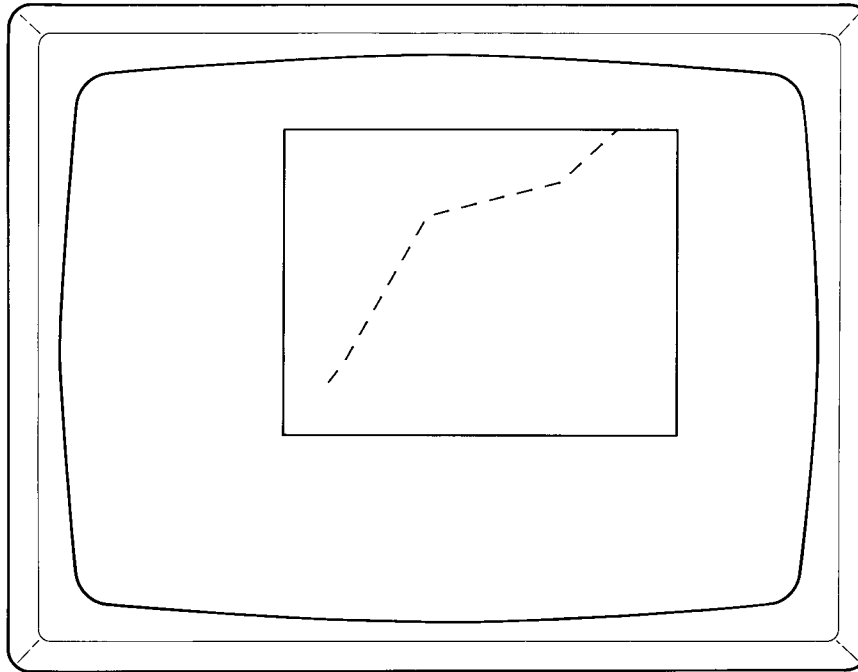
```
MSCALE X offset, Y offset
```

`MSCALE` specifies the current user units as millimetres. Once the `MSCALE` statement has been executed, the soft clip area is defined as follows:



with the CRT displaying only the area shown. The offset position allows you to position the "window" of the CRT anywhere within the range of from $-9E13$ to $9E13$.

Example



```

1    REM *** This is the continuing example (CONT17)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
●75  MSCALE 10,10 ! Sets SOFT CLIP in mm with 10x10 offset
80  ! SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100  FRAME ! Boxes the SOFT CLIP area
120  LINE TYPE 4 ! Specifies dashed LINE TYPE
130  FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140  READ X_plot_point,Y_plot_point ! Obtains data coordinates
150  PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160  NEXT Loop_index ! Repeats loop
170 Enhancements: !
180  LINE TYPE 6 ! Specifies dashed LINE TYPE
210  UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220  LINE TYPE 1 ! Specifies solid LINE TYPE
240 Label_axes: !
250  DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260  FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280  READ Label$ ! READs data
300  NEXT X_label ! Repeats loop
320  FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350  NEXT Y_label ! Repeats loop
360 Label_text: !
370  DEG ! Change to degrees mode
480 Copies: !
490  DUMP GRAPHICS ! Hard copy of CRT display
500  END

```

More About Plotting Units

The System 45B gives you the choice of using either the graphic display units (GDUs) or the user defined units (UDUs).

The GDUs are always dimensioned with the shorter side of the hard clip area (either horizontal or vertical) equal to 100. The longer side then becomes equal to 100 times the ratio of the longer side divided by the shorter side.

GDUs allow plotter space access on a percent of full scale basis. Some people prefer to do their plotting in UDUs and then switch back to GDUs to position the pen for labeling their plots.

User defined units are defined by the most recently executed `SCALE`, `SHOW`, or `MSCALE` statement.

The following statements interpret their parameters according to the current units mode selected:

<code>MOVE</code>	<code>DRAW</code>	<code>PLOT</code>	<code>RPLLOT</code>
<code>IFLOT</code>	<code>CLIP</code>	<code>AXES</code>	<code>GRID</code>
<code>WHERE</code>	<code>POINTER</code>	<code>DIGITIZE</code>	<code>CURSOR</code>

SETGU and SETUU Statements

Several graphic statements (as indicated) interpret their parameters using the units currently selected. These two statements provide the means to change the units mode.

Syntax:

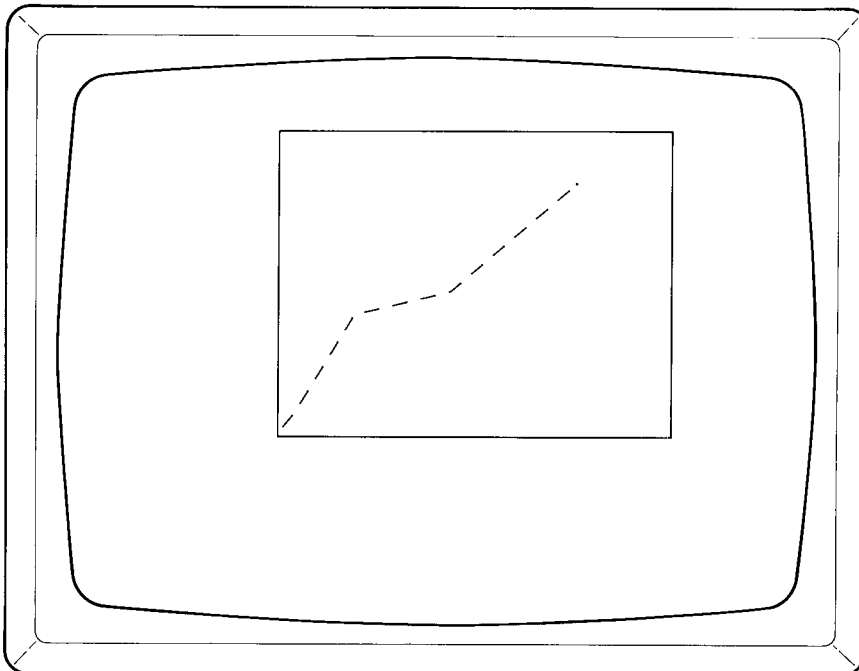
SETGU or SETUU

SETGU sets the current units to graphic display units. SETUU sets the current units to user defined units. UDU's are also set by the PLOTTER IS and the LIMIT statements. The default UDU's are defined to be equivalent to GDU's until a SCALE, SHOW or MSCALE statement is executed.

GDUs are primarily intended to position the pen on the plotting device (i.e., the plotting area as specified by the LIMIT statement).

UDUs are defined by the most recently executed SCALE, SHOW or MSCALE statement.

Example



```

1      REM *** This is the continuing example (CONT18)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280 READ Label$ ! READs data
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
●380 SETGU ! Change back to Graphic Display Units
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

RATIO Function

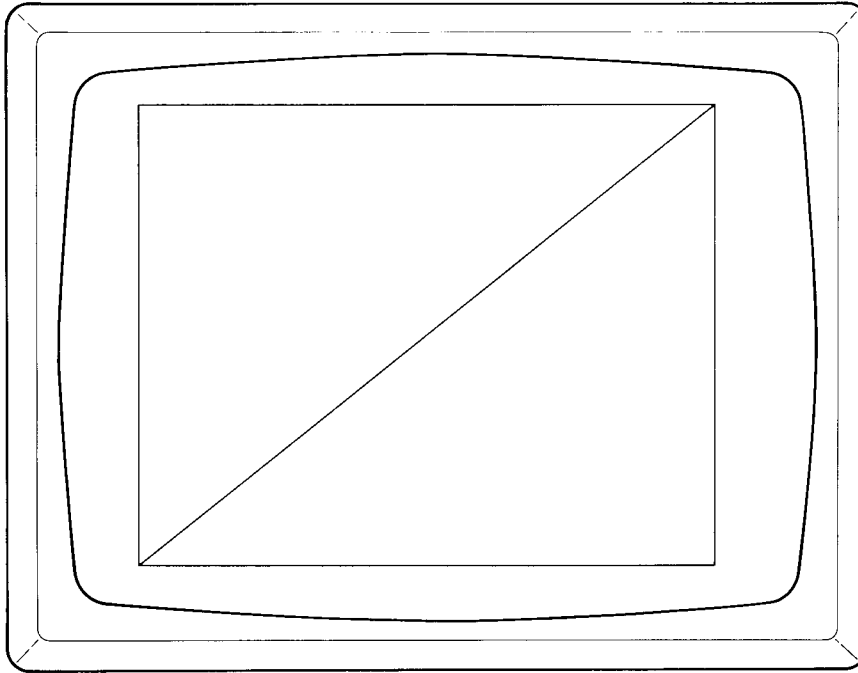
The **RATIO** function returns a value equal to the ratio of the physical dimension of the hard clip limits; that is, the X dimension divided by the Y dimension.

Syntax:

RATIO

The function can be used to find the maximum length of the longer side of the hard clip area. If the X axis is the longer axis, the ratio is >1 ; if the X axis is the shorter axis, the ratio is <1 .

The following program always moves the pen to the opposite corner of point 0, 0 (in GDU's) regardless of the size or shape of the plotting area.

Example

```

1  REM ** This is the RATIO example (RATIO)
10 ! This program DRAWS a line to the opposite corner from GDU point 0,0
20 ! giving you the maximum values of the plotter limits
30 ! in GDU'S. NOTE: The minimum value is always 0,0
40 !
50 PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT as the plotter
60 GRAPHICS                     ! Enables the CRT
70 LIMIT 0,184.47,0,149.82     ! Uses the full CRT area
80 !
90 FRAME                        ! FRAMES the SOFT CLIP area
100 MOVE 0,0                    ! MOVES to point 0,0 in GDU's
110 !
120 ! ***** Compute max & min values (GDU's) of the size of the plotting area
●130 Xgdu_max=100*MAX(1,1/RATIO)
●140 Ygdu_max=100*MAX(1,1/RATIO)
150 !
160 DRAW Xgdu_max,Ygdu_max      ! DRAWS a line to the opposite corner
170 END

```

Summary

The following graphics programming statements were covered in this chapter —

- LIMIT
- CLIP
- UNCLIP
- LOCATE
- SCALE
- SHOW
- MSCALE
- SETGU
- SETUU
- RATIO

In this chapter you learned what current units are, how to position your plots on your graphics device, and how to scale those plots on your graphics device so the data which is plotted can be easily interpolated.

The next chapter covers some plotting enhancements, such as relative and absolute plotting, changing the direction of your plots, and laying out your plots by marking your X and Y axes or by plotting on a grid.

Plotting Enhancements

- page 67 • **AXES** (draws an X and / or Y axis)
- page 70 • **GRID** (divides the plotted area into a grid reference)
- page 72 • **MOVE** (moves the pen)
- page 74 • **DRAW** (draws a line)
- page 76 • **RPLOT** (plots from a fixed reference point)
- page 78 • **IPLOT** (plots from the last plotted point)
- page 81 • **PDIR** (plots at an angular tangent)

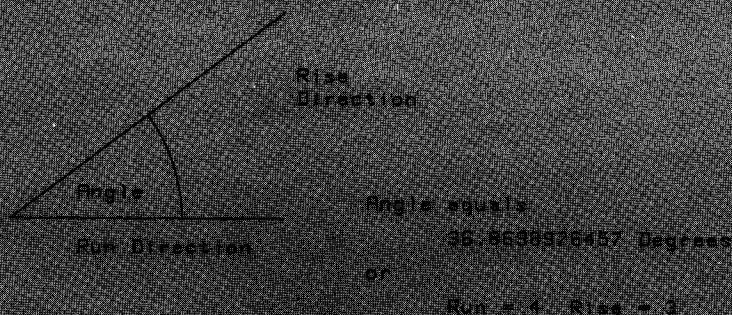
Terms

- **Absolute Plotting** – Plotting to a coordinate which has its X and Y values specified in the current user units.
- **Relative Plotting** – Plotting which specifies plotting from an origin rather than to a specific X,Y coordinate.
- **Origin** – The coordinate point at which a plotting operation begins.
- **Plotted Point** – The point which has been plotted or drawn to.

Pen Control Parameters

- Odd = Drop pen (pen down or start line generation)
- Even = Lift pen (pen up or stop line generation)
- Positive = Pen change after motion
- Negative = Pen change before motion

Implementing an Angular Tangent



Chapter 4

Overview

There are other statements which enable the plotting of data. You can use the `MOVE` statement to move the pen; you can draw a line with the `DRAW` statement.

Your plotting can be done in either of two ways, absolute or relative.

Absolute plotting uses a specific X-Y coordinate that is being plotted to. The `PLOT` statement does this, as does the `DRAW` statement.

Relative plotting bases its plot on an **origin** point. The origin can be the same place for successive `RPLLOT` statements, or it can be the point which was plotted to last as in the `IPLOT` statement. In effect, the `RPLLOT` statement references all of its lines to a local origin. The `IPLOT` statement's origin changes from plotted point to plotted point.

Interpolation of plots can be made easier by the use of the `AXES` and `GRID` statements. The `AXES` statement allows you to draw an axis, with tick marks, along either or both the X-Y axes. The `GRID` statement can be used to section the plotting area into a grid coordinate system.

Enhancing the Plotting Area

There are many ways to enhance your graphics plot. One of the most common enhancements is the use of axes to help your interpolation of the plot.

AXES Statement

The `AXES` statement draws a pair of axes, with optional, equally spaced, tick marks.

Syntax:

```
AXES [X tick spacing, Y tick spacing[, X intersection, Y intersection[, X major
count, Y major count [, major-tick size ] ] ] ]
```

The X and Y tick-spacing parameter values are interpreted in current units. The sign of each parameter is ignored.

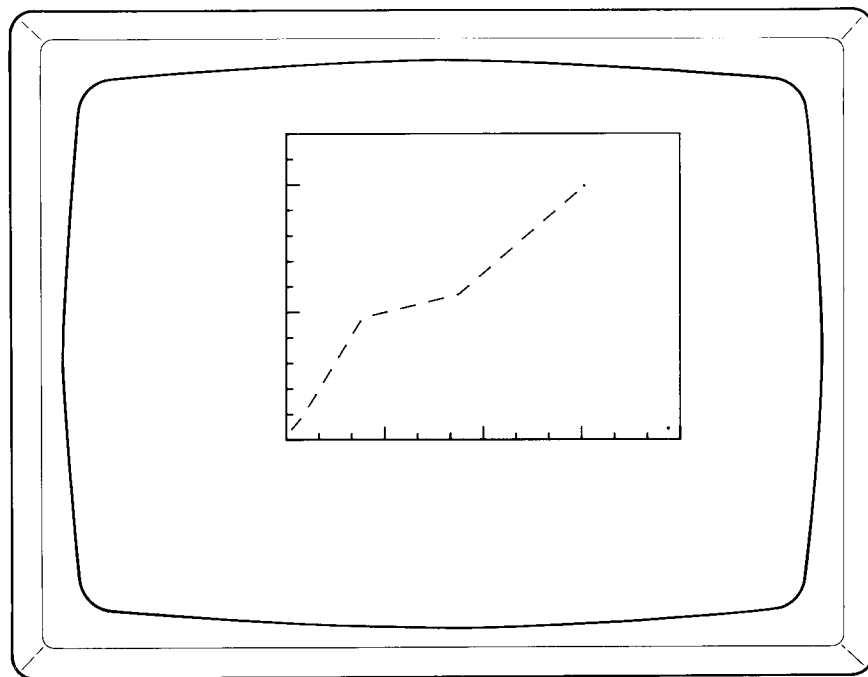
The `AXES` statement generates an X axis at the Y intersection and a Y axis at the X intersection, assuming that these values are within the current user units clipping limits. The axes extend across the soft clipping area. The default value for X and Y intersection is 0,0.

The X and Y major counts are unitless integer values which specify the number of minor tick intervals between major tick marks. The tick marks are positioned along each axis such that a major tick mark falls on the origin (whether visible or not). Tick marks may or may not coincide with the edge of the clipping boundary. The sign of each major count parameter is ignored and the default value for each is one (i.e. all major ticks).

The major-tick size specifies the length of the major tick marks (end to end) and like the minor tick marks, they are symmetric about the axes. The minor ticks are one half the length of the major ticks. The default size of the major tick marks is 2 GDU's. Both the major and minor tick marks are clipped at the soft clip limits.

Axes and tick marks are drawn using the current line type and pen number .

Example



```

1    REM *** This is the continuing example (CONT19)
10 Set up: !
20 DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30 PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40 GRAPHICS ! Enables the CRT
50 LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60 LOCATE 30,120,30,100 ! Defines the area to be scaled
70 CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80 SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 AXES 12,10,0,0,3,5,6 ! TICKS every 12 X units;every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKS every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280 READ Label$ ! READs data
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

GRID Statement

The `GRID` statement can be used as an alternative to the `AXES` statement when a full grid is desired.

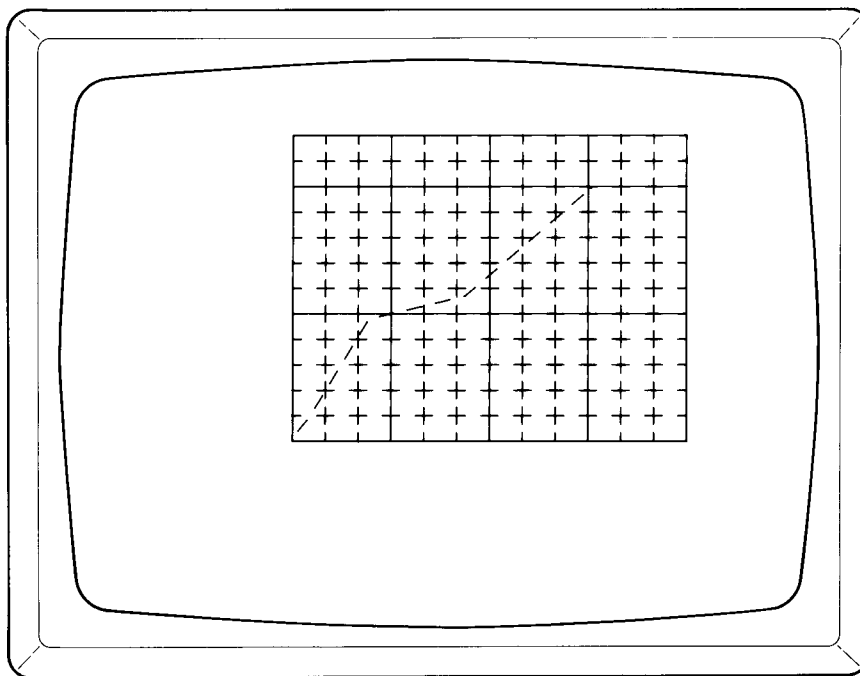
Syntax:

```
GRID [X tick spacing, Y tick spacing[, X intersection, Y intersection [, X major
count, Y major count [, major-tick size] ] ] ]
```

The parameters are the same as for the `AXES` statement. The `GRID` statement draws the major tick marks as lines extending across the soft clip area, located at the major count intervals. The minor tick mark intersections are drawn as cross ticks. The way to specify the minor tick size is to set the major tick mark size equal to twice whatever you want the minor tick mark size to be. Specifying a major tick mark size of 0 results in a dot at the minor tick mark intersection.

Also, like the `AXES` statement, the grid lines and tick marks are drawn using the current line type and pen number.

Example



```

1      REM *** This is the continuing example (CONT20)
10 Set up: !
20 DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30 PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40 GRAPHICS ! Enables the CRT
50 LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60 LOCATE 30,120,30,100 ! Defines the area to be scaled
70 CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80 SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 ! AXES 12,10,0,0,3,5,6 ! TICKs every 12 X units;every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKs every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
●115 GRID 12,10,72,50,3,5,8 ! GRID;TICKs every 12 X;every 10 Y units
116 ! Intersection at X=72, Y=50
117 ! Major TICKs every 3 X; every 5 Y
118 ! Minor TICK length 4 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
280 READ Label$ ! READs data
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

MOVE Statement

The `MOVE` statement lifts the pen and then moves to the absolute X, Y coordinate in current units. This statement provides an easy way of moving the pen without drawing a line and without regard to whether the pen is currently up or down. The pen's physical movement is restricted only by the hard clip limits.

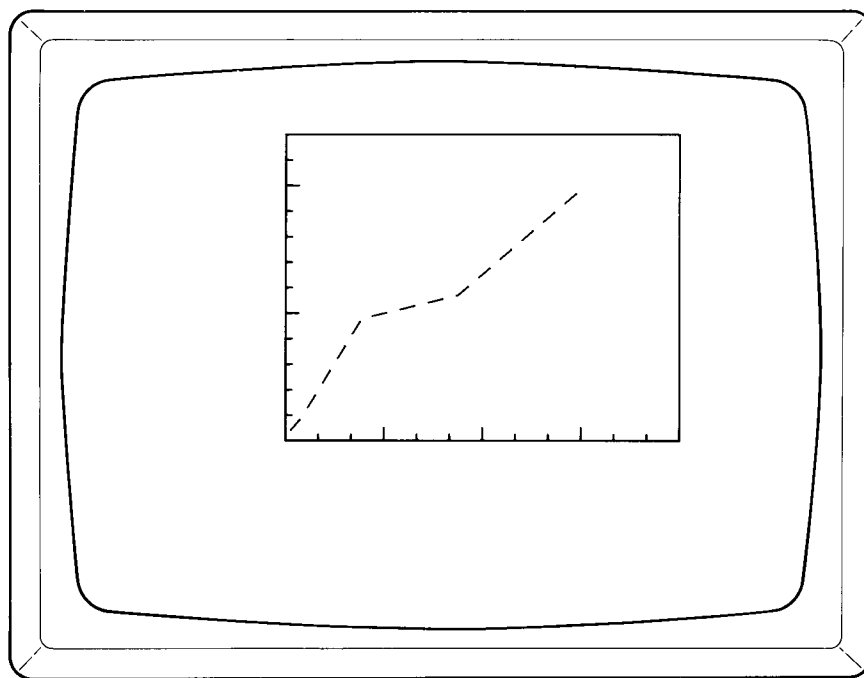
Syntax:

```
MOVE X parameter, Y parameter
```

The X and Y parameters are interpreted according to the current units.

`MOVE X, Y` is equivalent to `PLOT X, Y, -2`.

Example



DRAW Statement

The **DRAW** statement drops the pen and then moves it to the absolute X, Y coordinate leaving a line behind it. This is done in current units using the current pen number and line type. This statement provides an easy way of drawing a line from the current pen's location to a new location without regard as to whether the pen is currently up or down.

Syntax:

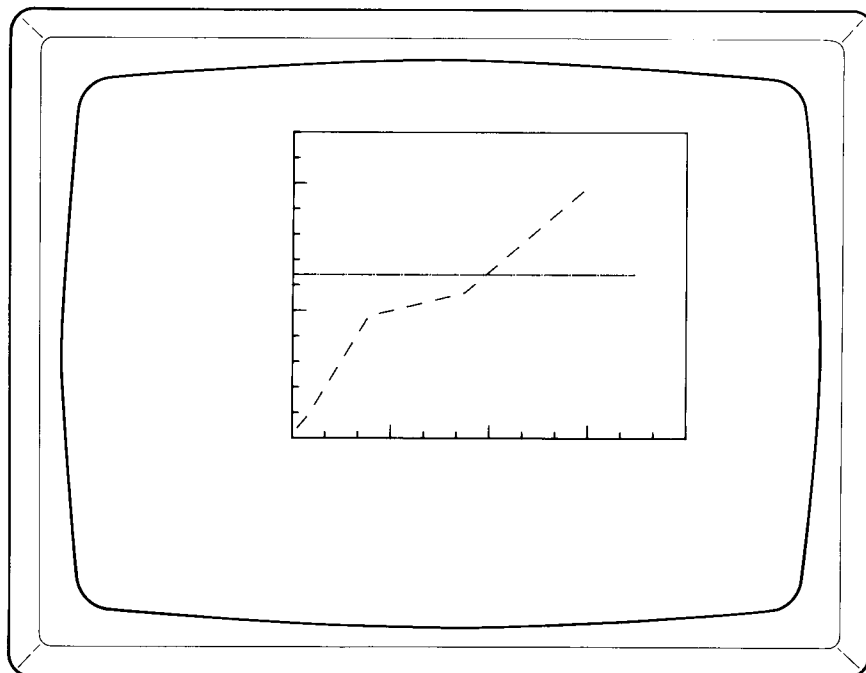
```
DRAW X parameter, Y parameter
```

The X and Y parameters are interpreted according to the current units.

`DRAW X, Y` is equivalent to `PLOT X, Y, -1`.

The line that is drawn is clipped as described for the **CLIP** statement. When drawing in GDUs only the hard clip limits are used. That is, drawn lines extend no further than the hard clip limit in any direction. When drawing in UDUs the lines extend no further than the soft clip boundaries in any direction.

Example



```

1      REM *** This is the continuing example (CONT22)
10 Set up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 AXES 12,10,0,0,3,5,6 ! TICKs every 12 X units; every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKs every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
190 MOVE 0,64 ! Positions PEN at X=0, Y=64
200 DRAW 125,64 ! Draws line to X=125, Y=64
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
270 MOVE X_label,-8 ! Positions PEN at X=X_label,Y=-8
280 READ Label$ ! READs data
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
330 MOVE -25,Y_label ! Positions PEN at X=-25, Y=Y_label
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
390 MOVE 5,90 ! Positions PEN at X=5, Y=90
430 MOVE 15,10 ! Positions PEN at X=15, Y=10
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

RPLOT Statement

The **RPLOT** statement provides capability for relative plotting with pen control. The origin is assumed to be the last absolute plotted point. The line is drawn using the current pen number and line type.

Syntax:

```
RPLOT X parameter, Y parameter [ , pen control]
```

The **RPLOT** statement interprets the X and Y parameters according to current units relative to a local origin. The local origin is the last absolute plotted point resulting from one of the following statements.

PLOT	DRAW	MOVE
FRAME	AXES	GRID
LABEL	IPLOT	

The local coordinate system can be rotated about its origin relative to the master coordinate system by means of the **PDIR** (plot direction) statement. That is, a figure can be drawn or repeated with the **RPLOT** statement at another angle by specifying the angle with the **PDIR** statement.

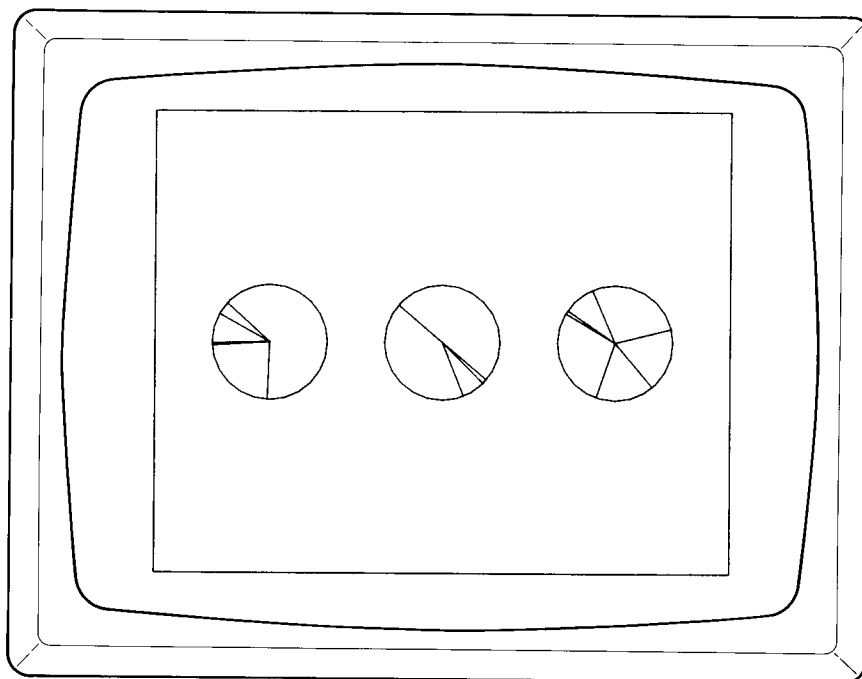
Pen control for the **RPLOT** statement is the same as the pen control for **PLOT** and **IPLOT**. The optional pen control parameter specifies the up or down pen movement and defaults to plus one (the pen moves and then drops). The pen control parameter is interpreted as follows.

Odd	= Drop pen (pen down or start line generation)
Even	= Lift pen (pen up or stop line generation)
Positive	= Pen change after motion
Negative	= Pen change before motion

The pen control parameter remains in effect until it is changed by a default condition, or by another pen control parameter, **PENUP** or **MOVE**.

When plotting in GDUs, only the hard clip limits are used. Drawn lines extend no further than the hard clip limit in any direction. When plotting in UDUs, the line extends no further than the soft clip boundaries in any direction.

Example



```

1  REM ** This is the RPLLOT example (RPLLOT)
10  PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT as the plotter
20  GRAPHICS                      ! Enables the CRT
30  DEG                          ! Sets the degrees mode
40  SHOW 0,10,-1,1               ! Sets the SHOW area
50  FRAME                        ! Draws a FRAME around the SOFT CLIP area
60  Pie_charts:!!
70  FOR Circle=2 TO 8 STEP 3      ! Selects one of three circles
80  MOVE Circle,0                ! MOVE to point to draw the circle
90  FOR Arc=1 TO 360              ! Specify point on circle to draw arc
100  PDIR Arc                    ! Specify Plot DIRECTION for "RPLLOT"
●110  RPLLOT 1,0                 ! Draws arc
120  NEXT Arc                    ! Repeats the loop
130  Random_lines:!!
140  FOR Section=1 TO RND*5+2     ! Specify a random number of sections
150  MOVE Circle,0                ! MOVE to the center of the circle
160  PDIR RND*360                 ! Specify a random line direction
●170  RPLLOT 1,0,-1              ! Draws a line section
180  NEXT Section                ! Repeats the loop
190  NEXT Circle                  ! Repeats the loop for the next circle
200  END

```

IPLOT Statement

The `IPLOT` statement provides capability for incremental plotting with pen control. The origin is assumed to be the last plotted point. That is, the origin is always the pen's position before this statement is executed.

Syntax:

```
IPLOT X increment, Y increment [ , pen control]
```

The `IPLOT` statement interprets the `X` and `Y` parameters according to the current units relative to a local origin. The local origin is that of the pen position before the `IPLOT` statement is executed (i.e. the current pen position).

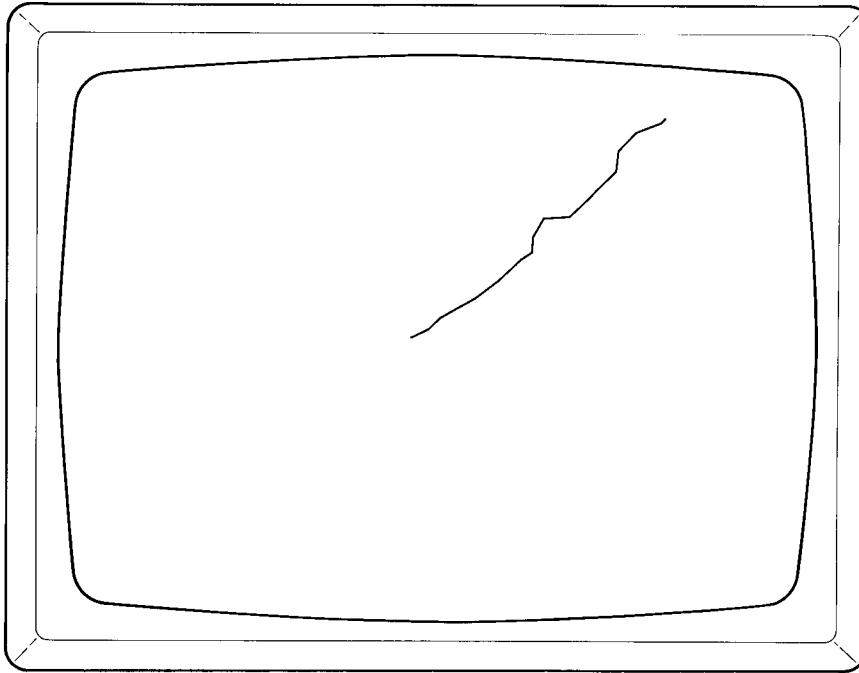
The local coordinate system can be rotated about its origin relative to the master coordinate system by means of the `PDIR` (plot direction) statement. That is, a figure can be drawn or repeated at another angle by specifying the angle with the `PDIR` statement.

The pen control for `IPLOT` is the same as the pen control for `PLOT` and `RPLLOT`. The optional pen control parameter specifies up or down pen movement, and defaults to plus one (the pen moves and then drops). The pen control parameter is interpreted as follows.

Odd	= Drop pen (pen down or start line generation)
Even	= Lift pen (pen up or stop line generation)
Positive	= Pen change after motion
Negative	= Pen change before motion

The pen control parameter remains in effect until it is changed by a default condition, or another pen control parameter, `PENUP` or `MOVE`. When plotting in GDUs only the hard clip limits are used. That is, drawn lines extend no further than the hard clip limit in any direction. When plotting in UDUs the line extends no further than the soft clip boundaries in any direction.

Example



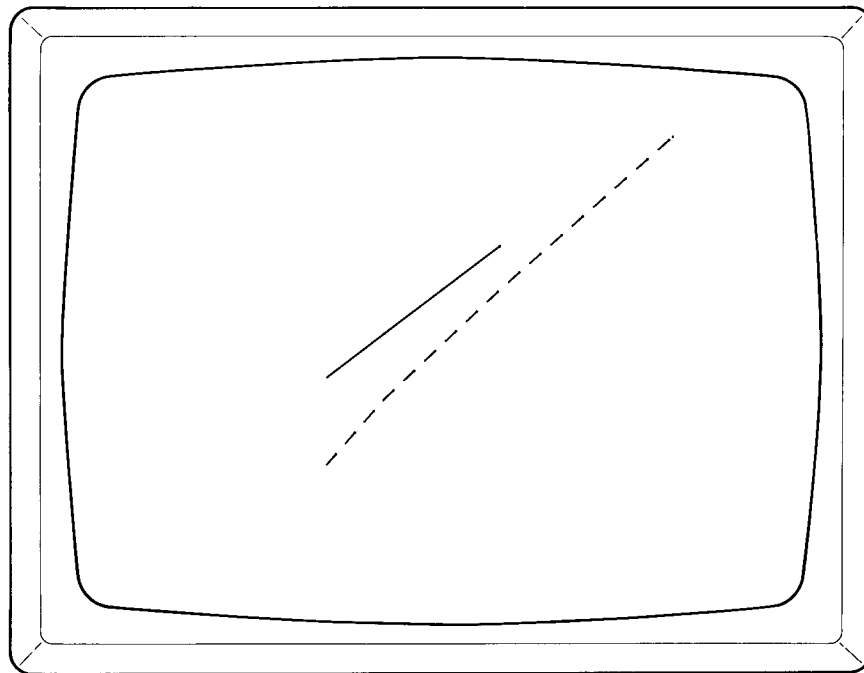
```

1  REM ** This is the IPLOT example (IPLOT)
10 PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20 GRAPHICS                ! Enables the CRT
30 MOVE 62.5,50            ! MOVES to the center of the CRT
40 FOR Loop=1 TO 25        ! Starts loop
● 50 IPLOT RND*6,RND*5,-1    ! IPLOT at a random coordinate
60 NEXT Loop               ! Repeats the loop
70 END

```

RPLOT vs IPLOT

RPLOT (relative plotting) is different from IPLOT (incremental plotting). In relative plotting, the lines are plotted with respect to a fixed point of origin. With incremental plotting, the lines are plotted with respect to the last plotted point. Running this example program should clarify the difference between the RPLOT and IPLOT.



```

1  REM ** This is the contrast between RPLOT and IPLOT (R_VS_I)
2  REM ** The first line plotted is Relatively PLOTted
3  REM ** The second line plotted is Incrementally PLOTted
10 PLOTTER IS "GRAPHICS"           ! Specifies CRT as plotter
20 GRAPHICS                        ! Enables the CRT
30 DEG                             ! Sets DEGREES MODE
40 SCALE 0,20,0,20                 ! SCALES the SOFT CLIP area to 20 x 20
50 DATA 0,1,2,3,4,5,6,7          ! Data for plots
60 MOVE 2,8                        ! Positions the PEN
70 FOR Loop=1 TO 4                 ! Initiates the loop
80 READ Plot_data1,Plot_data2      ! Reads the data
●90 RPLOT Plot_data1,Plot_data2     ! RPLOTS the data
100 NEXT Loop                      ! Repeats the loop
110 RESTORE 50                     ! Re-enables the DATA statement
120 WAIT 1000                      ! WAITS so you can see the first line
130 MOVE 2,4                       ! Positions the PEN
140 LINE TYPE 4                    ! Specifies a dashed LINE TYPE
150 FOR Loop=1 TO 4                ! Initiates the loop
160 READ Plot_data1,Plot_data2     ! Reads the data
●170 IPLOT Plot_data1,Plot_data2    ! IPLOTS the data
180 NEXT Loop                      ! Repeats the loop
190 END

```

Rotating the Plot

A plot of a figure that has been drawn by the `RPLLOT` or `IPLOT` statements can be turned or rotated. This feature can be used to give apparent motion to a figure, or to provide a different point of view to a plot.

`PDIR` is the statement that provides the angle of rotation for rotating the plot. `PDIR` is also helpful in positioning incrementally defined figures.

PDIR Statement

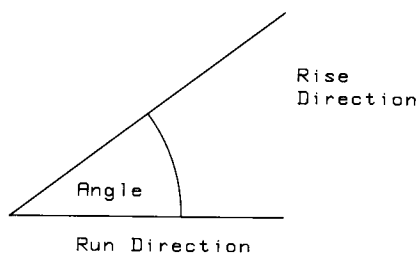
The `PDIR` statement sets the angle of rotation for relative (`RPLLOT`) and incremental (`IPLOT`) plotting.

Syntax:

```
PDIR angle or PDIR run, rise
```

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

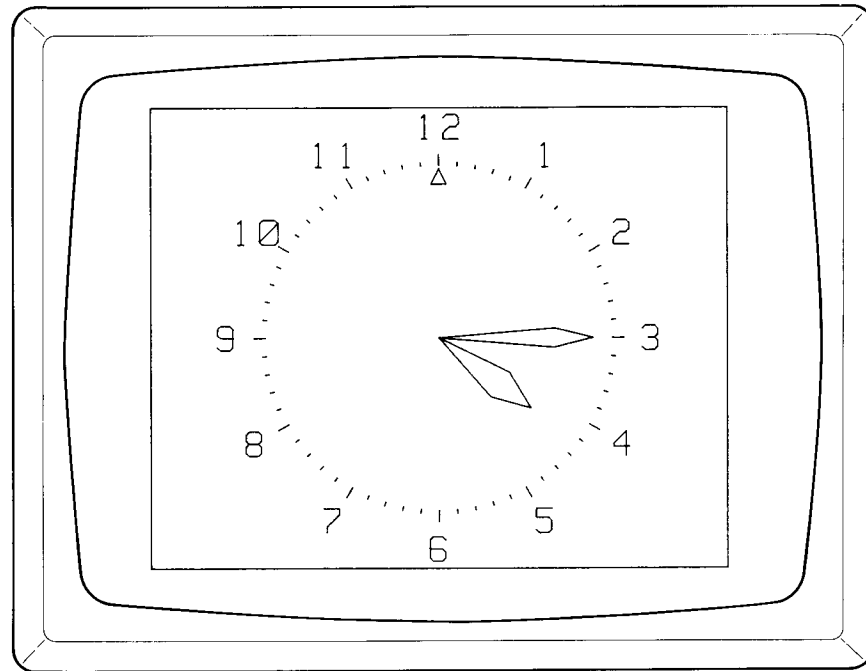
- As the counter-clockwise angle, in the current angular units set by the `DEG`, `RAD`, and the `GRAD` statements, between the new X-axis and the horizontal axis.
- The rotation parameter can also be specified as the run and rise units (in current units) of a vector drawn in the desired direction. The run parameter is the distance out along the X axis, while the rise parameter is the final height which the line must rise along the Y axis.



```
Angle equals
36.8698976457 Degrees
or
Run = 4 Rise = 3
```

On the CRT you can make a figure appear to move or rotate by erasing the old figure and redrawing it after executing a `PDIR` statement with a new angle. To erase the old figure on the CRT, either execute a `GOCLEAR` statement or redraw the old figure using a negative pen number.

Example



```

1    REM ** This is the PDIR example (PDIR)
10   ! This program shows how the PDIR statement can be used to rotate
20   ! a set of fixed points, in this case, the hands of a clock.
30   ! ****
40   PLOTTER IS 13,"GRAPHICS"           ! Specifies the CRT as the plotter
50   GRAPHICS                          ! Enables the CRT
60   SHOW -6,6,-6,6                    ! Sets X = Y
70   FRAME                             ! FRAMES the SOFT CLIP area
80   DEG                               ! Sets degrees mode
90   !
100  ! Draw the face of a clock
110  CSIZE 10                          ! Sets Character SIZE to 10 GDU's
120  LONG 5                            ! Specify centered numbers
130  FOR Minute=1 TO 60
140  MOVE 0,0                          ! FOR..NEXT to specify Minute mark
150  PDIR 360*(1-Minute/60)            ! MOVES to the center of the clock
160  IPOINT 0,4.5,-2                   ! Plot DIRection for Minute
170  IPOINT 0,.1,-1                    ! IPOINT to rim of the clock
180  IF Minute MOD 5 THEN GOTO 220     ! IPOINT the Minute mark
190  IPOINT 0,.2                        ! If not on 5 Minute mark, skip to NEXT
200  IPOINT 0,.7,-2                    ! IPOINT 5 Minute mark with longer length
210  LABEL USING "K";Minute/5          ! Moves to center of number
220  NEXT Minute                       ! LABEL with a number
230  !
240  ! ***** Enter the current time
250  EXIT GRAPHICS                      ! EXIT GRAPHICS for data input
260  INPUT "Enter the hour",Hour       ! Enter the Hour
270  Hour=INT(Hour) MOD 12              ! Adjust Hour to 12 Hour clock
280  INPUT "Enter the minutes",Minute  ! Enter the Minute
290  Minute=INT(Minute) MOD 60          ! Adjust the Minute to a 60 Minute Hour

```

```

300 Minute=Minute+60*Hour          ! Combine Minutes and Hours
310 GRAPHICS                      ! Returns to the GRAPHICS mode
320 !
330 ! ***** Draw, erase and rotate using PDIR and IPLOT statements
340 Loop: !
350 PEN -1                        ! Specify erase (PEN -1)
360 GOSUB Second_hand             ! Erase Second hand
370 Second=(Second+1) MOD 60      ! Increment seconds
380 PEN 1                         ! Specify to draw (PEN 1)
390 GOSUB Second_hand             ! Draws Second hand
400 IF NOT Second THEN PEN -1     ! Specify erase if Second = 0
410 GOSUB Minute_hand            ! Draw or erase Minute hand
420 GOSUB Hour_hand              ! Draw or erase Hour hand
430 IF NOT Second THEN Minute=Minute+1 ! If Second = 0, increment Minute
440 PEN 1                         ! Specify Draw (PEN 1)
450 GOSUB Minute_hand            ! Draws Minute hand
460 GOSUB Hour_hand              ! Draws Hour hand
470 WAIT 370                     ! Waits to adjust clock speed
480 GOTO Loop                    ! Repeats loop for next Second
490 END                          ! You must press STOP to STOP
500 ! ***** Routines for the Second, Minute and Hour hands
510 !
520 !
530 Second_hand: !
● 540 PDIR -Second*6
550 MOVE 0,0
560 IPLOT 0,4,4,-2
570 IPLOT -.2,-.4,-1
580 IPLOT .4,0
590 IPLOT -.2,.4
600 RETURN
610 !
620 !
630 Minute_hand: !
● 640 PDIR -Minute*6
650 MOVE 0,0
660 IPLOT -.25,3,-1
670 IPLOT .25,1
680 IPLOT .25,-1
690 IPLOT -.25,-3
700 RETURN
710 !
720 !
730 Hour_hand: !
● 740 PDIR -Minute*6/12
750 MOVE 0,0
760 IPLOT -.4,2,-1
770 IPLOT .4,1
780 IPLOT .4,-1
790 IPLOT -.4,-2
800 RETURN

```

Summary

The statements covered in this chapter include —

- MOVE
- DRAW
- RPLOT
- IPLOT
- PDIR
- AXES
- GRID

In this chapter you learned how to move the pen, draw a line to a specific coordinate, how to plot relative to an origin, how to plot an incremental figure, how to plot at an angular tangent, and how to mark axes and grids onto your plotting area.

The next chapter covers labeling and lettering of your graphic's plot.

Labeling Statements

- page 87 • **LABEL** (labels characters on plotting area)
- page 87 • **LABEL USING** (labels formatted characters on plotting area)
- page 90 • **LETTER** (labels characters under keyboard control on plotting area)
- page 94 • **CSIZE** (specifies character size, aspect ratio and slant)
- page 99 • **LDIR** (specifies angle at which characters are labeled)
- page 102 • **LOG** (specifies relationship of labeled characters to an origin point)

Terms

- **Hard Clip** - The physical limits of the plotting device, beyond which no line can be drawn.
- **Angle** - The direction at which a line or label is drawn, represented in degrees counter-clockwise from the horizontal.
- **Slant** - The angle at which a character is drawn, represented in clockwise degrees from the vertical.
- **Origin** - The coordinate point at which a plotting operation begins.

Implementing the LDIR Angle



Angle equals
36.8698976457 Degrees
or
Run = 4 Rise = 3

Label Origins



Chapter 5

Overview

System 45B graphics allows you to label or letter your plots or graphics outputs. Labeling can be done with the `LABEL` statement, while labeling using a format can be done with the `LABEL USING` statement. You can also “type” directly onto the plot with the `LETTER` statement.

If you wish to change the size, aspect ratio or **slant** of the characters which are being labeled, you can use the `CSIZE` statement.

Positioning of the labeling string can be changed by the `LDIR` statement, so you can label at any **angle**.

Justification (alignment) of text can be accomplished by using the `LORG` statement. In case you are not familiar with justification of text, the following example is provided for you.

This text is
left-justified.

And this text is
centered.

While this text is
right-justified.

The default labeling conditions are characters with a size of .3 cm, and an aspect ratio of 9/15. The characters are slanted at 0 from the vertical and the text is labeled from left to right on a straight line (this is similar to the text in this book). All of the labels are left justified (a label origin value of one).

LABEL and LABEL USING Statements

The LABEL and LABEL USING statements are very similar to the PRINT and PRINT USING statements, except that the output is directed to the current plotter as a label. The available labeling area is defined by the current **hard clip** area.

Syntax:

```
LABEL list
or
LABEL USING image specifier ; list
```

LABEL and LABEL USING follow the same rules as PRINT and PRINT USING with one exception. The exception is that an ASCII Ex (CHR\$(3)) terminates the labeled output at that point in the list.

For example, the program

```
1  REM ** This is the little example (LITTLE)
10 PLOTTER IS "GRAPHICS"      ! Specifies the CRT as the plotter
20 GRAPHICS                  ! Enables the CRT
30 MOVE 0,50                 ! Positions the PEN
40 LABEL "THIS IS A"&CHR$(3)&"TEST" ! LABELs text
50 END
```

results in just

```
THIS IS A
```

being labeled.

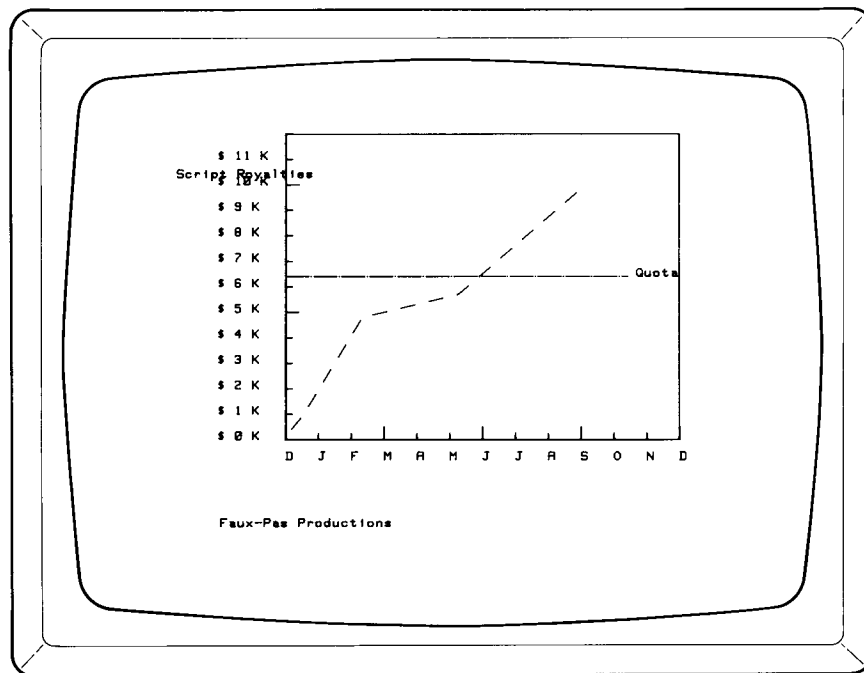
The position and rotation of a label is controlled by the pen positions and the **LOG** (label **origin**) and **LDIR** (label direction) statements. The size and aspect ratio as well as the slant of the characters are controlled by the **CSIZE** (character size) statement.

Remember, labeling can be done anywhere within the hard clip limits. The soft clip limits do not affect the labels.

Note

The LABEL USING statement requires the use of image specifiers with variables and text. Failure to use either numeric image specifiers with numeric variables or string image specifiers with string variables results in errors 100 and 101 being generated under program execution.

Example



```

1      REM *** This is the continuing example (CONT23)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 AXES 12,10,0,0,3,5,6 ! TICKs every 12 X units;every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKs every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
190 MOVE 0,64 ! Positions PEN at X=0, Y=64
200 DRAW 125,64 ! Draws line to X=125, Y=64
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
●230 LABEL " Quota" ! LABELs text
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
270 MOVE X_label,-8 ! Positions PEN at X=X_label,Y=-8
280 READ Label$ ! READs data
●290 LABEL Label$ ! LABELs text
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
330 MOVE -25,Y_label ! Positions PEN at X=-25, Y=Y_label
●340 LABEL "$";Y_label/10;"K" ! LABELs text
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
390 MOVE 5,90 ! Positions PEN at X=5, Y=90
●420 LABEL "Script Royalties" ! LABELs text
430 MOVE 15,10 ! Positions PEN at X=15, Y=10
●470 LABEL USING "K";"Faux-Pas Productions" ! LABELs text
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

LETTER Statement

The `LETTER` statement allows you to draw all keyboard alphanumerics and symbols by typing them in on the keyboard. The size, slant and direction of the characters are controlled the same as with the `LABEL` statement (by the `CSIZE` and `LDIR` statements).

Syntax:

```
LETTER
```

Executing the `LETTER` statement directly from the keyboard or under program control automatically places the System 45B into the `GRAPHICS` mode.

The normal lower-left corner of the character is placed at the cursor position. You can preset the position of the cursor with the `POINTER` statement before executing the `LETTER` statement.

On the CRT, the cursor for the `LETTER` statement is a small flashing line. The actual cursor position is the center of the flashing line. This is where the pen draws the character.

To move the CRT cursor, press the shifted or unshifted arrow keys     as needed. The shifted keys move the cursor one dot row each time the key is pressed; the unshifted keys move the cursor one character cell (refer to the `CSIZE` statement) each time the key is pressed. Pressing  moves the cursor to the lower-left corner of the CRT hard clip limits. Remember, pressing the key firmly repeats the key action.

The `LETTER` statement is much slower than typing, because the characters are drawn on the CRT. You should wait for the character to be drawn completely before pressing the next character key.

Erasing Characters

GCLEAR erases **everything** which you have on the CRT in the graphics mode. Erasing a single letter consists of exiting the graphics mode, changing the pen to a negative number, re-executing the **LETTER** statement, positioning the cursor under the character which you want to erase, typing the same character, exiting the graphics mode, changing the pen to a positive number, re-executing the **LETTER** statement, positioning the cursor and pressing the proper key for the character.

To exit the **LETTER** mode, press either   any special function key (such as ) or .

The **LETTER** statement leaves both the pen position and the cursor position at the same point whenever you exit graphics.

The Graphicswriter

The **LETTER** statement allows the System 45B to use its graphics labeling similar to a typewriter. This use can be enhanced by using the following available keys.

	moves the cursor up one line each time it is pressed.
	moves the cursor down one line each time it is pressed.
	moves the cursor left one character space each time it is pressed.
	moves the cursor right one character space each time it is pressed.

Pressing the  with any of the “arrow” keys moves the cursor one resolution unit in that direction. (On the CRT, this is approximately .033 mm. On the 9872A, this is equal to .025mm.)

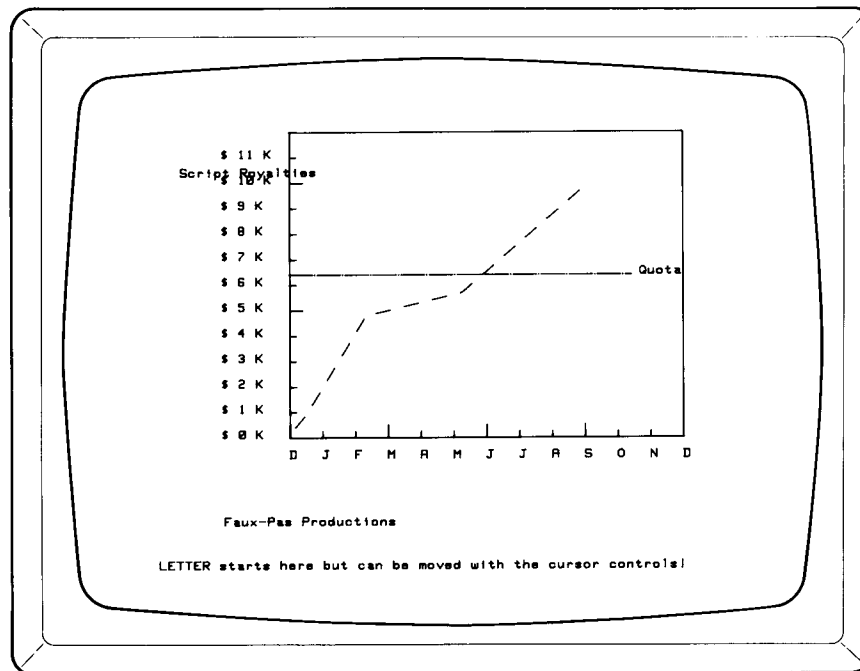
These directional movements are valid even when the label line is at an angle due to a **LDIF** statement.

Pressing **CONTROL** and one of the typing keys generates an ASCII control character. The following control key combinations may be useful to you as typing aids.

CONTROL M	Carriage Return. The cursor moves to the beginning of the line or to the point last moved to with an arrow key.
CONTROL J	Linefeed. The cursor moves down one line.
CONTROL K	Vertical Tab. The cursor moves up one line.
CONTROL H	Backspace. The cursor moves backwards one character space.

The **LETTER** statement always uses **LOG** 1. The cursor is always positioned at the lower-left corner of the character cell.

Example



```

1    REM *** This is the continuing example (CONT24)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 AXES 12,10,0,0,3,5,6 ! TICKs every 12 X units;every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKs every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
190 MOVE 0,64 ! Positions PEN at X=0, Y=64
200 DRAW 125,64 ! Draws line to X=125, Y=64
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
230 LABEL " Quota" ! LABELs text
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
270 MOVE X_label,-8 ! Positions PEN at X=X_label,Y=-8
280 READ Label$ ! READs data
290 LABEL Label$ ! LABELs text
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
330 MOVE -25,Y_label ! Positions PEN at X=-25, Y=Y_label
340 LABEL "#";Y_label/10;"K" ! LABELs text
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
390 MOVE 5,90 ! Positions PEN at X=5, Y=90
420 LABEL "Script Royalties" ! LABELs text
430 MOVE 15,10 ! Positions PEN at X=15, Y=10
470 LABEL USING "K";"Faux-Pas Productions" ! LABELs text
●475 LETTER ! Allows you to annotate the plot
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

CSIZE Statement

The `CSIZE` (character size) statement specifies the size and aspect ratio (character cell) and the slant of the alphanumerics or symbols to be drawn for labels.

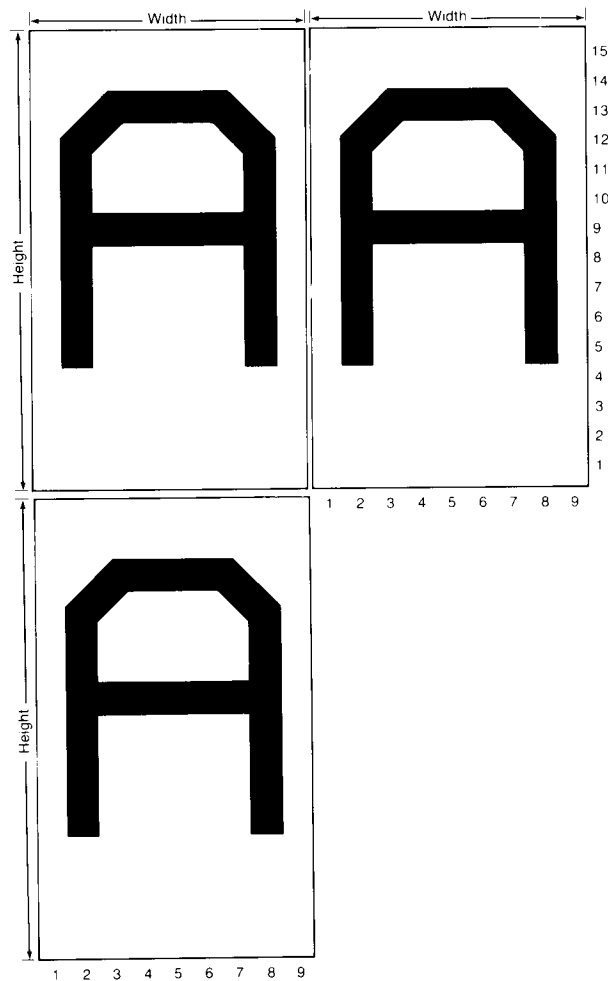
Syntax:

```
CSIZE height[, aspect ratio [, slant]]
```

The height parameter is in GDU's and must be given. The default value is approximately 3.3 GDU's (15/4.54) which is set by the `PLOTTER IS` or the `LIMIT` statement.

The aspect ratio parameter specifies the character cell aspect ratio, defined as the width/height. The default value is 9/15. The width of a character cell is defined as the height times the aspect ratio.

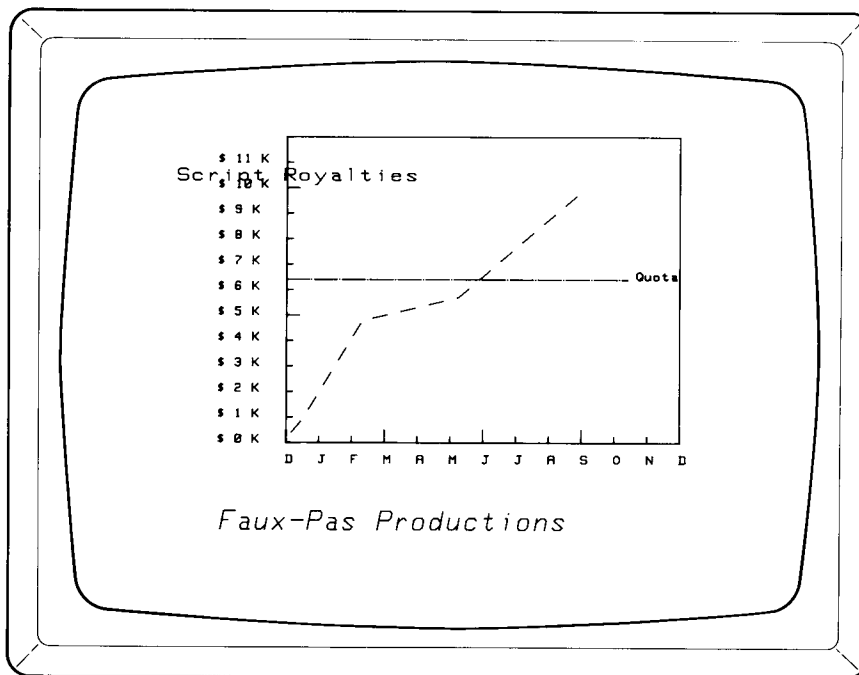
The character cells are adjacent, both horizontally and vertically, as shown here



This results in the spacing between adjacent characters being dependent upon each character's positioning in its cell and the `CSIZE` statement which is currently in effect.

Of course, a line is equal to the height of a character cell, while a space between characters is equal to the width of a character cell.

Examples



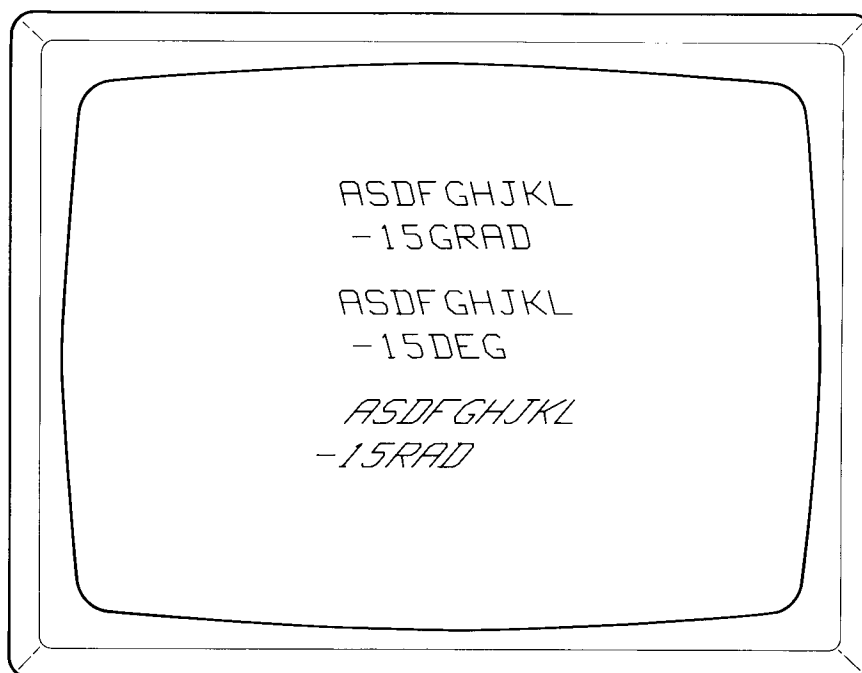
```

1      REM *** This is the continuing example (CONT25)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 AXES 12,10,0,0,3,5,6 ! TICKs every 12 X units;every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKs every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
190 MOVE 0,64 ! Positions PEN at X=0, Y=64
200 DRAW 125,64 ! Draws line to X=125, Y=64
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
230 LABEL " Quota" ! LABELs text
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
270 MOVE X_label,-8 ! Positions PEN at X=X_label,Y=-8
280 READ Label$ ! READs data
290 LABEL Label$ ! LABELs text
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
330 MOVE -25,Y_label ! Positions PEN at X=-25, Y=Y_label
340 LABEL "$";Y_label/10;"K" ! LABELs text
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
390 MOVE 5,90 ! Positions PEN at X=5, Y=90
● 400 CSIZE 5,.7 ! Sets Character SIZE and aspect ratio
420 LABEL "Script Royalties" ! LABELs text
430 MOVE 15,10 ! Positions PEN at X=15, Y=10
● 450 CSIZE 8,.5,13 ! Sets the SIZE, aspect ratio, and slant
470 LABEL USING "K";"Faux-Pas Productions" ! LABELs text
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

The character slant is an angular measurement, expressed in the current machine mode (DEG,RAD or GRAD) in effect at the time the CSIZE statement is executed, which causes the characters to be labeled at the specified angle.

This program demonstrates the effects that the various modes have on the same angular setting.

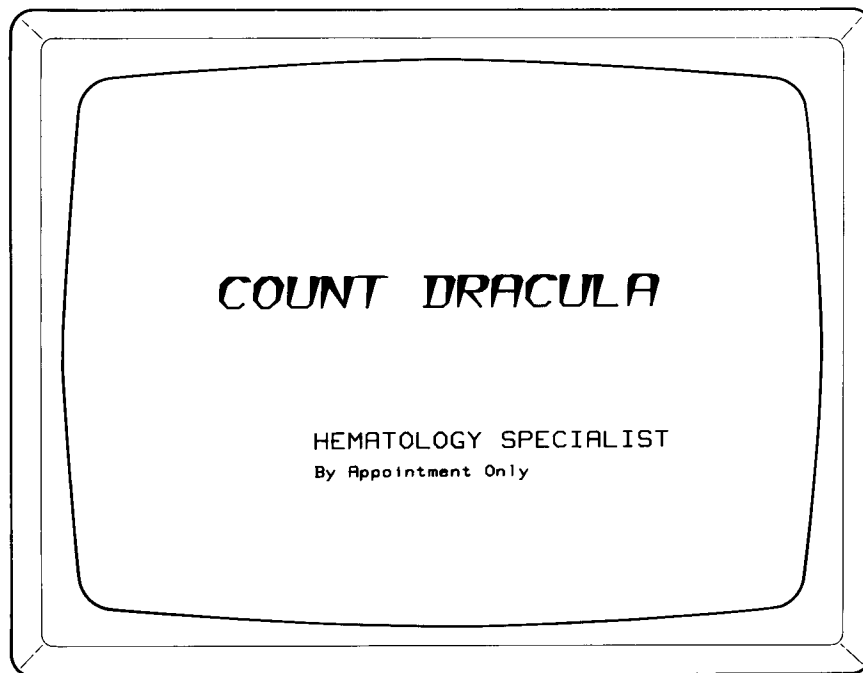


```

10  REM ** This is the Angle example (ANGLE)
20  PLOTTER IS "GRAPHICS"           ! Specifies the CRT as the plotter
30  GRAPHICS                        ! Enables the CRT
40  Yposition=15                    ! Sets Yposition
50  DATA RAD,DEG,GRAD              ! DATA for Mode$
60  RAD                             ! Sets RADians mode
70  GOSUB Sample                    ! Performs LABELing
80  DEG                             ! Sets DEGrees mode
90  GOSUB Sample                    ! Performs LABELing
100 GRAD                            ! Sets GRADs mode
110 GOSUB Sample                    ! Performs LABELing
120 END
130 Sample: !
●140 CSIZE 10,.6,-15                ! Sets Angle to -15
150 MOVE 40,Yposition               ! Positions the PEN
160 READ Mode$                      ! Keeps track of current mode
170 LABEL "ASDFGHJKL"               ! Characters to be LABEled
180 LABEL "-15";Mode$               ! Angular setting and current mode
190 Yposition=Yposition+25          ! Increments Yposition
200 RETURN

```

Another interesting effect which can be obtained with the angle parameter is the re-drawing of characters while incrementing their angle by a degree at a time. This is shown in the following program.



```

1    REM ** This is the character angle example (DRACUL)
10   PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT as the plotter
20   GRAPHICS                     ! Enables the CRT
30   DEG                          ! Sets the degrees mode
40   FOR Angle=1 TO 17             ! LABELs 17 times at 1 degree increments
●50   CSIZE 6,.6,Angle             ! Sets Character SIZE, ratio and angle
60   MOVE 40,50                   ! Positions the PEN
70   LABEL "COUNT DRACULA"       ! LABELs text
80   NEXT Angle                   ! Repeats the loop
●82   CSIZE 3,.6,0                ! Sets Character SIZE, ratio and angle
90   MOVE 50,35                   ! Positions the PEN
100  LABEL "HEMATOLOGY SPECIALIST" ! LABELs text
●110 CSIZE 2                      ! Sets Character SIZE
120  LABEL "By Appointment Only"  ! LABELs text
130  DUMP GRAPHICS                ! Thermal Printer output
140  END

```

LDIR Statement

The `LDIR` (label direction) statement specifies the angle at which the `LETTER` and the `LABEL` statements draw characters. The label direction is distinct from the character slant.

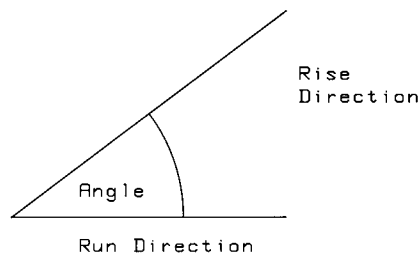
Syntax:

```
LDIR angle
or
LDIR run, rise
```

The `LDIR` statement allows you to rotate a label about its origin (refer to `LOG`) by specifying the required angle.

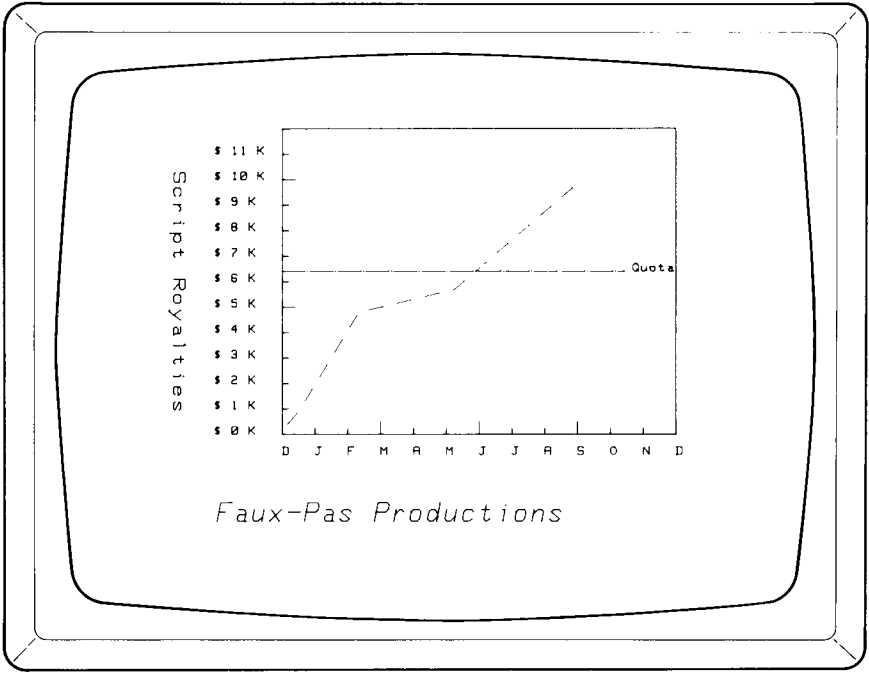
The angle rotation can be specified in either of two ways:

- As the counter-clockwise angle, in the current angular units set by the `DEG`, `RAD`, and the `GRAD` statements, between the new X axis and the horizontal axis.
- The rotation parameter can also be specified as the run and rise units (in current units) of a vector drawn in the desired direction. The run parameter is the distance out along the X axis, while the rise parameter is the final height which the line must rise along the Y axis.



```
Angle equals
36.8698976457 Degrees
or
Run = 4 Rise = 3
```

Example



```

1    REM *** This is the continuing example (CONT26)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 AXES 12,10,0,0,3,5,6 ! TICKs every 12 X units; every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKs every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
190 MOVE 0,64 ! Positions PEN at X=0, Y=64
200 DRAW 125,64 ! Draws line to X=125, Y=64
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
230 LABEL "Quota" ! LABELs text
240 Label_axes: !
250 DATA I,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
270 MOVE X_label,-8 ! Positions PEN at X=X_label, Y=-8
280 READ Label$ ! READs data
290 LABEL Label$ ! LABELs text
300 NEXT X_label ! Repeats loop
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
330 MOVE -25,Y_label ! Positions PEN at X=-25, Y=Y_label
340 LABEL "#";Y_label/10;"K" ! LABELs text
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
390 MOVE 5,90 ! Positions PEN at X=5, Y=90
400 CSIZE 5,.7 ! Sets Character SIZE and aspect ratio
●410 LDIR -90 ! Labels down
420 LABEL "Script Royalties" ! LABELs text
430 MOVE 15,10 ! Positions PEN at X=15, Y=10
●440 LDIR 0 ! Labels left to right
450 CSIZE 8,.5,13 ! Sets the SIZE, aspect ratio, and slant
470 LABEL USING "K";"Faux-Pas Productions" ! LABELs text
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

LORG Statement

The LORG (label origin) statement sets the label position which determines where subsequent labels are placed relative to the current pen position.

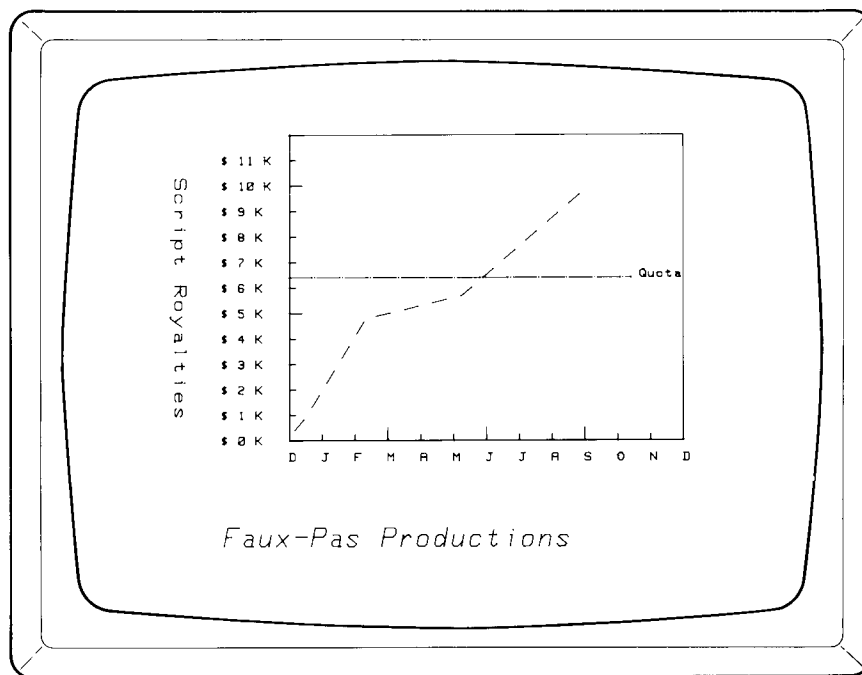
Syntax:

LORG origin position

The origin position can have a value between one and nine. If LORG is not executed, the default origin position is one.

LORG 1, 2 and 3 causes left justified labels. LORG 4, 5 and 6 causes centered labels. LORG 7, 8 and 9 causes right justified labels.

Example



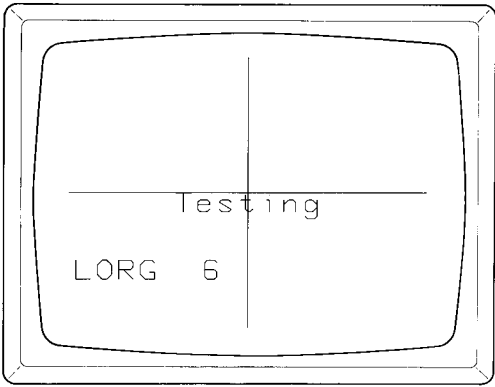
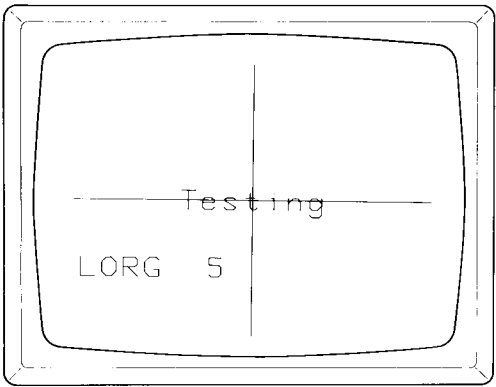
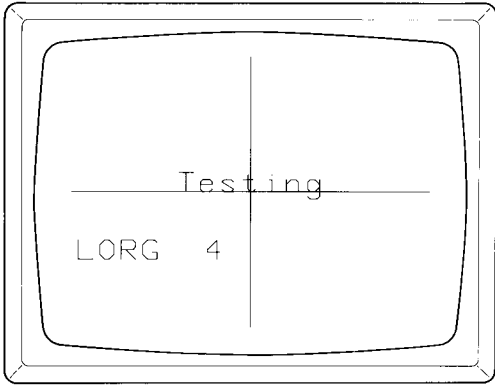
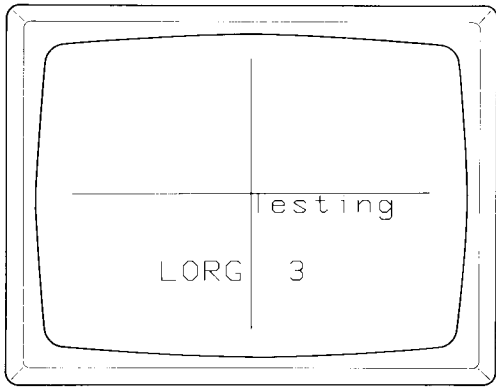
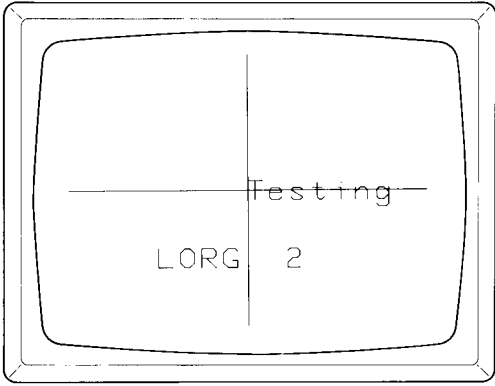
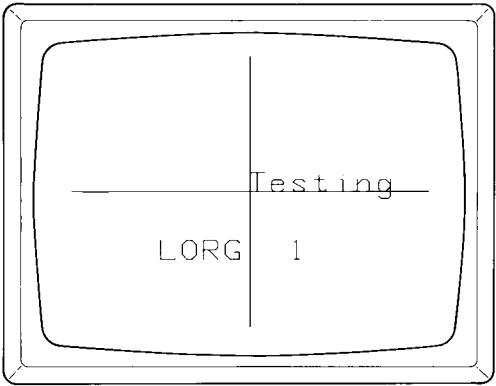
```

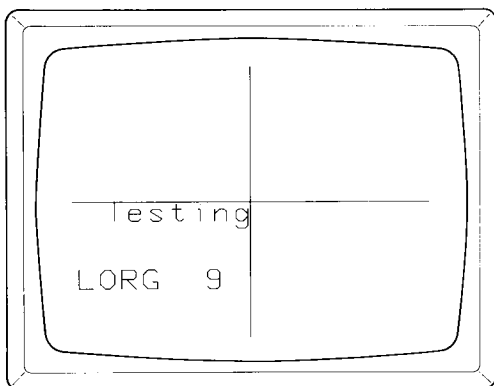
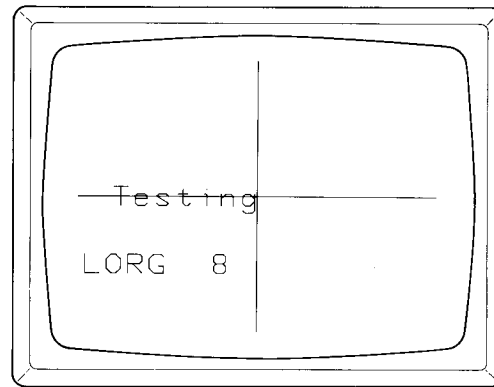
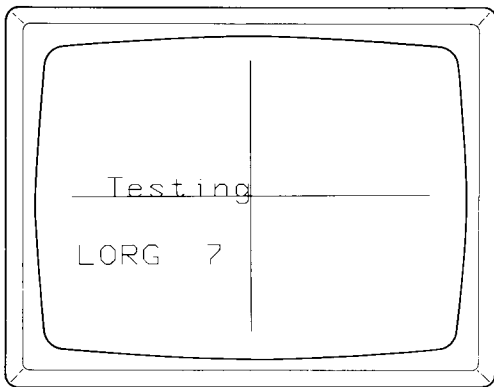
1    REM *** This is the continuing example (CONT27)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 AXES 12,10,0,0,3,5,6 ! TICKS every 12 X units;every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKS every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
190 MOVE 0,64 ! Positions PEN at X=0, Y=64
200 DRAW 125,64 ! Draws line to X=125, Y=64
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
230 LABEL " Quota" ! LABELs text
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
270 MOVE X_label,-8 ! Positions PEN at X=X_label,Y=-8
280 READ Label$ ! READs data
290 LABEL Label$ ! LABELs text
300 NEXT X_label ! Repeats loop
● 310 LORG 2 ! Left justify & center labels
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
330 MOVE -25,Y_label ! Positions PEN at X=-25, Y=Y_label
340 LABEL "#";Y_label/10;"K" ! LABELs text
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
390 MOVE 5,90 ! Positions PEN at X=5, Y=90
400 CSIZE 5,.7 ! Sets Character SIZE and aspect ratio
410 LDIR -90 ! Labels down
420 LABEL "Script Royalties" ! LABELs text
430 MOVE 15,10 ! Positions PEN at X=15, Y=10
440 LDIR 0 ! Labels left to right
450 CSIZE 8,.5,13 ! Sets the SIZE, aspect ratio, and slant
● 460 LORG 3 ! Positions the label
470 LABEL USING "K";"Faux-Pas Productions" ! LABELs text
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

The following example demonstrates the various ways that you can position a label by using the LORG statement. All LORG positions are calculated based upon an upper-case character.

If you would like a hard copy printout and your System 45B has the built-in thermal printer option, change line 110 to DUMP GRAPHICS.





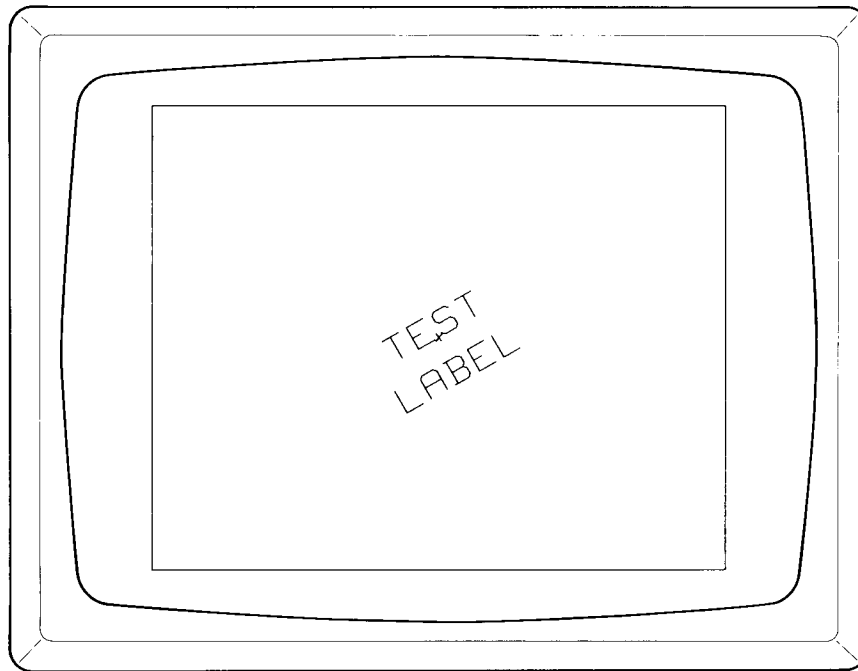
```

1  REM ** This program demonstrates the LORG statement (LORGEX)
10 PLOTTER IS "GRAPHICS"      ! Specifies the CRT as the plotter
20 GRAPHICS                   ! Enables the CRT
30 SCALE -10,10,-10,10        ! SCALES the LOCATE area to 20x20
40 FOR Long_value=1 TO 9      ! Initializes a loop for LORG values
50 AXES                       ! Draws AXES at 0,0 intersection
60 LORG Long_value            ! Sets LORG equal to value of loop
70 MOVE 0,0                   ! MOVES PEN to AXES intersection
80 LABEL "Testing"            ! LABELs text with LORG currently in effect
90 MOVE -5,-5                 ! MOVES into lower left quadrant
100 LABEL "LORG ";Long_value  ! LABELs LORG which is currently in effect
110 WAIT 1500                 ! Display GRAPHICS for 1.5 seconds
120 GCLEAR                    ! Clears Graphics memory and the CRT
130 NEXT Long_value           ! Repeats the loop for NEXT Long_value
140 END

```

If the label is rotated by the label direction statement (see LDIR), the relationship stays fixed relative to the label, not the plotting area. This is shown in the following example.

Example



```

1      REM ** This is another LORG example (LORG3)
10     ! This program demonstrates how the LORG is positioned
20     ! relative to the LDIR statement
30     PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT as the plotter
40     GRAPHICS                     ! Enables the CRT
50     SCALE -1,1,-1,1              ! SCALES the LOCATE area to 2x2
60     DEG                          ! Sets degrees mode
70     FRAME                        ! FRAMES the SOFT CLIP area
80     !
90     ! ***** MOVE to point,LABEL an "X" to indicate the PEN position
●100    LORG 5                      ! Specifies LORG to center LABELs
110    CSIZE 3                      ! Sets Character SIZE,ratio and slant
120    MOVE 0,0                    ! MOVES to center of plotting area
130    LABEL USING "K";"X"          ! LABEL the "X"
140    !
150    ! ***** MOVE to point, and LABEL
160    MOVE 0,0                    ! MOVES to center of plotting area
170    LDIR 30                     ! LABEL at 30 degree angle
●180    LORG 4                      ! Specify LABEL origin at bottom centered
190    CSIZE 10                    ! Sets Character SIZE,ratio and slant
200    LABEL USING "K";"TEST"       ! LABELs "TEST"
210    LABEL USING "K";"LABEL"      ! LABELs "LABEL"
220    END

```

Summary

The following statements were explained in this chapter

- LABEL
- LABEL USING
- LETTER
- CSIZE
- LDIR
- LORG

In this chapter you learned how to label a plot, change the size, aspect ratio, and slant of each character, change the angle at which a line is labeled, and how to center and align (justify) labels.

The following chapter covers the statements associated with digitizing and cursor control.

Digitizing and Cursor Control

- page 111 • **POINTER** (positions the cursor and specifies the type of cursor)
 page 114 • **CURSOR** (returns current cursor coordinates and pen up/down information)
 page 116 • **DIGITIZE** (allows positioning of cursor before returning current cursor coordinates and pen up/down information)
 page 120 • **WHERE** (returns several pen coordinates and pen up/down information)

Terms

- **Pen** - The device which is used to draw or plot lines, and to label characters.
 • **Cursor** - The device which is used to obtain digitizing information.
 • **Digitizing** - The process of obtaining an X,Y coordinate pair based on the location of the cursor.

Digitizing Information

Device	Cursor Moved by	Coordinates Entered by
CRT	 and  and 	  or any 
HP 9872A		
Incremental Plotter	 and  and 	  or any 

Chapter 6

Pen vs Cursor Concept

All plotting operations are done with a **pen**. Conversely, all digitizing operations are done with a **cursor**. On a lot of plotting/digitizing devices (such as the 9872A Plotter), the pen and the cursor can be the same physical component on the device. However, the System 45B considers the pen and the cursor as two logically separate devices. Consequently your System 45B is able to keep track of one set of coordinates for the location of the pen, and another set of coordinates for the cursor's location. Moving the pen does not affect the position that the cursor occupies.

The `POINTER` statement is used to position a cursor on your graphic display at a specific X,Y coordinate. The `POINTER` statement is also used to select which type of two available cursors (large cross-hair, or small blinking cross-hair) is currently being used.

The `CURSOR` statement allows your System 45B to obtain the present X,Y coordinate of the cursor from the specified plotting device, as well as whether the pen is raised or lowered on the plotting surface.

The `DIGITIZE` statement pauses the program execution until you have positioned the cursor over a specific point on the plot and pressed  to enter the X,Y coordinate data. The pen up/down information is also obtained by the `DIGITIZE` statement.

The `WHERE` statement is similar to the `CURSOR` statement, except that the X,Y coordinate data and the up/down information refers to the pen which is currently in use.

Remember, the pen position is the location at which plotting and labeling operations are done; the cursor position is the location at which digitizing operations are done.

Only six statements allow the cursor to be used. The six statements are –

- | | |
|--|-------------------------|
| ● <code>LIMIT</code> (without parameters) | ● <code>POINTER</code> |
| ● <code>LOCATE</code> (without parameters) | ● <code>CURSOR</code> |
| ● <code>CLIP</code> (without parameters) | ● <code>DIGITIZE</code> |

Of the four statements explained in this chapter, three of them (`POINTER`, `CURSOR` and `DIGITIZE`) refer to the cursor position. Only the `WHERE` statement refers to the pen position.

POINTER Statement

The `POINTER` statement moves the cursor to a specified absolute position. This is similar to a `MOVE` statement for the plotter's pen.

Syntax:

```
POINTER X parameter, Y parameter[, cursor type]
```

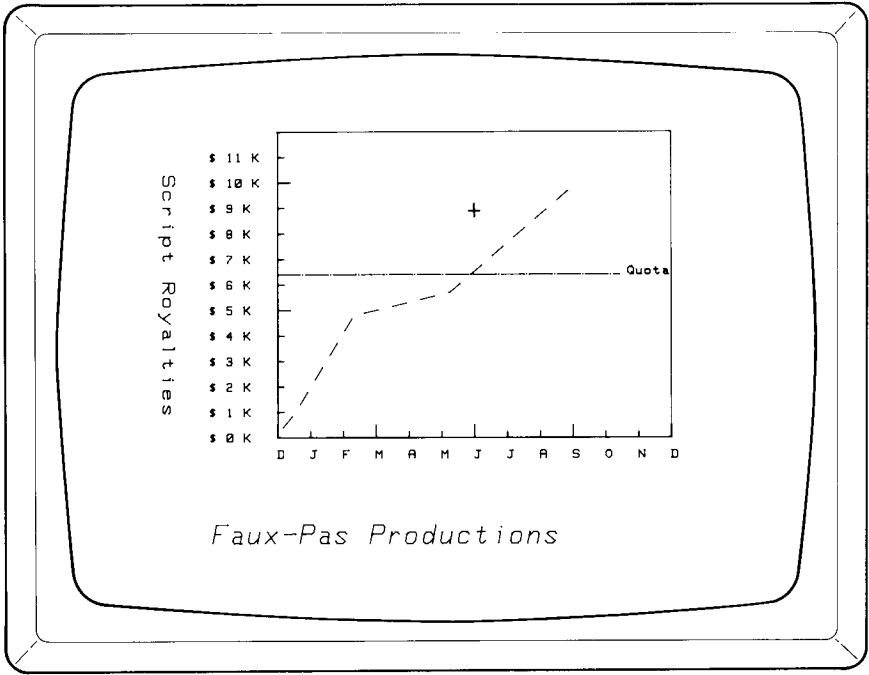
The X and Y parameters are specified in current units. If the `POINTER` is positioned outside the hard clip limit, the cursor is turned off.

The cursor type parameter specifies one of two types of cursors available on the CRT.

- An odd number produces a full screen crossed-line cursor.
- An even number produces a small blinking cross cursor.

Once a cursor type is selected, it is the type used by all other statements that use the cursor (e.g. `DIGITIZE` and `LIMIT` without parameters) until changed again. The default cursor type for the CRT is the full screen crossed-lines. For external plotting devices, the cursor type parameter is ignored.

Example



```

1    REM *** This is the continuing example (CONT28)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 AXES 12,10,0,0,3,5,6 ! TICKS every 12 X units; every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKS every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
190 MOVE 0,64 ! Positions PEN at X=0, Y=64
200 DRAW 125,64 ! Draws line to X=125, Y=64
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
230 LABEL " Quota" ! LABELs text
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
270 MOVE X_label,-8 ! Positions PEN at X=X_label,Y=-8
280 READ Label$ ! READs data
290 LABEL Label$ ! LABELs text
300 NEXT X_label ! Repeats loop
310 LORG 2 ! Left justify & center labels
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
330 MOVE -25,Y_label ! Positions PEN at X=-25, Y=Y_label
340 LABEL "#";Y_label/10;"K" ! LABELs text
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
390 MOVE 5,90 ! Positions PEN at X=5, Y=90
400 CSIZE 5,.7 ! Sets Character SIZE and aspect ratio
410 LDIR -90 ! Labels down
420 LABEL "Script Royalties" ! LABELs text
430 MOVE 15,10 ! Positions PEN at X=15, Y=10
440 LDIR 0 ! Labels left to right
450 CSIZE 8,.5,13 ! Sets the SIZE, aspect ratio, and slant
460 LORG 3 ! Positions the label
470 LABEL USING "K";"Faux-Pas Productions" ! LABELs text
480 Copies: !
485 POINTER 75,80,2 ! Positions the CURSOR
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

CURSOR Statement

The `CURSOR` statement returns the cursor's coordinates and the pen status information to the specified variables, without pausing program execution.

Syntax:

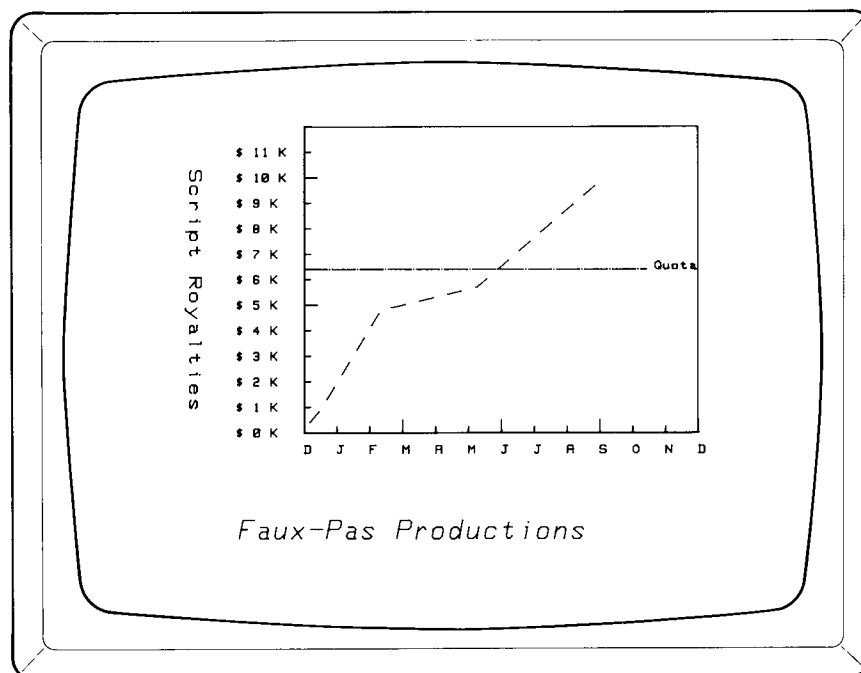
```
CURSOR X variable, Y variable [, pen status string variable]
```

The returned X and Y values are the coordinate values of the cursor in current units.

The optional third variable must be a string variable or substring. If the pen status string variable is present, it is assigned the character 0 or 1 depending on the pen's up or down status. If the pen is up, 0 is assigned to the string variable and if the pen is down, 1 is assigned to the string variable.

Remember, the pen and the cursor are NOT the same thing. On some plotters, they are physically the same (e.g. HP 9872A Plotter), but the logical location of the pen and the cursor are kept track of separately by the computer.

Example



```

1      REM *** This is the continuing example (CONT29)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 AXES 12,10,0,0,3,5,6 ! TICKS every 12 X units;every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKS every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
190 MOVE 0,64 ! Positions PEN at X=0, Y=64
200 DRAW 125,64 ! Draws line to X=125, Y=64
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
230 LABEL " Quota" ! LABELs text
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
270 MOVE X_label,-8 ! Positions PEN at X=X_label,Y=-8
280 READ Label$ ! READs data
290 LABEL Label$ ! LABELs text
300 NEXT X_label ! Repeats loop
310 LORG 2 ! Left justify & center labels
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
330 MOVE -25,Y_label ! Positions PEN at X=-25, Y=Y_label
340 LABEL "#";Y_label/10;"K" ! LABELs text
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
390 MOVE 5,90 ! Positions PEN at X=5, Y=90
400 CSIZE 5,.7 ! Sets Character SIZE and aspect ratio
410 LDIR -90 ! Labels down
420 LABEL "Script Royalties" ! LABELs text
430 MOVE 15,10 ! Positions PEN at X=15, Y=10
440 LDIR 0 ! Labels left to right
450 CSIZE 8,.5,13 ! Sets the SIZE, aspect ratio, and slant
460 LORG 3 ! Positions the label
470 LABEL USING "K";"Faux-Pas Productions" ! LABELs text
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
● 495 CURSOR X,Y ! Obtains current X,Y CURSOR coordinates
496 PRINT X,Y ! PRINTs the X,Y coordinates on the CRT
500 END

```

DIGITIZE Statement

The `DIGITIZE` statement pauses the program and allows you to reposition the cursor and then waits for you to press   or a special function key. Once this is done, the coordinate values of the cursor (in current units) are assigned to the first two variables specified in the digitize statement. The third variable, if specified, is assigned the pen's status information. The cursor type which is used in the `DIGITIZE` statement is the one which was specified last in a `POINTER` statement.

Syntax:

```
DIGITIZE X variable, Y variable[, pen status string variable]
```

This statement is similar to the computer's `INPUT` statement except that a minimum of two numeric variables are assigned values from the active plotter. The other difference is that if a `DIGITIZE` statement (or a `LIMIT` or `LOCATE`) is waiting for coordinates to be entered, you cannot `EDIT` the program until either the `DIGITIZE` statement has been completed or the program halted by pressing .

The optional third variable must be a string variable or a substring. If the pen status string variable is present, it is assigned the character 0 or 1, depending on the pen being up (0) or down (1), just as in the `CURSOR` statement.

On most plotting devices, the pen becomes the cursor for digitizing. On the CRT, the cursor is separate from what would be considered the pen. The cursor on the CRT has the lower-left corner of the CRT as its default position. If you wish the cursor to be positioned elsewhere, execute a `POINTER` statement for the desired location prior to executing the `DIGITIZE` statement.

Digitizing from the CRT

There are two types of cursors available on the CRT: a small blinking cross and full-screen crossed lines. The cursor type is selected from the last specified cursor selection in either the `POINTER` or its default value set by the `PLOTTER IS` or `LIMIT` statement.

To move the CRT cursor press the , , , or  keys as needed. The shift of the keys moves the cursor one dot row each time the key is pressed, the unshift of these keys moves the cursor one character cell. Press  to move the cursor to the lower-left corner.

To enter a digitized point from the CRT press  or  or one of the special function keys. If  is pressed the computer takes the coordinate of the digitized point and then continues program execution. If a special function key is pressed instead of the continue key, the computer takes the digitized point and performs the operation as defined by the special function key. Refer to the System 45B Operating and Programming Manual for more information on the operations and definitions of special function keys.

If the `LIMIT`, `LOCATE`, or `CLIP` statements are executed without parameters, the computer waits for the two opposite corners to be entered while in a digitize mode. The cursor on the CRT can be moved and the point entered as described in the previous two paragraphs.

Digitizing with the 9872A

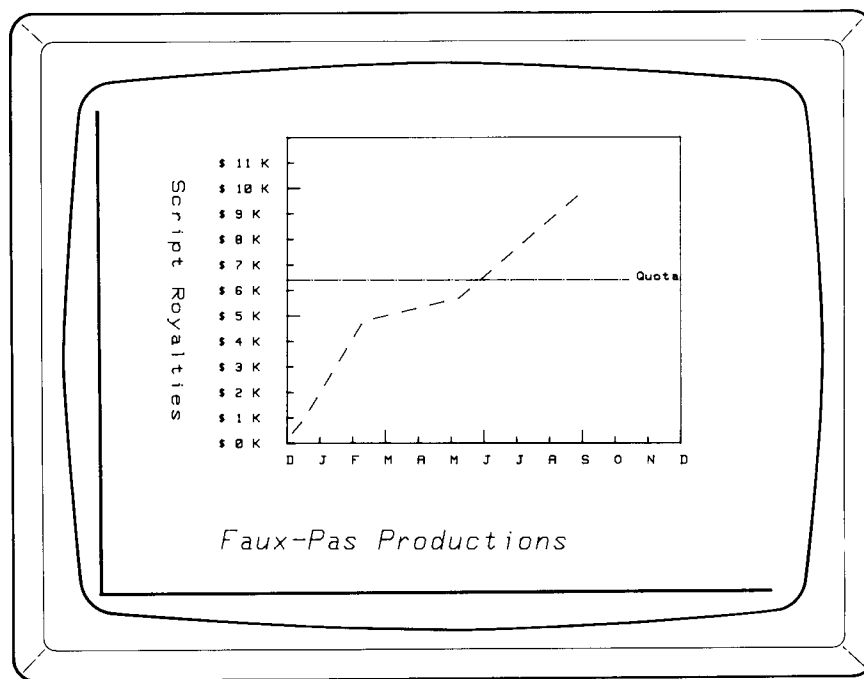
On most plotting devices, the pen is used for digitizing. On the 9872A, a special digitizing sight is optional for digitizing operations. This sight can be loaded into one of the pen stalls and is selected with a `PEN` statement for use with the `DIGITIZE` statement.

When a `DIGITIZE` statement is executed, the plotter's pen arm moves to the last specified cursor position. The  key indicator lights and the program execution pauses. The pen should then be positioned over the point to be digitized by using the pen arm control keys on the plotter's front panel. Press the  key to send the coordinate values of the point to the computer. The program execution automatically resumes after entering the coordinates.

Digitizing with an Incremental Plotter

Points are digitized on the Incremental Plotter in a similar manner to digitizing points on the CRT. The incremental plotter's cursor is moved by the four arrow keys (   ) on the System 45B, just as you move the cursor on the CRT. When the cursor is in the proper position, press  to enter the point's X,Y coordinate.

Example



```

1    REM *** This is the continuing example (CONT30)
10 Set up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 AXES 12,10,0,0,3,5,6 ! TICKS every 12 X units;every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKS every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
190 MOVE 0,64 ! Positions PEN at X=0, Y=64
200 DRAW 125,64 ! Draws line to X=125, Y=64
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
230 LABEL " Quota" ! LABELs text
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
270 MOVE X_label,-8 ! Positions PEN at X=X_label,Y=-8
280 READ Label$ ! READs data
290 LABEL Label$ ! LABELs text
300 NEXT X_label ! Repeats loop
310 LONG 2 ! Left justify & center labels
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
330 MOVE -25,Y_label ! Positions PEN at X=-25, Y=Y_label
340 LABEL "$";Y_label/10;"K" ! LABELs text
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
390 MOVE 5,90 ! Positions PEN at X=5, Y=90
400 CSIZE 5,.7 ! Sets Character SIZE and aspect ratio
410 LDIR -90 ! Labels down
420 LABEL "Script Royalties" ! LABELs text
430 MOVE 15,10 ! Positions PEN at X=15, Y=10
440 LDIR 0 ! Labels left to right
450 CSIZE 8,.5,13 ! Sets the SIZE, aspect ratio, and slant
460 LONG 3 ! Positions the label
470 LABEL USING "K";"Faux-Pas Productions" ! LABELs text
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
●495 DIGITIZE X,Y ! Obtains the X,Y CURSOR coordinates
496 PRINT X,Y ! PRINTs the X,Y coordinates on the CRT
500 END

```

WHERE Statement

The `WHERE` statement returns the pen's coordinate values of the last plotted or moved-to point. These values are in current units. If the last point is outside the clip boundaries the returned value does not reflect the current physical position of the pen but rather the logical position of the pen (its last plotted or moved-to point).

Syntax:

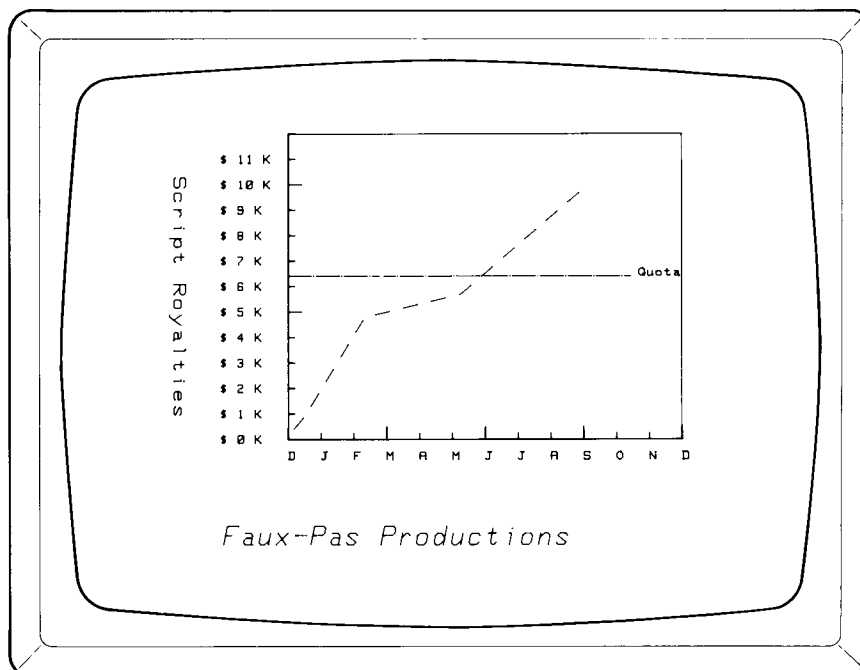
```
WHERE X variable, Y variable[, pen status string variable]
```

The returned X and Y values are the logical coordinates of the pen position in current units. If the pen was sent outside of the soft clip area, its coordinates are returned even though it is outside the soft clip area.

The optional third variable must be a string variable or a substring. If the pen status string variable is present, it is assigned the character 0 or 1, depending on whether the pen is up or down, similar to the `CURSOR` and `DIGITIZE` statements.

Press the space bar after running this example, because the X,Y coordinates are printed on the CRT.

Example



```

1      REM *** This is the continuing example (CONT31)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 AXES 12,10,0,0,3,5,6 ! TICKs every 12 X units;every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKs every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
190 MOVE 0,64 ! Positions PEN at X=0, Y=64
200 DRAW 125,64 ! Draws line to X=125, Y=64
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
230 LABEL " Quota" ! LABELs text
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
270 MOVE X_label,-8 ! Positions PEN at X=X_label,Y=-8
280 READ Label$ ! READs data
290 LABEL Label$ ! LABELs text
300 NEXT X_label ! Repeats loop
310 LORG 2 ! Left justify & center labels
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
330 MOVE -25,Y_label ! Positions PEN at X=-25, Y=Y_label
340 LABEL "#";Y_label/10;"K" ! LABELs text
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
390 MOVE 5,90 ! Positions PEN at X=5, Y=90
400 CSIZE 5,.7 ! Sets Character SIZE and aspect ratio
410 LDIR -90 ! Labels down
420 LABEL "Script Royalties" ! LABELs text
430 MOVE 15,10 ! Positions PEN at X=15, Y=10
440 LDIR 0 ! Labels left to right
450 CSIZE 8,.5,13 ! Sets the SIZE, aspect ratio, and slant
460 LORG 3 ! Positions the label
470 LABEL USING "K";"Faux-Pas Productions" ! LABELs text
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
●495 WHERE X,Y ! Obtains the X,Y PEN coordinates
496 PRINT X,Y ! PRINTs the X,Y coordinates on the CRT
500 END

```

Summary

This chapter covered the following statements

- `POINTER`
- `CURSOR`
- `DIGITIZE`
- `WHERE`

Other information in this chapter covered the difference between the pen and the cursor, the difference between plotting and digitizing, how to position and change the cursor type using the `POINTER` statement, how to obtain the current cursor location by using the `CURSOR` statement, how to enter an X,Y coordinate by using the `DIGITIZE` statement, and how to find the current position of the pen by using the `WHERE` statement.

The final chapter in this manual is concerned with some advanced graphic techniques such as loading and storing the entire graphics display, and curve smoothing. There are also some other example programs which you may find interesting in an approach to problem solving as well as in the way certain statements are used.

Advanced Graphic Techniques

Page 124 • **GLOAD** (loads integer array from R/W memory to CRT/graphics memory)

Page 124 • **GSTORE** (stores CRT/graphics memory array into integer array in R/W memory)

Terms

• **Reflected Plot** — A plot produced by interchanging either the X min, X max coordinates, the Y min, Y max coordinates or both X and Y coordinate pairs to change the plot.

• **HPGL** — The instruction set resident within the HP 9872A Graphics Plotter.

Reflected Plots

Parameter Sequence	Resulting Plot
X min, X max, Y min, Y max	Normal
X max, X min, Y min, Y max	Reflects Across Y Axis
X min, X max, Y max, Y min	Reflects Across X Axis
X max, X min, Y max, Y min	Reflects Through the Origin

Array Specifications for GLOAD and GSTORE

Array Type — Integer

Maximum Number of Elements — 16 381

Pointer — First Array Element

Chapter 7

GLOAD and GSTORE Statements

The GLOAD and GSTORE statements allow you to copy the CRT graphics memory to or from an integer array within the computer's R/W memory. The CRT graphics memory contains the data that produces dots, which make up the CRT graphics display. When copied into the computer's R/W memory the data can be stored (GSTORE), manipulated, MAT PRINT onto a mass storage device, MAT READ from a mass storage device, and then later copied back (GLOAD) into the CRT memory for display.

A complex graphics display may take several minutes to generate using the plotting statements, but by using GSTORE and then later GLOAD, the same display can be re-displayed in a fraction of a second.

Syntax:

```
GLOAD integer array name (*)
or
GSTORE integer array name (*)
```

The integer array used with these statements must have 36 elements for each horizontal dot row, plus one extra element for a pointer. The pointer is the first element of the array.

If you want to store the entire graphics display, you need to dimension an integer array with 16 381 elements. One element is for the pointer, plus 36 elements for each row, and there are 455 rows ($1 + 36 \times 455 = 16\,381$). If you are using OPTION BASE 0 the subscript of the first element of the array is zero (e.g., Array(0)) and the dimension size could be as shown.

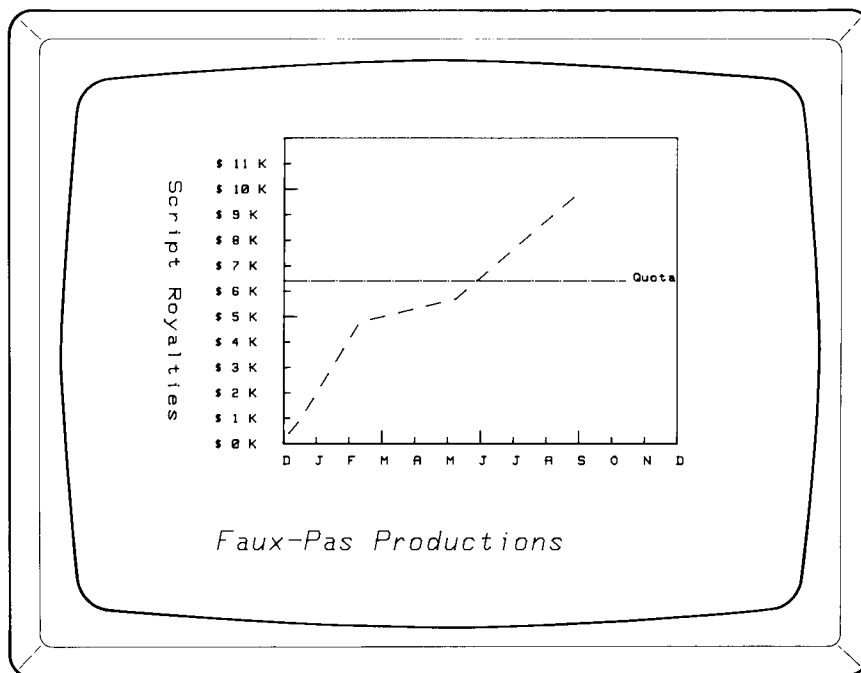
```
10 OPTION BASE 0
20 INTEGER A(16380)
```

Zero through 16 380 equals 16 381 elements.

If it is not necessary to copy the entire graphics display you can dimension the integer array smaller. An integer array with 8 173 elements ($1 + 227 \times 36 = 8\,173$) holds about one half of the graphics display (227 dot rows).

For example, by adding the following program lines to your continuing example, the entire CRT graphics display could be stored into an array which is called "Array". As you can see in the program, the contents of "Array" are then stored away onto the mass storage device. At some later time, you could then read the data file containing the contents of "Array" back into your System 45B, and by using a `GLOAD` statement, re-display your CRT picture.

Example



```

1      REM *** This is the continuing example (CONT32)
10 Set_up: !
●15  INTEGER Array(16380) ! Dimension Array for GSTORE
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start_plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 AXES 12,10,0,0,3,5,6 ! TICKs every 12 X units;every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKs every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
190 MOVE 0,64 ! Positions PEN at X=0, Y=64
200 DRAW 125,64 ! Draws line to X=125, Y=64
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
230 LABEL " Quota" ! LABELs text
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
270 MOVE X_label,-8 ! Positions PEN at X=X_label,Y=-8
280 READ Label$ ! READs data
290 LABEL Label$ ! LABELs text
300 NEXT X_label ! Repeats loop
310 LONG 2 ! Left justify & center labels
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
330 MOVE -25,Y_label ! Positions PEN at X=-25, Y=Y_label
340 LABEL "$";Y_label/10;"K" ! LABELs text
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
390 MOVE 5,90 ! Positions PEN at X=5, Y=90
400 CSIZE 5,.7 ! Sets Character SIZE and aspect ratio
410 LDIR -90 ! Labels down
420 LABEL "Script Royalties" ! LABELs text
430 MOVE 15,10 ! Positions PEN at X=15, Y=10
440 LDIR 0 ! Labels left to right
450 CSIZE 8,.5,13 ! Sets the SIZE, aspect ratio, and slant
460 LONG 3 ! Positions the label
470 LABEL USING "K";"Faux-Pas Productions" ! LABELs text
480 Copies: !
●485 GSTORE Array(*) ! Stores Graphics Memory in Array
486 ASSIGN #1 TO "Array" ! Assigns MAT PRINT # to Array
487 MAT PRINT #1;Array ! Prints data onto mass storage
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

—



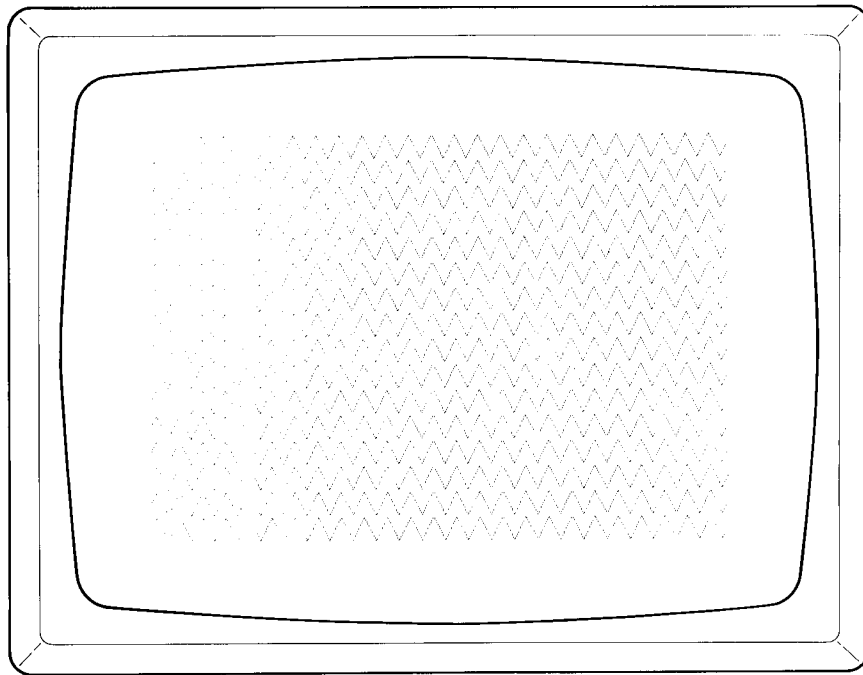
The following expression can be used to compute the starting address for a given dot row. Note: the assigned value must be negative.

$$A(0) = -(\text{Row} \times 36 + 1)$$

where the variable Row is the number (0 to 454) that you want to start the load or store operation. The top row number is zero (0) and the bottom row number is 454.

The program produces a pattern across the top of the CRT at the normal line generating speed. Then by using `GSTORE` and `GLOAD`, the pattern is repeated 15 times down the screen (25 dot rows at a time). As you can see, the pattern is repeated faster than it can be generated the first time.

Example



```

1    REM *** This is the GLOAD/GSTORE example (GL_GS)
10   ! This program draws a pattern across the top of the CRT using the normal
20   ! plotting statements. The pattern is then repeated down the screen using
30   ! the GSTORE and GLOAD statements. The pattern is repeated much faster
40   ! than it was drawn the first time.
50   Set_up: !
60   OPTION BASE 0                ! Array begins with 0
70   INTEGER Array(900)           ! Dimension INTEGER Array for GSTORE
80                               ! ( 901 elements total )
90   PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT
100  GRAPHICS                     ! Enables the CRT
110  DEG                          ! Set DEGREES mode
120  SCALE 0,25*360,-10,10        ! Scales the LOCATE rectangle to 9000x20
130  Start_plot: !
140  ! ***** The pattern is drawn here
150  MOVE 0,9.5                   ! Positions PEN at X=0, Y=9.5
160  FOR X=0 TO 9000 STEP 90      ! Initializes loop
170  DRAW X,.5*SIN(X)+9.5         ! DRAWS line
180  NEXT X                       ! Repeats loop
190  Store_array: !
200  ! ***** The pattern is STOREd in the Array
210  Array(0)=-1                  ! Specifies GSTORE start point
●220  GSTORE Array(*)             ! GSTORE Graphics Memory into Array
230  Recall_array: !
240  ! ***** The pattern is repeated down the CRT 15 times using GLOAD
250  Width=25                     ! Specifies the width of the pattern
260  FOR Pattern=1 TO 15          ! Initializes loop
270  Array(0)=- (Pattern*Width*36+1) ! Specifies GLOAD start point
●280  GLOAD Array(*)             ! GLOAD the pattern into Graphics memory
290  NEXT Pattern                ! Repeats loop
300  END

```

Reflective Plotting

Reflected plots are those that appear to be upside down, backwards or mirror images of what would be considered a normal plot. Sometimes it is desirable to produce different types of reflected plots, such as, when you want to draw on the back side of clear plastic or film to produce a right reading image when read from the front side. This technique is very useful in slide generation because the resulting slides are more resistant to handling or wear in addition to allowing you to mark and erase marks along the top side of the slide.

Reflected plots are produced by exchanging one or more sets of parameters in the following Graphics statements.

```

LIMIT
SHOW
SCALE

```

With each of these three statements, the parameters are in the order of

X1,X2,Y1,Y2

If you enter the parameters as X2,X1,Y1,Y2 the resulting plot is reflected across the Y-axis.

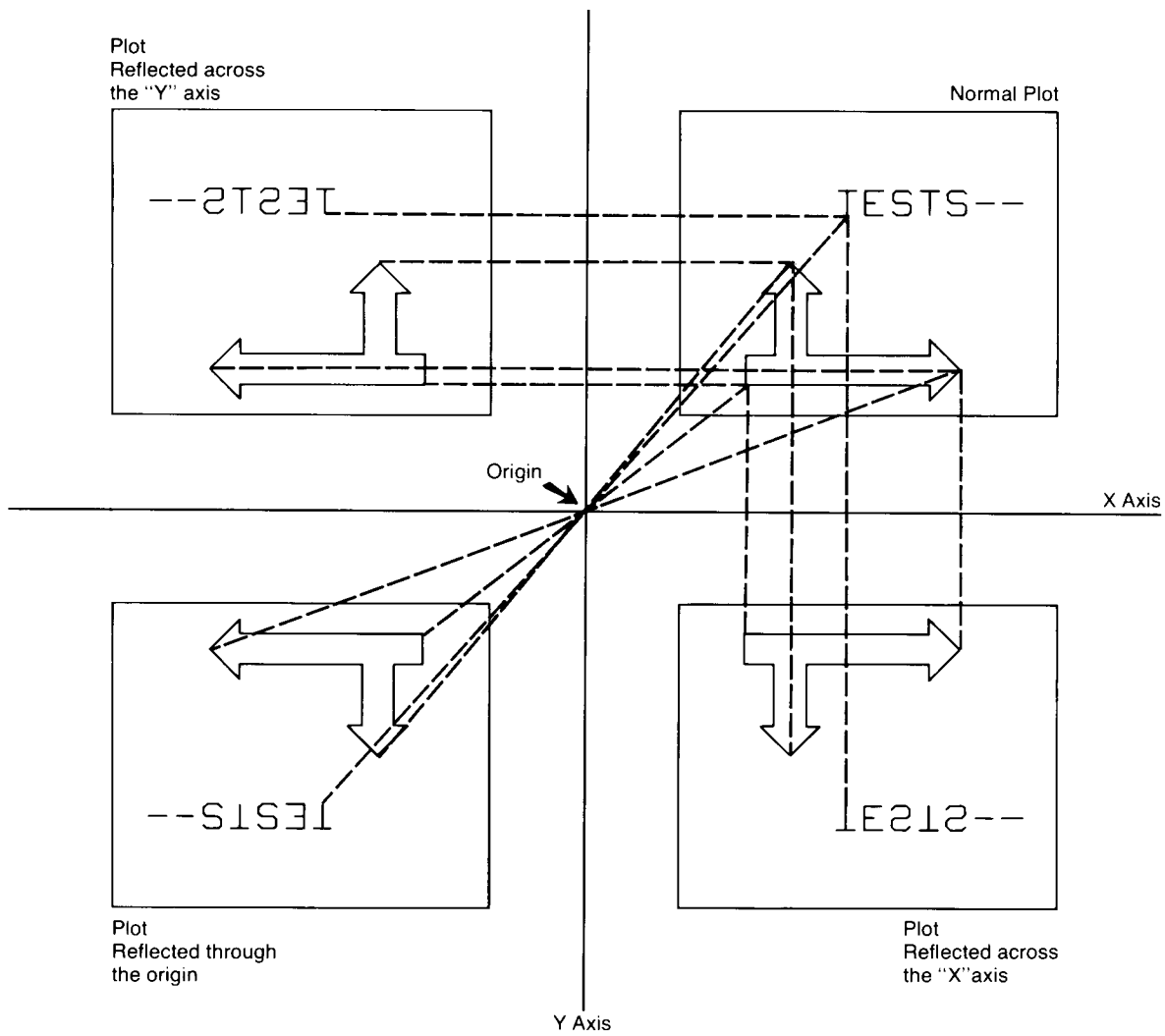
If you enter the parameters as X1, X2, Y2, Y1 the resulting plot is reflected across the X-axis.

If you enter the parameters as X2,X1,Y2,Y1 the resulting plot is reflected through its origin.

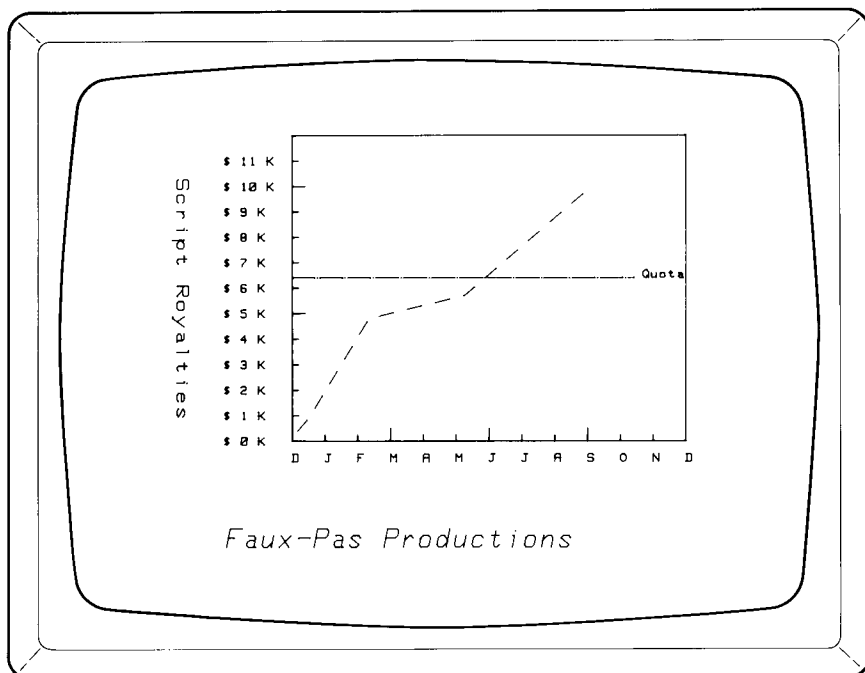
Labeling is done in the GDU's mode. Therefore, the `LIMIT` statement is the only statement which reflects labels as well as plots.

The `SHOW` and `SCALE` statements position the labels relative to the X,Y position called for by a `MOVE` statement, regardless of how that looks due to the reflection.

The following illustrations show the three types of reflected plots:



Example



```

1    REM *** This is the continuing example (CONT33)
10 Set up: !
20  DATA 2,4,6,9,28,49,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
40  GRAPHICS ! Enables the CRT
50  LIMIT 184,0,140,0 ! Invert the HARD CLIP limits to reflect
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
90 Start plot: !
100 FRAME ! Boxes the SOFT CLIP area
110 AXES 12,10,0,0,3,5,6 ! TICKS every 12 X units;every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKS every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120 LINE TYPE 4 ! Specifies dashed LINE TYPE
130 FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140 READ X_plot_point,Y_plot_point ! Obtains data coordinates
150 PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160 NEXT Loop_index ! Repeats loop
170 Enhancements: !
180 LINE TYPE 6 ! Specifies dashed LINE TYPE
190 MOVE 0,64 ! Positions PEN at X=0, Y=64
200 DRAW 125,64 ! Draws line to X=125, Y=64
210 UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220 LINE TYPE 1 ! Specifies solid LINE TYPE
230 LABEL " Quota" ! LABELs text
240 Label_axes: !
250 DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260 FOR X_label=0 TO 144 STEP 12 ! Initializes loop
270 MOVE X_label,-8 ! Positions PEN at X=X_label,Y=-8
280 READ Label$ ! READs data
290 LABEL Label$ ! LABELs text
300 NEXT X_label ! Repeats loop
310 LONG 2 ! Left justify & center labels
320 FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
330 MOVE -25,Y_label ! Positions PEN at X=-25, Y=Y_label
340 LABEL "$";Y_label/10;"K" ! LABELs text
350 NEXT Y_label ! Repeats loop
360 Label_text: !
370 DEG ! Change to degrees mode
380 SETGU ! Change back to Graphic Display Units
390 MOVE 5,90 ! Positions PEN at X=5, Y=90
400 CSIZE 5,.7 ! Sets Character SIZE and aspect ratio
410 LDIR -90 ! Labels down
420 LABEL "Script Royalties" ! LABELs text
430 MOVE 15,10 ! Positions PEN at X=15, Y=10
440 LDIR 0 ! Labels left to right
450 CSIZE 8,.5,13 ! Sets the SIZE, aspect ratio, and slant
460 LONG 3 ! Positions the label
470 LABEL USING "K";"Faux-Pas Productions" ! LABELs text
480 Copies: !
490 DUMP GRAPHICS ! Hard copy of CRT display
500 END

```

More About the HP 9872A Plotter

The HP 9872A Plotter responds to a graphics language called HPGL. The advantage of this language set is that the 9872A can be driven by any controller or computer which can output these HPGL instructions using proper I/O formats. The advantage of the unified graphics statements which are presented in this manual is that the unified graphics language allows you to go from one plotting device to another plotting device by changing only one statement, as opposed to changing every HPGL statement in a program.

Some people may wish to incorporate some of the HPGL instructions which are found in Section IV of the HP 9872A Graphics Plotter Operating and Service Manual.

While all HPGL instructions are accessible to you, you may probably need only the following three HPGL instructions.

- VA
- VN
- VS

VA (velocity adaptive) allows the pen to plot incoming data at rates approaching the incoming data rate. Remember that the physical limit of the pen is 36 cm/sec.

VS (velocity selection) sets the pen speed for each pen at integer intervals of between 1 cm/sec to 36 cm/sec. You have the option of sending this instruction to apply for just one pen, or all pens. The VS instruction is very useful for slowing down the pen speed to improve the quality of your plotted lines on the HP9872A.

VN (velocity normal) returns the plotting speed to 36 cm/sec. It is used to clear the VA and VS instructions from the 9872A.

Once you change the pen speed, consider the fact that a complex plot which takes 1 minute to plot on the 9872A at 36 cm/sec, is going to take 36 minutes to plot at 1 cm/sec.

The method to use to send an HPGL instruction is simple –

- `PRINTER IS 7,5`
- `PRINT "HPGL instruction"`

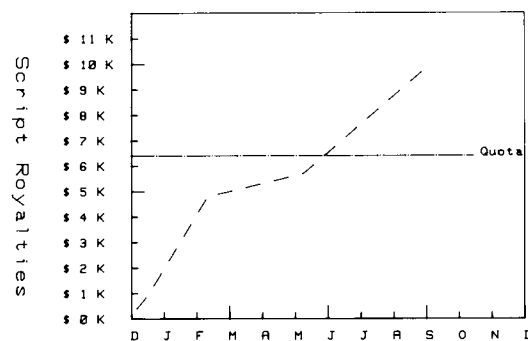
inserted after you go into the `GRAPHICS` mode of your program, with a `PRINT` for each individual HPGL instruction you wish to use.

The only problem which may arise as a result of this is that executing the `PLOTTER IS` or the `LIMIT` statement in the program clears out all HPGL instructions to their default conditions (i.e. pen speed to 36 cm/sec.). Therefore, you would either have to

1. Not execute the `PLOTTER IS` or `LIMIT` statements.
- or
2. Put the `PRINTER IS` and `PRINT` statements **after** the `PLOTTER IS` and `LIMIT` statements.

The following example has been provided to assist you. Remember, if you want to clear out any of the HPGL instructions, just execute either a `PLOTTER IS` or a `LIMIT` statement.

Example



Faux-Pas Productions

```

1      REM *** This is the continuing example (CONT34)
10 Set_up: !
20  DATA 2,4,6,9,28,48,63,57,109,99.5 ! Data for example plot
30  PLOTTER IS 7,5,"9872A" ! Specifies the 9872A
40  GRAPHICS ! Enables the CRT
50  LIMIT 0,184,0,140 ! Defines the HARD CLIP limits area
60  LOCATE 30,120,30,100 ! Defines the area to be scaled
70  CLIP 30,120,30,100 ! Defines the SOFT CLIP area
80  SCALE 0,144,0,120 ! SCALES the LOCATE area to 144x120
●85  PRINTER IS 7,5 ! Specify 9872A to receive HPGL command
●86  PRINT "VS2" ! Slow plotter to 2 cm/sec
90 Start_plot: !
100  FRAME ! Boxes the SOFT CLIP area
110  AXES 12,10,0,0,3,5,6 ! TICKs every 12 X units;every 10 Y units
111 ! AXES intersection at X=0, Y=0
112 ! Major TICKs every 3 X; every 5 Y
113 ! Major TICK length 6 GDU's
120  LINE TYPE 4 ! Specifies dashed LINE TYPE
130  FOR Loop_index=1 TO 5 ! Begins loop to PLOT 5 points
140  READ X_plot_point,Y_plot_point ! Obtains data coordinates
150  PLOT X_plot_point,Y_plot_point ! PLOTS data coordinates
160  NEXT Loop_index ! Repeats loop
170 Enhancements: !
180  LINE TYPE 6 ! Specifies dashed LINE TYPE
190  MOVE 0,64 ! Positions PEN at X=0, Y=64
200  DRAW 125,64 ! Draws line to X=125, Y=64
210  UNCLIP ! Sets SOFT CLIP area to HARD CLIP area
220  LINE TYPE 1 ! Specifies solid LINE TYPE
230  LABEL " Quota" ! LABELs text
240 Label_axes: !
250  DATA D,J,F,M,A,M,J,J,A,S,O,N,D ! Data for X axis labels
260  FOR X_label=0 TO 144 STEP 12 ! Initializes loop
270  MOVE X_label,-8 ! Positions PEN at X=X_label,Y=-8
280  READ Label$ ! READs data
290  LABEL Label$ ! LABELs text
300  NEXT X_label ! Repeats loop
310  LORG 2 ! Left justify & center labels
320  FOR Y_label=0 TO 115 STEP 10 ! Initializes loop
330  MOVE -25,Y_label ! Positions PEN at X=-25, Y=Y_label
340  LABEL "$";Y_label/10;"K" ! LABELs text
350  NEXT Y_label ! Repeats loop
360 Label_text: !
370  DEG ! Change to degrees mode
380  SETGU ! Change back to Graphic Display Units
390  MOVE 5,90 ! Positions PEN at X=5, Y=90
400  CSIZE 5,.7 ! Sets Character SIZE and aspect ratio
410  LDIR -90 ! Labels down
420  LABEL "Script Royalties" ! LABELs text
430  MOVE 15,10 ! Positions PEN at X=15, Y=10
440  LDIR 0 ! Labels left to right
450  CSIZE 8,.5,13 ! Sets the SIZE, aspect ratio, and slant
460  LORG 3 ! Positions the label
470  LABEL USING "K";"Faux-Pas Productions" ! LABELs text
480 Copies: !
490  DUMP GRAPHICS ! Hard copy of CRT display
500  END

```

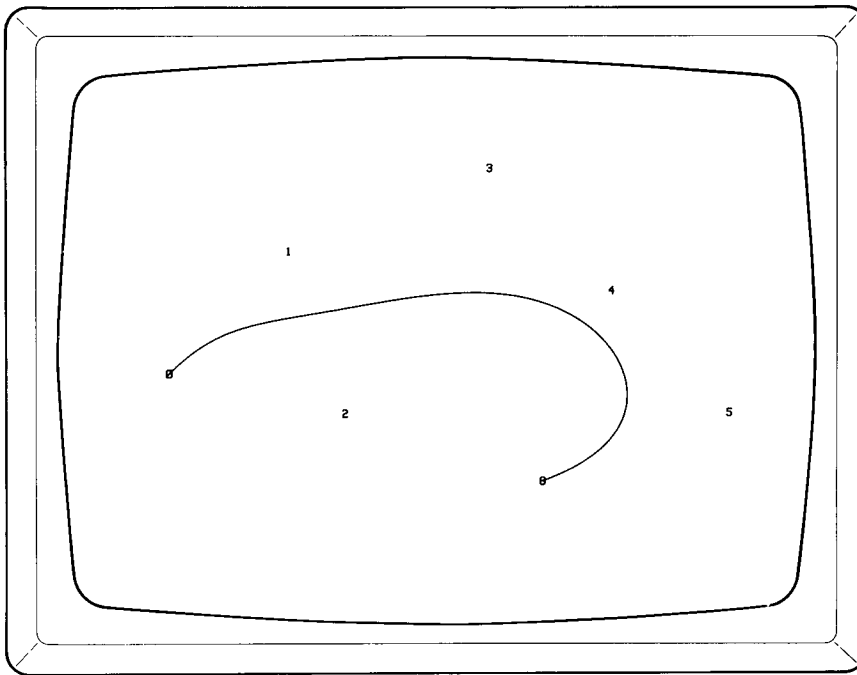
Curve Fitting

Curve fitting is a technique by which the general tendencies of data points can be plotted, as well as the individual points which make up the data for the curve.

There are many curve-fitting applications available. Once you have decided which algorithm best suits your application, the graphic display can be used to show the results of the plot.

For this example, the Bezier curve smoothing algorithm is used. The program allows you to specify which data points are to be entered as the basis for the smoothing algorithm. When using the Bezier technique, only the first and last points actually lie on the curve. The derivatives, order and shape of the curve are determined by the other vertices.

Example



```

1    REM ** Bezier Curve Fitting
10 Set up: PLOTTER IS 13,"GRAPHICS"    ! Specifies the CRT
20 GRAPHICS                            ! Enables the CRT
30 DIM P(20,2),C(20)                  ! Initializes Arrays
40 LONG 5                              ! Center all LABELs
50 CSIZE 1.7                           ! Sets CSIZE to 1.7 GDU's
60 Data input:INPUT "Order of Curve",N ! Number of entries
70 IF (N<0) OR (N>19) THEN 60         ! Error checking for wrong value
80 Ix=5                                ! Initialize variable Ix
90 Iy=0                                ! Initialize variable Iy
100 FOR I=0 TO N                       ! Initializes the loop
110 DIGITIZE P(I,1),P(I,2)             ! Inputs an X,Y coordinate
120 MOVE P(I,1),P(I,2)                 ! MOVES CURSOR to X,Y coordinate
130 LABEL USING "D";I                  ! LABEL X,Y coordinate
140 C(I)=FNFact(N)/(FNFact(I)*FNFact(N-I)) ! Calculate value
150 NEXT I                             ! Repeats LOOP for NEXT Input
160 PENUP                              ! Raises PEN
170 FOR T=0 TO 1 STEP .005             ! Loop to PLOT curve
180 X=Y=0                              ! Initializes variables
190   FOR I=0 TO N                     ! Loop to PLOT point
200   X=X+P(I,1)*C(I)*(T^I*(1-T)^(N-I)) ! Calculate X value
210   Y=Y+P(I,2)*C(I)*(T^I*(1-T)^(N-I)) ! Calculate Y value
220   NEXT I                           ! Repeats inner Loop
230 PLOT X,Y                           ! PLOT smoothed X,Y coordinate
240 NEXT T                             ! Repeats outer Loop
250 FRAME                              ! FRAMES the SOFT CLIP area
260 END
270 ! The following section defines the function
280 DEF FNFact(Z)
290 IF Z=0 THEN RETURN 1
300 RETURN Z*FNFact(Z-1)
310 FNEED

```

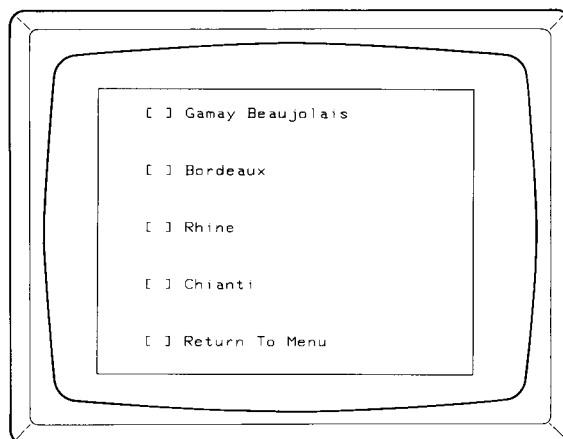
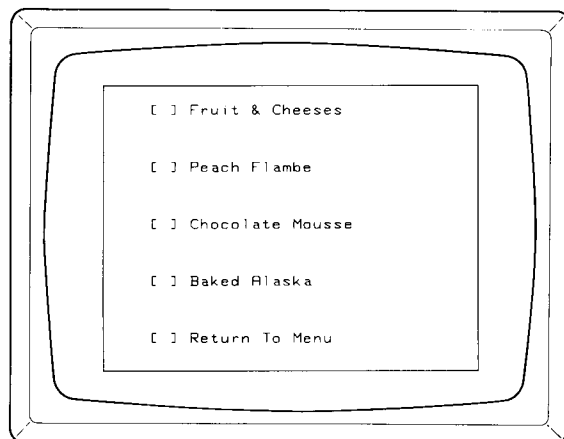
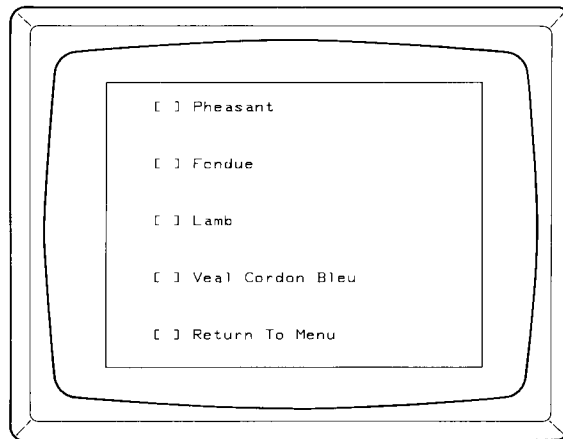
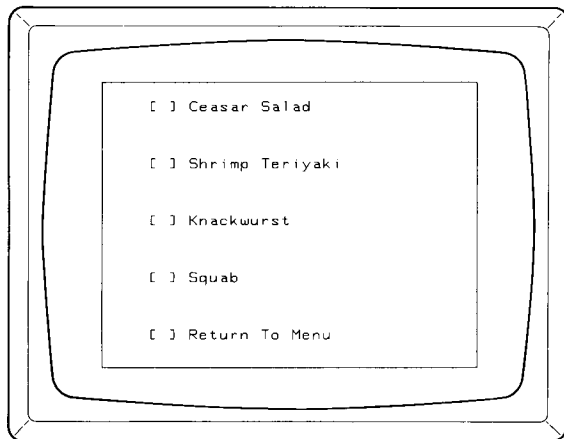
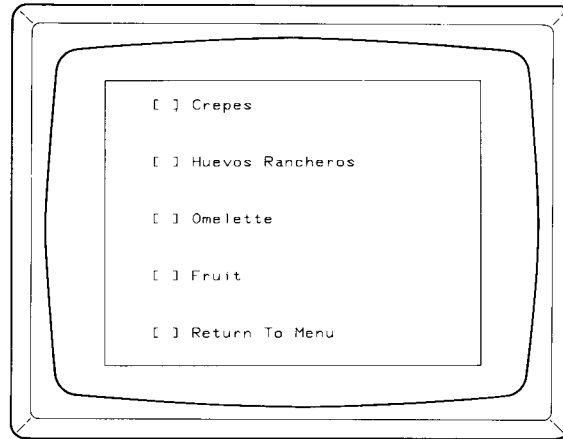
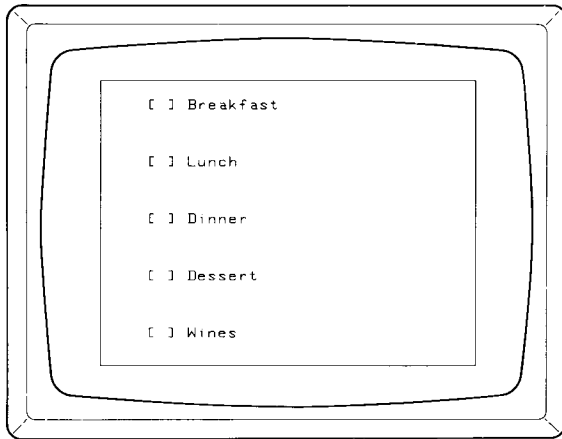
Menu Selection

Menu selection is a graphics technique which can prove useful in some applications. Its primary value is as an interactive tool which enables the operator of the computer to perform things which require a minimum of programming knowledge, similar to an **ON KEY** statement.

The menu selection technique is quite simple in implementation: present a choice of items, allow the operator to position the cursor at a selected item and press  to digitize the X,Y coordinate, then determine which menu item the X,Y coordinate represents and implement the appropriate program steps.

In this first example, the menu is used to provide the operator with a menu (literally). Since this is a simple application, the Y coordinate value is the only coordinate being checked. You should note that the **SCALE** statement is set up and the label is placed to provide minimum complications caused by positioning the cursor on the boundary between two adjacent items.

Examples

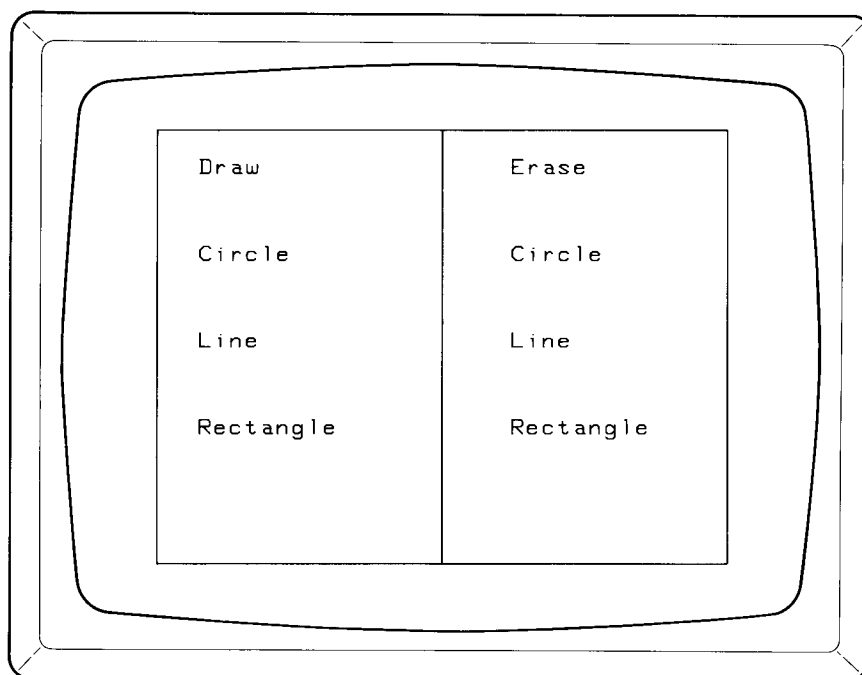


```

1      REM ** This is the Menu plot (MENU)
10 Set_up: !
20      Reselect=0 ! Initializes Variable
30      PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT
40      GRAPHICS ! Enables the CRT
50      SCALE 0,2,0,10 ! SCALES to 2x10
60      LONG 1 ! Positions LABELs
70      CSIZE 6 ! Sets Character SIZE
80      RESTORE 90 ! Positions DATA pointer
90      DATA Breakfast,Lunch,Dinner,Dessert,Wines
100     GOSUB Choices
110     GOSUB Find_it
●120    IF (Y>8) AND (Y<10) THEN GOSUB Breakfast ! Check value of Y
●130    IF (Y>6) AND (Y<8) THEN GOSUB Lunch ! coordinate to
●140    IF (Y>4) AND (Y<6) THEN GOSUB Dinner ! implement the proper
●150    IF (Y>2) AND (Y<4) THEN GOSUB Dessert ! operation
●160    IF (Y>0) AND (Y<2) THEN GOSUB Wines !
170    IF Reselect=1 THEN Set_up
180    END
190 Selection: !
200         GOSUB Choices
210         GOSUB Find_it
220         IF (Y>0) AND (Y<2) THEN Reselect=1
230         Y=0 ! Clear Y to prevent recursion
240         RETURN
250 Find_it: !
260         POINTER .10,5,-2 ! Positions CURSOR
270         DIGITIZE X,Y ! Inputs coordinate
280         GCLEAR ! Clears the Graphics memory & CRT
290         RETURN
300 Choices: !
310     FRAME ! FRAMES plotting area
320         FOR Cycle=9 TO 1 STEP -2 ! Initializes the loop
330             READ Meal$ ! Obtains text
340             MOVE .25,Cycle ! Positions the PEN
350             LABEL "[ ] ";Meal$ ! LABELs text
360             NEXT Cycle ! Repeats the loop
370             RETURN
380 Breakfast: !
390     RESTORE 400 ! Positions the DATA pointer
400     DATA Crepes,Huevos Rancheros,Omelette,Fruit,Return To Menu
410     GOSUB Selection
420     RETURN
430 Lunch: !
440     RESTORE 450 ! Positions the DATA pointer
450     DATA Caesar Salad, Shrimp Teriyaki,Knackwurst,Squab,Return To Menu
460     GOSUB Selection
470     RETURN
480 Dinner: !
490     RESTORE 500 ! Positions the DATA pointer
500     DATA Pheasant,Fondue,Lamb,Veal Cordon Bleu,Return To Menu
510     GOSUB Selection
520     RETURN
530 Dessert: !
540     RESTORE 550 ! Positions the DATA pointer
550     DATA Fruit & Cheeses,Peach Flambe,Chocolate Mousse,Baked Alaska,Return To Menu
560     GOSUB Selection
570     RETURN
580 Wines: !
590     RESTORE 600 ! Positions the DATA pointer
600     DATA Gamay Beaujolais,Bordeaux,Rhine,Chianti,Return To Menu
610     GOSUB Selection
620     RETURN

```

In the second example, the menu selection allows the operator to select a graphics driver routine. This routine is used in conjunction with further interaction in the form of specifying the origin and end points of the figure which is to be plotted or erased.



```

1    REM ** This is the DRIVER (DRIVER) example.
10  Overhead:
20    DEG                                ! Sets DEGrees mode
30    Reselect=0                         ! Initializes variable
40    DATA Draw,Erase,Circle,Circle,Line,Line,Rectangle,Rectangle
50    INTEGER Array(16380),Menu(16380)   ! Sets up arrays
60    PLOTTER IS 13,"GRAPHICS"           ! Specifies the CRT as the plotter
70    GRAPHICS                           ! Enables the CRT
80    SHOW 0,560,0,455                   ! Sets SOFT CLIP area to 10X10
90    LONG 5                             ! Sets LABEL origin
100   CSIZE 6                            ! Sets Character SIZE
110  Choices:
120    FRAME                             ! FRAMEs plotting area
130    MOVE 280,0                         ! MOVES the PEN
140    DRAW 280,455                       ! DRAWS a vertical line
150    FOR Cycle=9 TO 3 STEP -2           ! Begins the loop
160    READ Choice$,Duplicate$           ! Reads LABELs for line
170    MOVE 250,Cycle*40                  ! Positions LABEL
180    LABEL Choice$,Duplicate$          ! LABELs line
190    NEXT Cycle                         ! Repeats the loop
200    GSTORE Menu(*)                    ! STOREs Graphics memory

```

```

210 Set_up: !
220   PEN 1 ! Selects the PEN
230   GLOAD Menu(*) ! ReLOADS Graphics memory
240   GOSUB Find_it ! Selects an item
250   GCLEAR ! Erases Graphics memory & CRT
260   IF X>280 THEN PEN -1 ! Check X location for erase
● 270   IF (Y>240) AND (Y<320) THEN GOSUB Circle ! Determines operation
● 280   IF (Y>160) AND (Y<240) THEN GOSUB Line ! Determines operation
● 290   IF (Y>2) AND (Y<160) THEN GOSUB Rectangle ! Determines operation
300   GOTO Set_up ! Repeats for figure
310   END
320   REM ** The subroutines are in the following lines
330 Find_it: !
340     POINTER 5,5,-2 ! Positions CURSOR
350     DIGITIZE X,Y ! Inputs X,Y coordinate
360     RETURN
370 Circle: ! Draws a circle by digitizing 3 points on the circumference
380     ! When erasing the circle, choose three points which are
390     ! equidistant for best results. Jaggies may require several erasures.
400     GOSUB Recall
410     GOSUB Find_it
420     Xa=X
430     Ya=Y
440     GOSUB Find_it
450     Xb=X
460     Yb=Y
470     GOSUB Find_it
480     Xc=X
490     Yc=Y
500     Side_a=SQR(ABS(Xb-Xc)^2+ABS(Yb-Yc)^2) ! Calculates side BC
510     Side_b=SQR(ABS(Xc-Xa)^2+ABS(Yc-Ya)^2) ! Calculates side CA
520     Side_c=SQR(ABS(Xa-Xb)^2+ABS(Ya-Yb)^2) ! Calculates side AB
530     Angle_cab=ACS((Side_a^2+Side_b^2-Side_c^2)/(-2*Side_b*Side_c))
540     Avg_side=.5*(Side_a+Side_b+Side_c)
550     Radius=Side_a*Side_b*Side_c/(4*Avg_side*(Avg_side-Side_a)*TAN(Angle_cab/2))
560     M1=-1/((Ya-Yb)/(Xa-Xb))
570     M2=-1/((Yb-Yc)/(Xb-Xc))
580     Xa_prime=(Xa+Xb)/2
590     Ya_prime=(Ya+Yb)/2
600     Xb_prime=(Xb+Xc)/2
610     Yb_prime=(Yb+Yc)/2
620     B1=Ya_prime-M1*Xa_prime
630     B2=Yb_prime-M2*Xb_prime
640     Xintersect=(B1-B2)/(M2-M1)
650     Yintersect=(M2*B1-M1*B2)/(M2-M1)
660     MOVE SIN(0)*Radius+Xintersect,COS(0)*Radius+Yintersect
670     FOR I=0 TO 360 STEP 15
680       DRAW SIN(I)*Radius+Xintersect,COS(I)*Radius+Yintersect
690     NEXT I
700 Done_circle: ! Exit for circle routine
710     GOSUB Save_crt
720     X=Y=0
730     RETURN

```

```

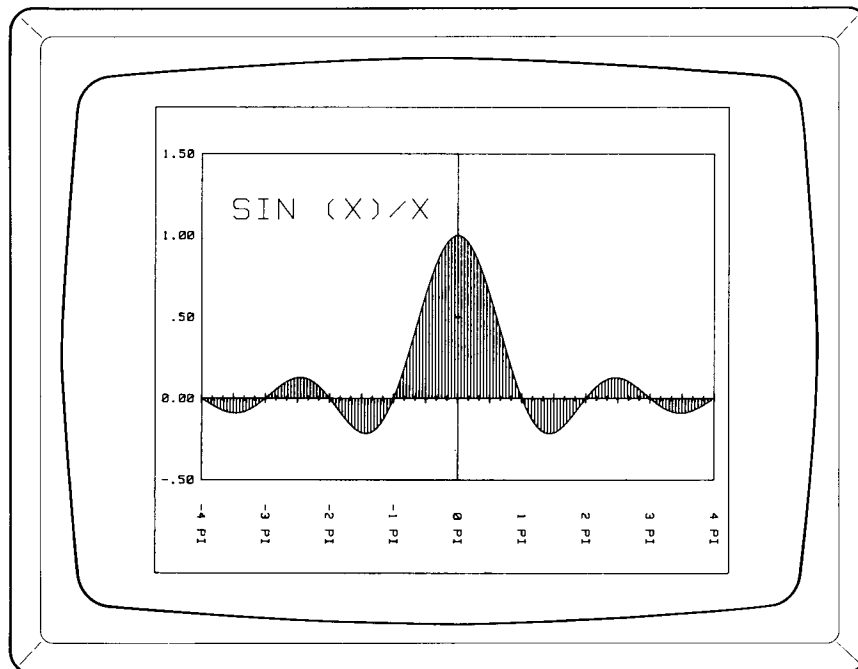
740 Line:!! Draws a line by inputting the end points
750   GOSUB Recall
760   GOSUB Find_it
770   Xorigin=X
780   Yorigin=Y
790   GOSUB Find_it
800   Xend=X
810   Yend=Y
820   MOVE Xorigin,Yorigin
830   DRAW Xend,Yend
840   GOSUB Save_crt
850   X=Y=0
860   RETURN
870 Rectangle:!! Draws a rectangle by inputting the opposite corners
880   GOSUB Recall
890   GOSUB Find_it
900   X1=X2=X
910   Y1=Y4=Y
920   GOSUB Find_it
930   X3=X4=X
940   Y2=Y3=Y
950   MOVE X1,Y1
960   DRAW X2,Y2
970   DRAW X3,Y3
980   DRAW X4,Y4
990   DRAW X1,Y1
1000  GOSUB Save_crt
1010  X=Y=0
1020  RETURN
1030 Recall:!! This loads the CRT picture from an array
1040  GLOAD Array(*)      ! ReLOADS the graphics drawing on the CRT
1050  RETURN
1060 Save_crt:!! This stores the CRT picture into an array
1070  WAIT 3000          ! Displays the picture for 3 seconds
1080  GSTORE Array(*)    ! STORES the CRT picture
1090  GCLEAR             ! Erases Graphics memory & CRT
1100  RETURN

```

Some Interesting Examples

The programs in this section are given to show some interesting capabilities and programming techniques. In most cases, each program line has a comment that describes what the line does. After running the program and understanding how it works, change a few lines to make the program do something different to fit one of your applications.

SIN(X)/X Example



```

1  REM ** This is the example for SIN(X)/x (SIN(X))
10  ! This program plots the SINE(X)/X from -4PI to +4PI.
20  ! The program waits for you to move the CURSOR to DIGITIZE
30  ! a point, and then prints the "X" and "Y" values.
40  !
50  !
60  PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT as the plotter
70  GRAPHICS                     ! Enables the CRT
80  FRAME                       ! FRAMES the SOFT CLIP area
90  PRINTER IS 0                 ! Specifies the Thermal Printer
100 RAD                          ! Sets the radians mode
110 LOCATE 10,120,20,90         ! Defines area to be SCALED

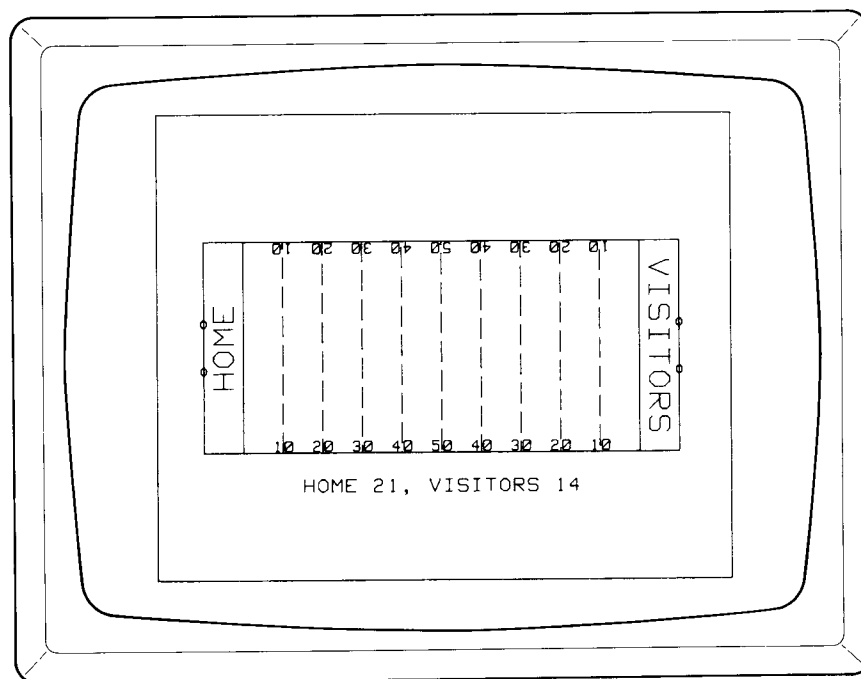
```

```

120 !
130 ! ***** Specify maximum and minimum values for user units
140 Xmin=-4*PI
150 Xmax=4*PI
160 Ymin=-.5
170 Ymax=1.5
180 !
190 ! ***** Specify user units, draw AXES and FRAME graph
200 SCALE Xmin,Xmax,Ymin,Ymax ! SCALE graph with user units
210 AXES PI/6,.5,0,0,3,10 ! Draw AXES
220 FRAME ! FRAME the plotting area
230 !
240 ! ***** LABEL axes, using subroutines
250 GOSUB Lxaxes ! LABEL X axis
260 GOSUB Lyaxes ! LABEL Y axis
270 !
280 ! ***** PLOT function with fill to base line
290 FOR X=Xmin TO Xmax STEP PI/20
300 IF X=Xmin THEN MOVE X,SIN(X)/X ! MOVE to start when X=Xmin
310 IF X=0 THEN Out ! Avoid dividing by 0
320 DRAW X,SIN(X)/X ! DRAW to next point on curve
330 MOVE X,0 ! MOVE to base line
340 DRAW X,SIN(X)/X ! DRAW back to curve for fill
350 Out: NEXT X ! Repeat
360 !
370 ! ***** MOVE and LABEL plot
380 MOVE -3.5*PI,1.1 ! MOVE to start of LABEL
390 LOG 1 ! Specify LABEL origin
400 CSIZE 8 ! Specify Character SIZE
410 LABEL USING "K";"SIN (X)/X" ! LABEL plot
420 CSIZE 3 ! Reset Character SIZE for DIGITIZE
430 POINTER -2*PI,.5,0 ! Place CURSOR at 0,0 as a small cross
440 !
450 ! ***** This allows you to DIGITIZE and PRINT points from the CRT
460 Loop: !
470 DIGITIZE X,Y ! Wait for CONTINUE to DIGITIZE X,Y
480 ! & PRINT X and Y on Thermal Printer
490 PRINT USING "2(K,XMD.2D),D/";"X=",X/PI," PI, Y=",Y
500 GOTO Loop ! Repeat this loop again
510 END
520 !
530 !
540 ! ***** LABEL axes routines
550 Lxaxes: !
560 CSIZE 3 ! Specify Character SIZE, ratio and slant
570 LDIR -(PI/2) ! Specify LABEL direction (down)
580 LOG 2 ! Specify LABEL origin
590 FOR Xposition=Xmin TO Xmax STEP PI
600 MOVE Xposition,Ymin ! MOVE to LABEL position
610 LABEL USING "M4DX,K";Xposition/PI,"PI" ! LABEL text
620 NEXT Xposition ! Repeat until complete
630 RETURN
640 !
650 Lyaxes: !
660 CSIZE 3 ! Specify Character SIZE, ratio and slant
670 LDIR 0 ! Specify LABEL direction (left to right)
680 LOG 8 ! Specify LABEL origin
690 FOR Yposition=Ymin TO Ymax STEP .5
700 MOVE Xmin,Yposition ! MOVE to LABEL position
710 LABEL USING "MD.DDX";Yposition ! LABEL text
720 NEXT Yposition ! Repeat until complete
730 RETURN

```

Football Field Example

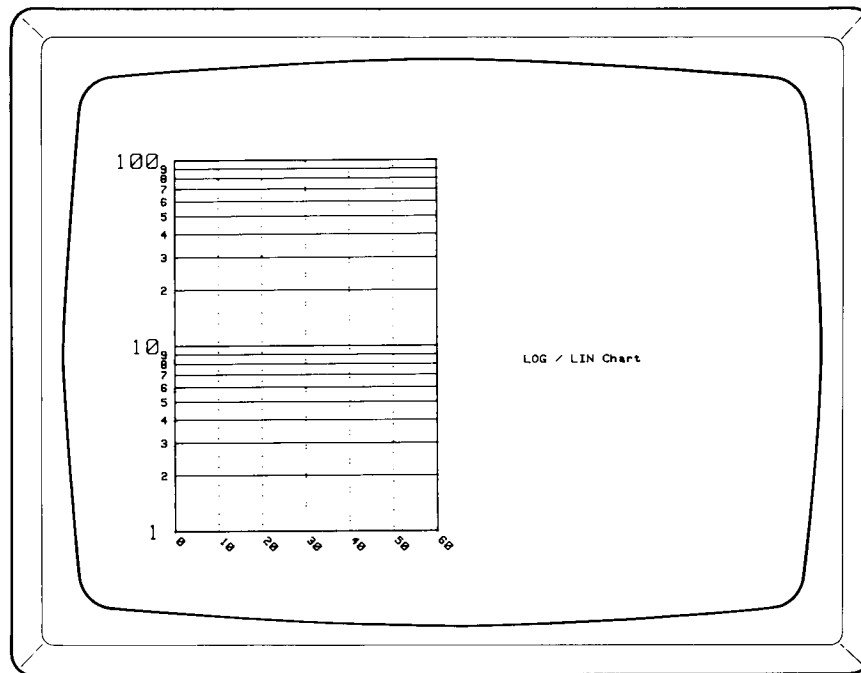


```

1  REM ** This is the Football example (FUTBAL)
10 ! This program draws a football field to demonstrate some interesting
20 ! capabilities and programming techniques using the Graphics statements
30 !
40 PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
50 GRAPHICS ! Enables the CRT
60 FRAME ! FRAMES the SOFT CLIP area
70 LOCATE 10,112,10,90 ! Defines the area to be SCALED
80 SHOW -60,60,-80/3,80/3 ! Sets overall size
90 DEG ! Sets the degrees mode
100 !
110 ! ***** Draw Yard Lines
120 CLIP -40,40,-80/3,80/3 ! CLIP length of Yard Lines
130 LINE TYPE 3 ! Specify dashed line type
140 GRID 10,70,-40,-30 ! Draw Yard Lines
150 !
160 ! ***** Draw Field Perimeter
170 CLIP -50,50,-80/3,80/3 ! CLIP at Field Perimeter
180 LINE TYPE 1 ! Specify solid line
190 FRAME ! Draw Perimeter
200 !
210 ! ***** Draw End Zones
220 CLIP -60,60,-80/3,80/3 ! CLIP at End Zones
230 FRAME ! Draw Perimeter at End Zones
240 !
250 ! ***** Draw Goalposts
260 CSIZE 3 ! Specify Character SIZE ,ratio and slant
270 LORG 5 ! Center LABELs
280 LDIR 0 ! LABEL left to right
290 FOR Goal=-1 TO 1 STEP 2 ! Specify one of two goals
300   FOR Post=-1 TO 1 STEP 2 ! Specify one of two posts
310     MOVE Goal*60,Post*6 ! MOVE to goalpost position
320     LABEL USING "K";"0" ! Draw goalpost
330   NEXT Post ! Repeat for other post
340 NEXT Goal ! Repeat for other goal
350 !
360 ! ***** LABEL Yard Lines
370 CSIZE 4 ! Specify Character SIZE
380 LORG 5 ! Center LABELs
390 FOR Side=-1 TO 1 STEP 2 ! Specify one of two field sides
400   FOR Yard=-40 TO 40 STEP 10 ! Specify Yard line
410     LDIR 90+90*SGN(Side) ! Invert LABEL depending on the side
420     MOVE Yard,Side*25 ! MOVE to Yard and Side
430     LABEL USING "DD";50-ABS(Yard) ! LABEL Yard line
440   NEXT Yard ! Repeat for NEXT Yard line
450 NEXT Side ! Repeat for other side
460 !
470 ! ***** LABEL End Zones
480 CSIZE 8 ! Specify Character SIZE,ratio and slant
490 LORG 5 ! Center LABELs
500 LDIR 90 ! LABEL bottom to top
510 MOVE -55,0 ! MOVE to left End Zone
520 LABEL USING "K";"HOME" ! LABEL End Zone
530 LDIR -90 ! LABEL top to bottom
540 MOVE 55,0 ! MOVE to right End Zone
550 LABEL USING "K";"VISITORS" ! LABEL End Zone
560 !
570 ! ***** LABEL Score
580 CSIZE 5 ! Specify Character SIZE,ratio and slant
590 LORG 5 ! Center LABELs
600 LDIR 0 ! LABEL left to right
610 MOVE 0,-35 ! MOVE to bottom of plotting area
620 LABEL USING "K";"HOME 21, VISITORS 14" ! LABEL Score
630 END

```

LOG/LIN Chart Example



```

1    REM ** This example draws a LOG/LIN Chart (LOGLIN)
10   Set_up:  !
20   DEG      ! Sets DEgrees mode
30   Ymin=0    ! Y axis minimum value
40   Ymax=100  ! Y axis maximum value
50   Log_interval=LGT(Ymax-Ymin) ! Number of decades along Y axis
60   Xmin=0    ! X axis minimum value
70   Xmax=60   ! X axis maximum value
80   PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
90   GRAPHICS  ! Enables the CRT
100  LOCATE 15,75,10,95      ! LOCATES SCALE area from 15,10 to 75,95
110  SCALE Xmin,Xmax,Ymin,Log_interval ! SCALES the LOCATE area to 60x2
120  LINE TYPE 3             ! Selects dashed LINE TYPE
130  AXES 10,0,0,0,1,1,170  ! Draws AXES with TICK marks on X axis
140  LINE TYPE 1             ! Selects solid LINE TYPE
150  FRAME                  ! FRAMEs the SOFT CLIP area
160  Logrithmic:  !
170  FOR Decade=0 TO Log_interval ! Loop for each Decade
180  FOR Interval=1 TO 9          ! Loop for interval TICKs
●190  MOVE 0,Decade+LGT(Interval) ! MOVES PEN to X=0, Y=Interval
●200  DRAW 100,Decade+LGT(Interval) ! DRAWS interval TICK
210  NEXT Interval              ! Repeats FOR NEXT Interval
220  MOVE 0,Decade             ! MOVES PEN to X=0, Y=Decade
230  DRAW 100,Decade           ! DRAWS Logrithmic TICK mark
240  NEXT Decade               ! Repeats loop FOR NEXT Decade

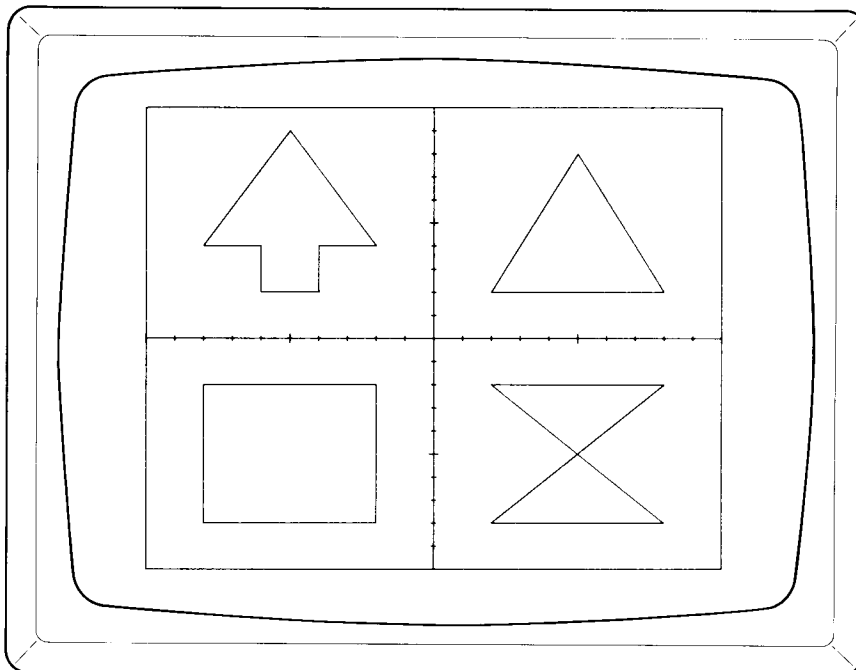
```

```

250 Log_labels: !
260   LORG 8      ! Label Origin 8
270   DATA 1,10,100 ! Data for labels
280   FOR Decade=0 TO Log_interval ! Loop for each Decade
290     CSIZE 3    ! Sets CSIZE to 3
300     FOR Interval=2 TO 9 ! Loop for interval TICKs
310       MOVE -.05,Decade+LGT(Interval) ! MOVES PEN to X=.0, Y=Interval
320       LABEL Interval ! LABELS Interval TICK
330     NEXT Interval ! Repeats FOR NEXT Interval
340     CSIZE 6    ! Sets CSIZE to 6
350     READ Log_$ ! READs Log Decade
360     MOVE -4,Decade ! MOVES PEN
370     LABEL Log_$ ! LABELs Y axis value
380   NEXT Decade ! Repeats FOR NEXT Y_axis label
390 Lin_labels: !
400   CSIZE 3    ! Sets CSIZE to 3
410   LORG 3     ! Label Origin 3
420   LDIR -45   ! Label DIRECTION at -45 deg
430   FOR X_axis=0 TO Xmax STEP 10 ! Loop for TICK marks
440     MOVE X_axis,0 ! MOVES PEN
450     LABEL X_axis ! LABELs X axis value
460   NEXT X_axis ! Repeats FOR NEXT X_axis label
470 Text: !
480   LDIR 0     ! Label DIRECTION at 0 deg
490   SETGU      ! Sets GDU's mode
500   MOVE 95,50 ! MOVES PEN to X=95,Y=0
510   LABEL "LOG / LIN Chart" ! LABELs title
520   END

```

PLOT Example



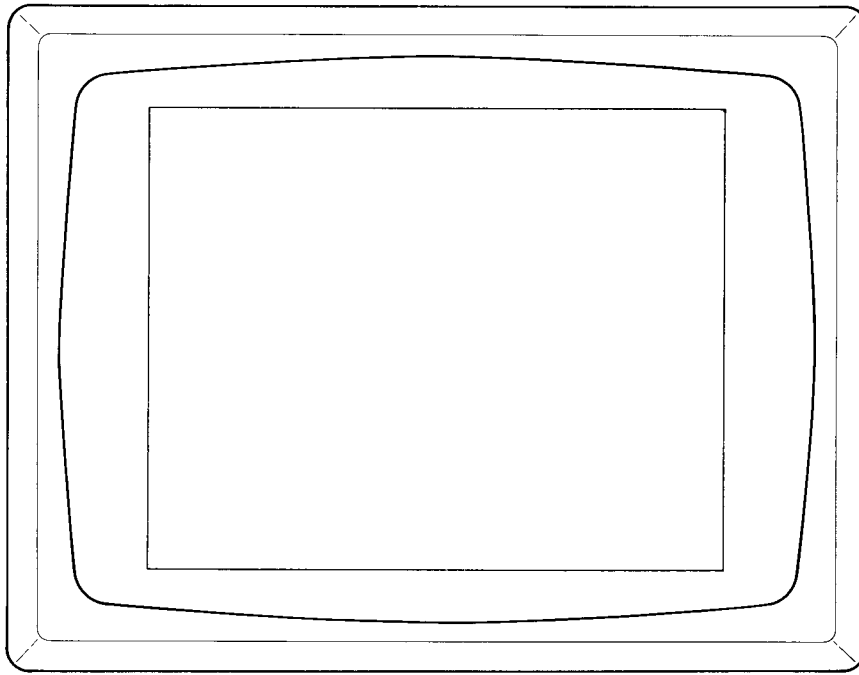
PLOT Example continued

```

1    REM ** This is the PLOT example (PLOT)
10   PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT as the plotter
20   GRAPHICS                      ! Enables the CRT
30   ON ERROR GOTO Oops            ! Specify a branch ON an ERROR
40   Set_scale: !
50   Xmin=Ymin=-10                 ! Sets lower left limits
60   Xmax=Ymax=10                  ! Sets upper right limits
70   SCALE Xmin,Xmax,Ymin,Ymax     ! SCALES the LOCATE area to 20x20
80   Axes_and_frame:!
90   AXES 1,1,0,0,5,5             ! Draws AXES
100  FRAME                        ! FRAMES SOFT CLIP area
110  Loop:!
120  READ Xvalue,Yvalue,Pen_control ! READ DATA for figure
130  PLOT Xvalue,Yvalue,Pen_control ! PLOT DATA
140  GOTO Loop                     ! Repeats until OUT OF DATA
150  Finish:!
160  END
170  Oops:!
180  IF ERR%#36 THEN Finish        ! On OUT OF DATA GOTO Finish
190  BEEP                          ! Sing out
200  EXIT GRAPHICS                ! Turn off the CRT
210  DISP ERR%#                  ! Display error message
220  GOTO Finish                  ! Terminate program
230  Figures:!
240  First_figure:!                a Triangle
250      DATA 2,2,1
260      DATA 8,2,-1
270      DATA 5,8,-1
280      DATA 2,2,2
290  Second_figure:!              an Arrow
300      DATA -4,2,-2
310      DATA -4,4,-1
320      DATA -2,4,-1
330      DATA -5,9,-1
340      DATA -8,4,-1
350      DATA -6,4,-1
360      DATA -6,2,-1
370      DATA -4,2,2
380  Third_figure:!               a Rectangle
390      DATA -2,-2,1
400      DATA -8,-2,-1
410      DATA -8,-8,-1
420      DATA -2,-8,-1
430      DATA -2,-2,2
440  Fourth_figure:!             an Hourglass
450      DATA 2,-2,1
460      DATA 8,-2,-1
470      DATA 2,-8,-1
480      DATA 8,-8,-1
490      DATA 2,-2,2

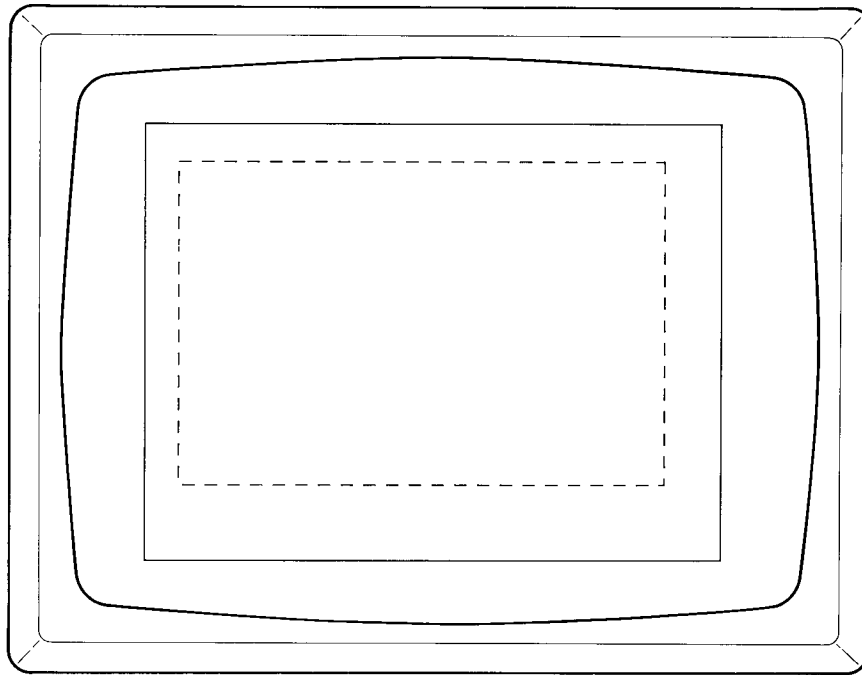
```

FRAME Example



```
1  REM  ** This is the FRAME example (FRAME)
10  PLOTTER IS 13,"GRAPHICS"  ! Specifies the CRT as the plotter
20  GRAPHICS                  ! Enables the CRT
●30  FRAME                    ! Draws a FRAME around the SOFT CLIP area
40  END
```

LIMIT Example

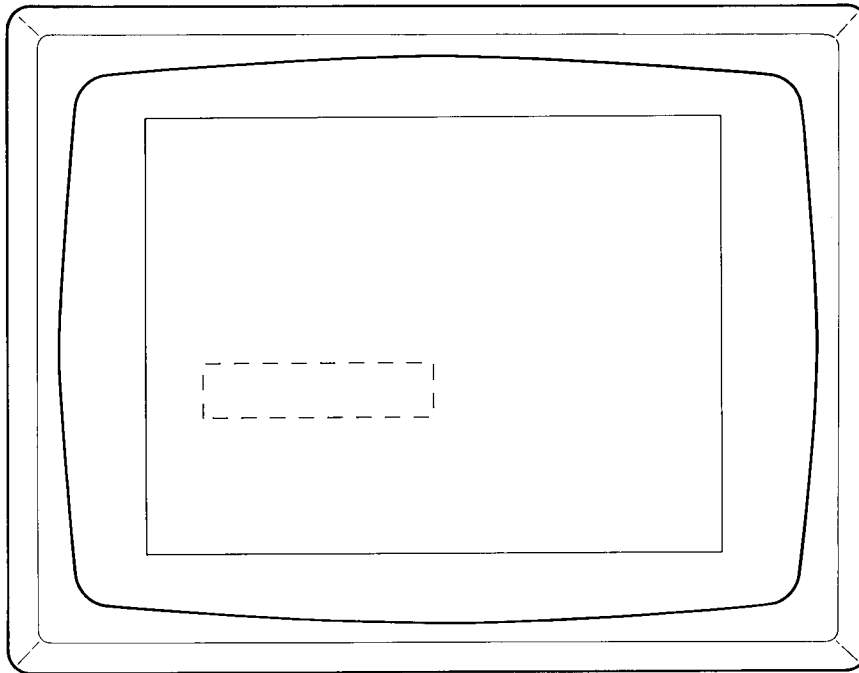


```

1  REM  ** This is the LIMIT example (LIMIT)
2  REM  ** Note that lines or LABELs can not be drawn outside the HARD
3  REM  ** CLIP limits, unless the LIMIT statement is changed by the
4  REM  ** execution of another LIMIT statement
5  REM  ****
10  PLOTTER IS 13,"GRAPHICS"  ! Specifies the CRT as the plotter
20  GRAPHICS                 ! Enables the CRT
30  FRAME                    ! Draws a FRAME around the SOFT CLIP area
●40  LIMIT 15,15+150,10,10+100 ! Specifies a 150mm x 100mm plotting area
41                               ! that is offset from the plotter's
42                               ! lower left physical bounds. The specified
43                               ! offset in this example is 10mm up and
44                               ! 15mm to the right of that point.
45                               ! REMEMBER, OFFSET UNITS are in MILLIMETRES!
50  LINE TYPE 4              ! Specifies a dashed line for new SOFT CLIP area
60  FRAME                    ! Draws a FRAME around the SOFT CLIP area
70  END

```

CLIP Example

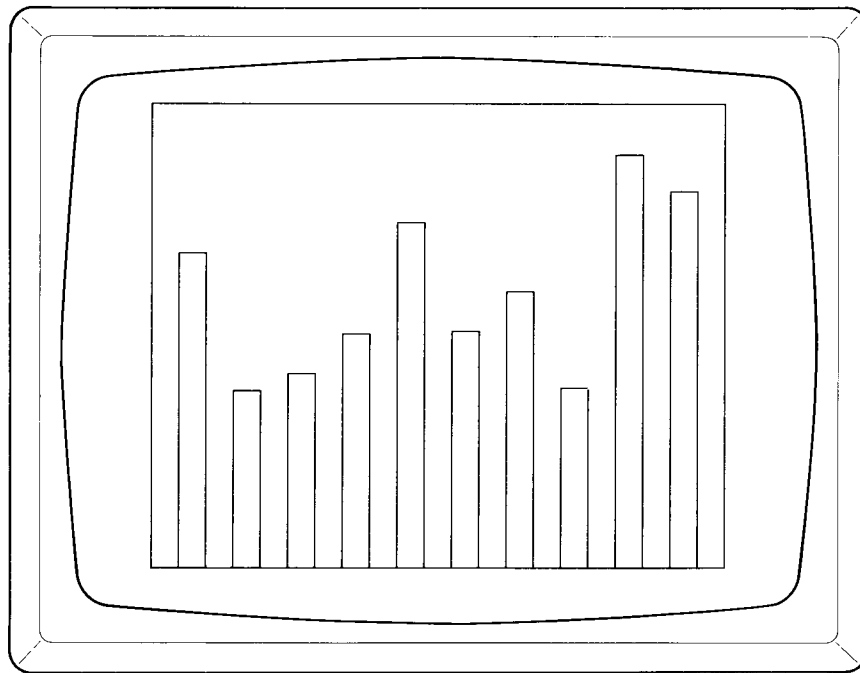


```

1  REM ** This is the CLIP example (CLIP)
10 PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20 GRAPHICS                 ! Enables the CRT
30 FRAME                    ! Draws a FRAME around the SOFT CLIP area
40 SCALE 0,100,0,80         ! SCALES the LOCATE area to 100x80
●50 CLIP 10,50,25,35        ! Specifies SOFT CLIP area
60 LINE TYPE 4              ! Specifies a dashed line for new SOFT CLIP area
70 FRAME                    ! Draws a FRAME around the SOFT CLIP area
80 END

```

Histogram Using CLIP Statement

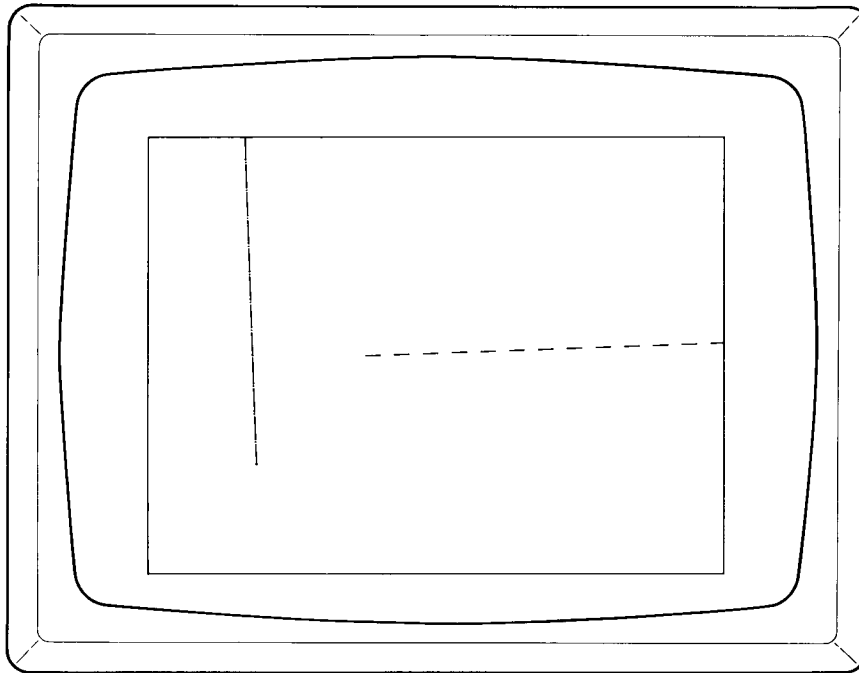


```

1  REM ** This example (BAR) shows how the CLIP statement
2  REM ** can draw a simple histogram.
10 PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT as the plotter
20 GRAPHICS                     ! Enables the CRT
30 SCALE 0,21,0,100             ! SCALES the default LOCATE area to 21x100
40 FRAME                       ! FRAMES the SOFT CLIP area
50 FOR Bar=1 TO 10              ! Specifies one of 10 Bars
60   Bar_length=RND*100          ! Generates random data for example
70   CLIP Bar*2-1,Bar*2,0,Bar_length ! Sets SOFT CLIP area equal to data
80   FRAME                      ! FRAMES the SOFT CLIP area
90 NEXT Bar                     ! Repeats for NEXT Bar
100 END

```

Plotting Outside the Hard Clip Area

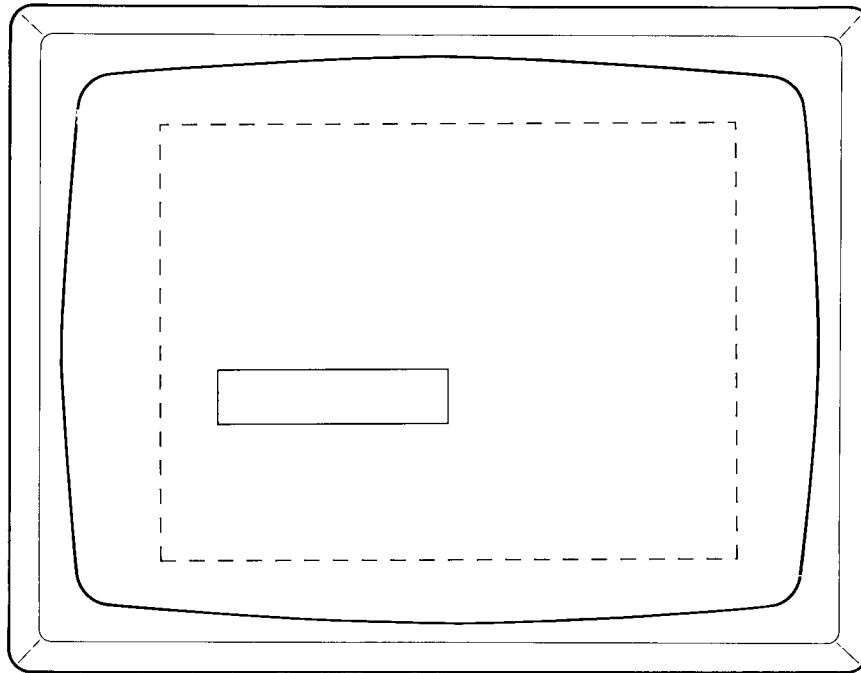


```

1  REM ** This example demonstrates plotting
2  REM ** outside of the physical plotting
3  REM ** limits of the device. (LOST)
10  PLOTTER IS "GRAPHICS"      ! Specifies the CRT as the plotter
20  GRAPHICS                  ! Enables the CRT
30  LIMIT 0,184,0,149         ! Sets LIMITs at physical maximum for the CRT
40  FRAME                     ! Draws border around the SOFT CLIP area
50  CLIP 0,1000,0,1000        ! Sets SOFT CLIP area greater than
51                               ! HARD CLIP area
60  LINE TYPE 4               ! Changes the LINE TYPE
70  MOVE 50,50                ! MOVES near the center of the CRT
● 80  DRAW 750,75              ! DRAWS line beginning inside and ending
81                               ! outside the HARD CLIP area
90  LINE TYPE 5               ! Changes LINE TYPE
● 100 DRAW 0,700               ! DRAWS a line beginning and ending
101                               ! outside the HARD CLIP area
110 LINE TYPE 6               ! Changes LINE TYPE
● 120 DRAW 25,25              ! DRAWS a line beginning outside and
121                               ! ending inside the HARD CLIP area
130  END

```

UNCLIP Example

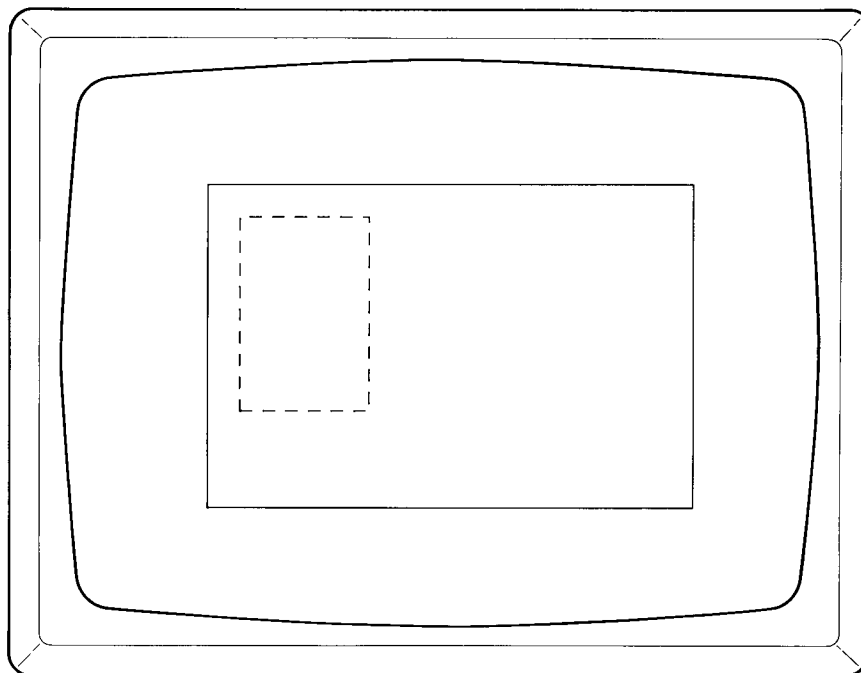


```

1  REM ** This is the UNCLIP example (UNCLIP)
10 PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20 GRAPHICS ! Enables the CRT
30 SCALE 0,100,0,80 ! SCALES the LOCATE rectangle to 100x80
40 CLIP 10,50,25,35 ! Specifies a new SOFT CLIP area
50 FRAME ! Draws a FRAME around the SOFT CLIP area
●60 UNCLIP ! Changes the SOFT CLIP area back to default
70 LINE TYPE 4 ! Specifies a dashed line for new SOFT CLIP area
80 FRAME ! Draws a FRAME around the SOFT CLIP area
90 END

```

LOCATE Example

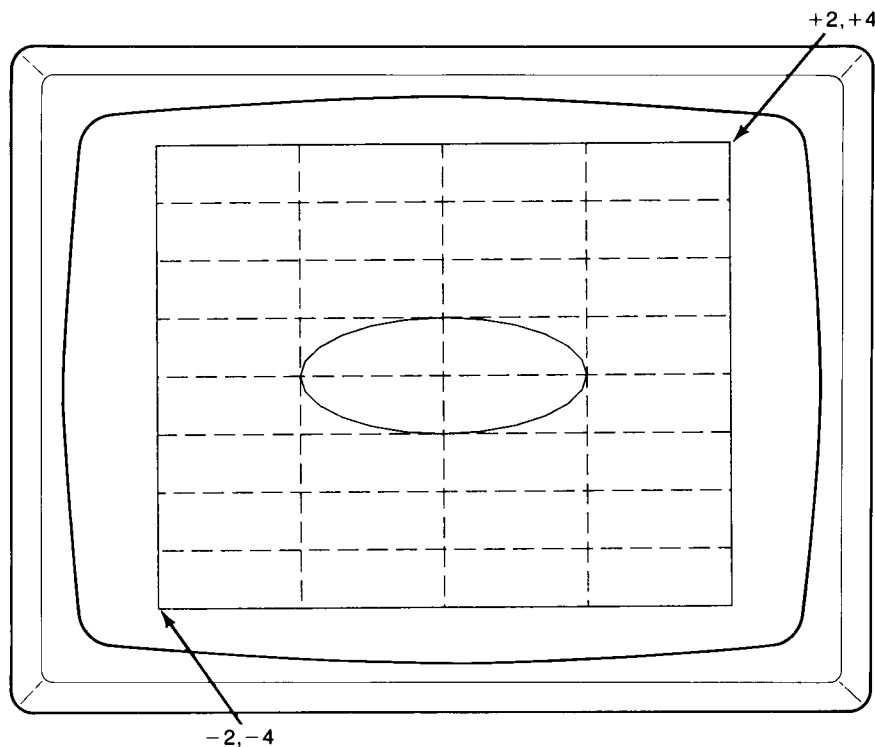


```

1  REM ** This is the LOCATE example (LOCATE)
2  REM ** Note while in the UDU's mode, plotted lines can not be drawn
3  REM ** outside the "LOCATE" boundaries. LABELS can be positioned
4  REM ** anywhere within the "LIMIT" limits.
5  REM *****
10 PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20 GRAPHICS                ! Enables the CRT
30 LIMIT 15,15+150,10,10+100 ! Specifies the plotting area
40 FRAME                   ! Draws a FRAME around the SOFT CLIP area
●50 LOCATE 10,50,30,90      ! "LOCATE" a rectangular area
51                          ! You can assign units of measure to this area
52                          ! The parameters are in GDU's ; they are a
53                          ! percentage of X/Y set in the LIMIT statement
60 LINE TYPE 4             ! Specifies a dashed line
70 FRAME                   ! Draws a FRAME around the SOFT CLIP area
80 END

```

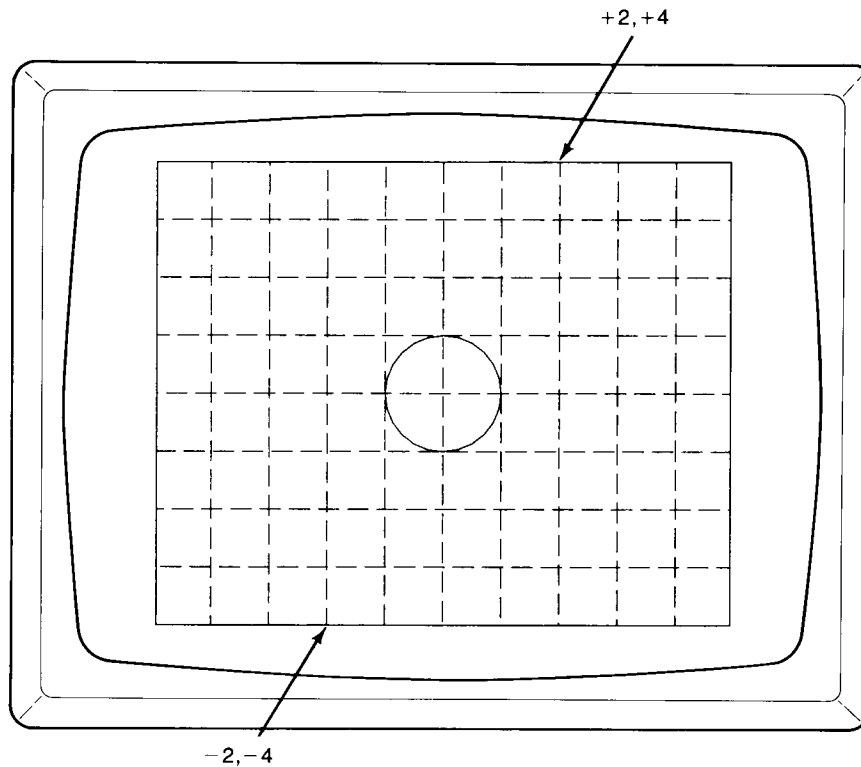
SHOW and SCALE Examples



```

1  REM ** This is another SCALE example (SCALE2)
2  REM ** Note that a unit-of-measure along the X axis does not
3  REM ** equal a unit-of-measure along the Y axis
4  REM *****
10  PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT as the plotter
20  GRAPHICS                     ! Enables the CRT
30  DEG                          ! Sets degrees mode
●40  SCALE -2,2,-4,4              ! SCALES the LOCATE rectangle to 6x6
50  LINE TYPE 3                  ! Selects a dashed LINE TYPE
60  GRID 1,1                     ! Draws a GRID for the PLOT
70  LINE TYPE 1                  ! Specifies a solid LINE TYPE
80  FRAME                        ! Draws a FRAME around the SOFT CLIP area
81  REM ** The next section attempts to draw a circle
82  REM ** Note that it is not round, due to the SCALE statement
83  REM *****
90  MOVE 0,1                     ! MOVE to the beginning of the circle
100  FOR Angle=0 TO 360           ! FOR...NEXT loop to specify an Angle
110  DRAW SIN(Angle),COS(Angle)   ! Draw the circle
120  NEXT Angle                   ! Repeat the loop
130  END

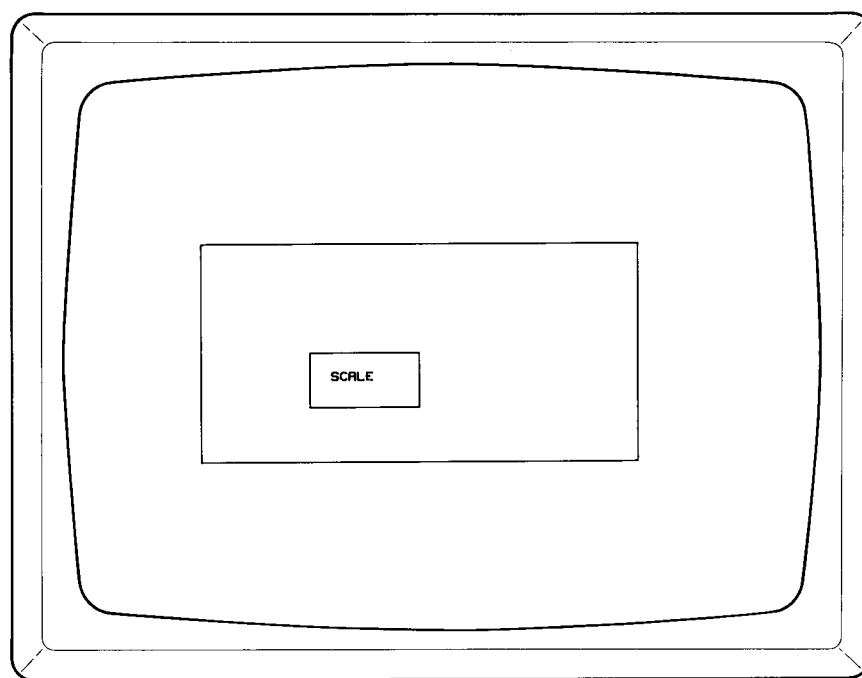
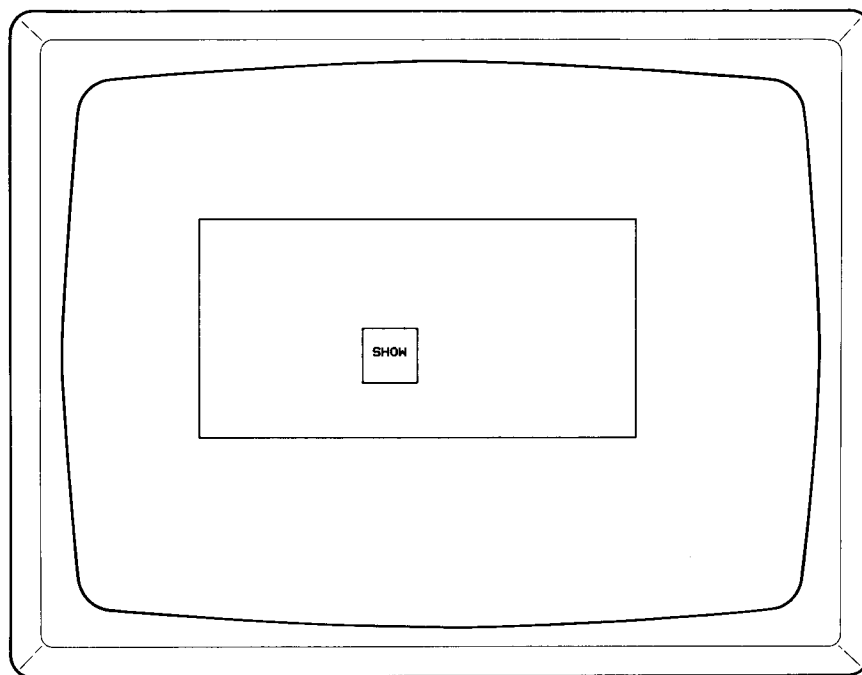
```



```

1  REM ** This is the SHOW example (SHOW)
10  ! This program demonstrates the effects of the SHOW statement on the
20  ! plotting area. Note that the length of a unit-of-measure in the X
30  ! and the Y direction are equal.
40  !
50  PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT as the plotter
60  GRAPHICS                     ! Enables the CRT
70  DEG                          ! Sets degrees mode
80  SHOW -2,2,-4,4               ! SHOWs an equal area along both X and Y axes
90  LINE TYPE 3                  ! Specifies the LINE TYPE for GRID
100 GRID 1,1                     ! Draws a GRID for the plot
110 LINE TYPE 1                  ! SPECIFIES a solid line for FRAME and circle
120 FRAME                        ! FRAME the plotting area
130 !
140 ! ***** DRAW A CIRCLE FOR THIS EXAMPLE
150 MOVE 0,1                     ! MOVE to the start of the circle
160 FOR Angle=0 TO 360 STEP 15   ! FOR...NEXT loop to specify an Angle
170 DRAW SIN(Angle),COS(Angle)   ! DRAW circle
180 NEXT Angle                   ! Repeat loop
190 END

```

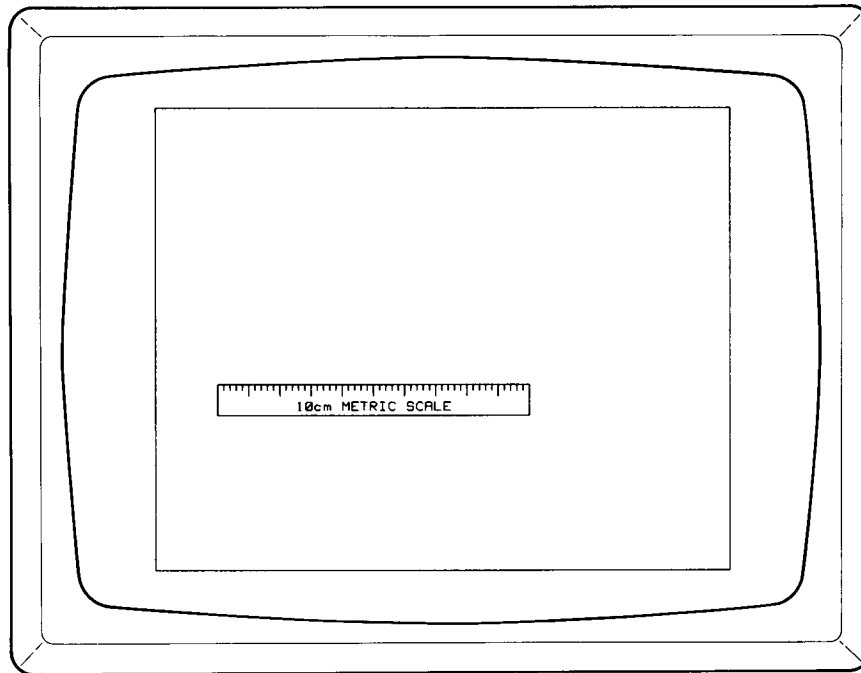


```

1  REM ** This example shows the SHOW statement (SHWSCL)
2  REM ** making the SOFT CLIP area display a square figure
10 PLOTTER IS "GRAPHICS"      ! Specifies the CRT as the plotter
20 GRAPHICS                  ! Enables the CRT
30 LIMIT 0,180,0,140        ! Sets HARD CLIP area
40 LOCATE 0,100,0,50         ! Sets scaled area so (2xY) = X
50 FOR Loop=1 TO 2          ! Starts loop for figure
60 FRAME                     ! FRAMES the SOFT CLIP area
●70 IF Loop=1 THEN SCALE 0,4,0,4 ! SCALES LOCATE area to 4x4
●80 IF Loop=2 THEN SHOW 0,4,0,4 ! SHOWs LOCATE area of 4x4
90 MOVE 1,1                  ! MOVES the PEN
100 IPLOT 0,1,-1             ! This figure is a 4 line box
110 IPLOT 1,0                ! SCALEing the non-square LOCATE area
120 IPLOT 0,-1               ! draws it as a rectangle; while SHOWing
130 IPLOT -1,0               ! draws it as a square
140 MOVE 1.2,1.5             ! MOVES the PEN
150 IF Loop=1 THEN LABEL "SCALE" ! Identifies the SCALE figure
160 IF Loop=2 THEN LABEL "SHOW"  ! Identifies the SHOW figure
170 WAIT 3000                ! Displays figure for 3 seconds
180 GCLEAR                   ! Clears CRT and graphics memory
190 NEXT Loop                 ! Repeats the loop
200 END

```

Metric Scale Example

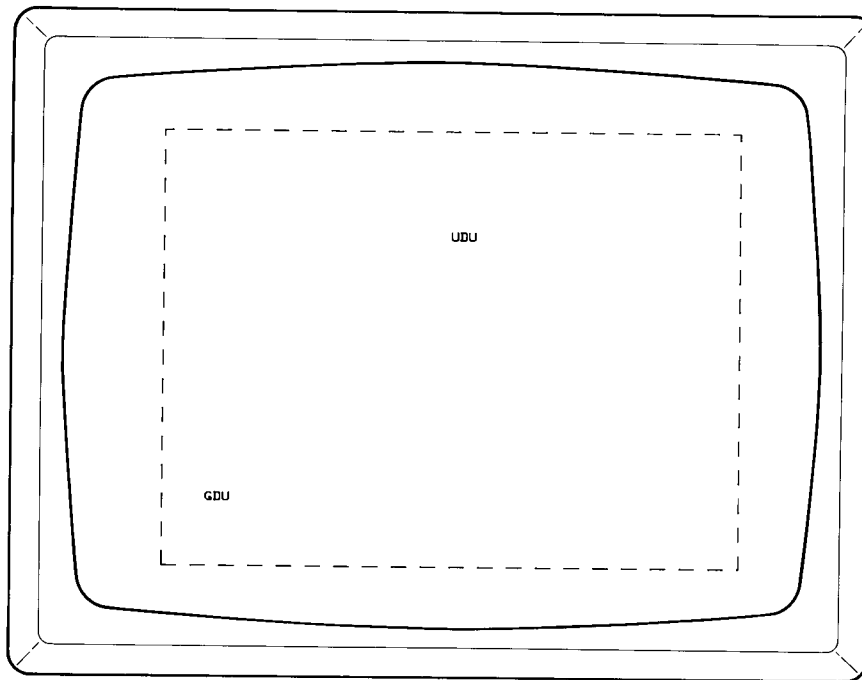


```

1  REM ** This is the MSCALE example (MSCALE)
2  REM ** Note: Delete the DUMP GRAPHICS (line 110) if your
3  REM ** System 45 does not have a thermal printer
4  REM *****
10 PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT as the plotter
20 GRAPHICS                     ! Enables the CRT
30 FRAME                        ! Draws a FRAME around the SOFT CLIP area
● 40 MSCALE 20,50                ! Offset point 0,0 50mm up and 20mm to right
41 REM ** Millimetres are now set as the current unit-of-measure
42 REM *****
50 CLIP 0,100,0,10             ! Sets SOFT CLIP area
60 FRAME                        ! Draws a FRAME around the SOFT CLIP area
70 AXES 2,10,0,10,5,10,5       ! Draws TICK marks along the FRAME
80 MOVE 50,3                   ! MOVES to the center of the SOFT CLIP area
90 LONG 5                       ! Centers the LABEL
100 LABEL "10cm METRIC SCALE"   ! LABELs inside the SOFT CLIP area
110 DUMP GRAPHICS               ! Thermal printer output
120 END

```

GDU and UDU Example

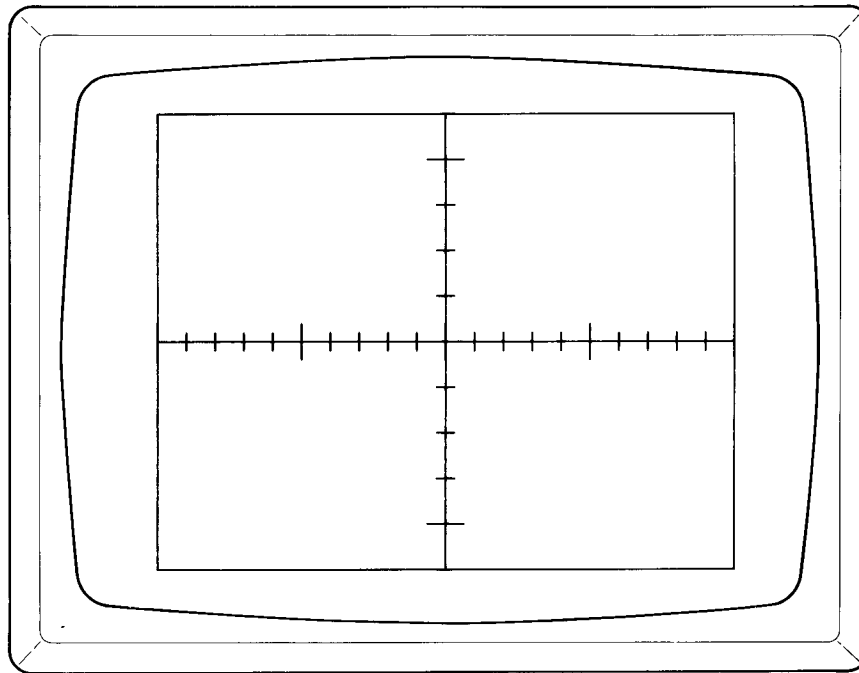


```

10  REM ** This example contrasts GDU's and UDU's (GDUUDU)
20  DATA GDU,UDU          ! Names of modes
30  PLOTTER IS "GRAPHICS"   ! Specifies the CRT as the plotter
40  GRAPHICS                ! Enables the CRT
50  LIMIT 0,184,0,149      ! Defines the HARD CLIP area
60  SCALE 0,20,0,20        ! SCALES LOCATE rectangle to 20x20
70  LINE TYPE 4            ! Selects a dashed LINE TYPE
80  FRAME                  ! Draws a FRAME around the SOFT CLIP area
●90  SETGU                 ! Sets Graphics Display Units (GDU's) mode
100  GOSUB Text            ! LABELing operation
●110 SETUU                ! Sets User Defined Units (UDU's) mode
120  GOSUB Text            ! LABELing operation
130  END
140 Text: !
150  LINE TYPE 1          ! Selects a solid LINE TYPE
160  MOVE 10,15           ! MOVES the PEN to 10,15 in current units
170  READ Text$           ! READs the data for the current mode
180  LABEL Text$          ! LABELs text
190  RETURN

```

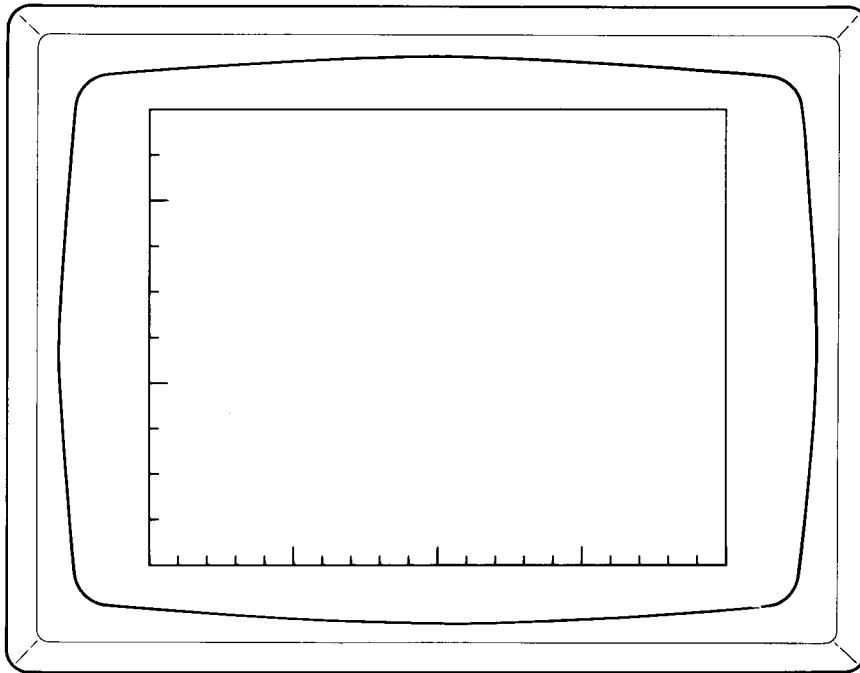
AXES Examples



```

1  REM  ** This example demonstrates the AXES statement (AXES)
10  PLOTTER IS "GRAPHICS"      ! Specifies the CRT as the plotter
20  GRAPHICS                   ! Enables the CRT
30  SCALE -10,10,-10,10       ! SCALES LOCATE rectangle to 20x20
40  AXES 1,2,0,0,5,4,8        ! Draws AXES, specifying:
41                               !   One minor tick per unit in X
42                               !   One minor tick per two units in Y
43                               !   AXES intersection at 0,0
44                               !   One major tick per 5 minor ticks in X
45                               !   One major tick per 4 minor ticks in Y
46                               !   Major tick size is 8 GDU's
50  FRAME                      ! Draws a FRAME around the SOFT CLIP area
60  END

```

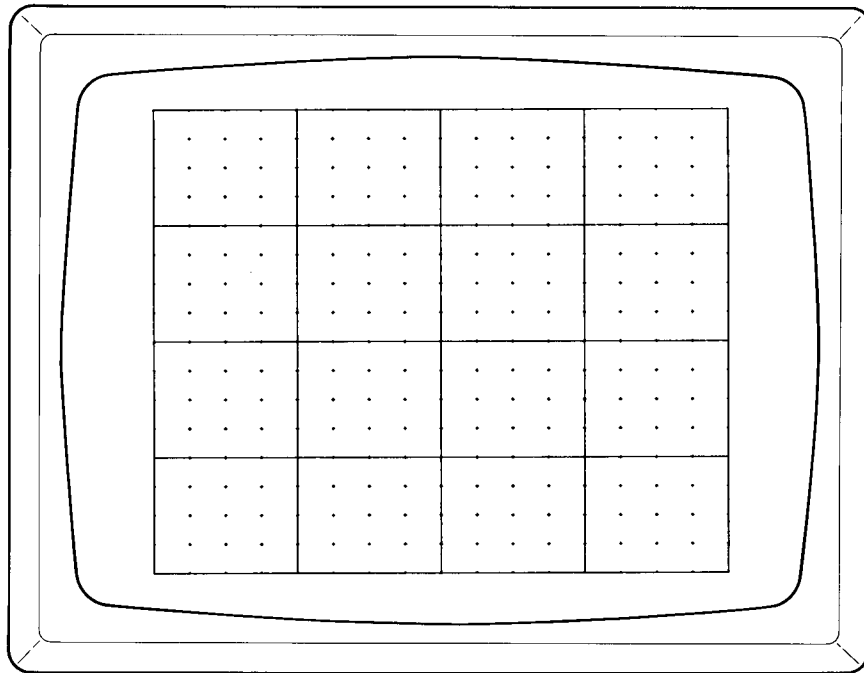


```

1  REM ** This example also demonstrates the AXES statement (AXES2)
10 PLOTTER IS "GRAPHICS"      ! Specifies the CRT as the plotter
20 GRAPHICS                  ! Enables the CRT
30 SCALE -10,10,-10,10       ! SCALES the LOCATE rectangle to 20x20
40 AXES 1,2,-10,-10,5,4,8    ! Draws AXES, specifying:
41                             !   One minor tick per unit in X
42                             !   One minor tick per two units in Y
43                             !   AXES intersection at -10,-10
44                             !   One major tick per 5 minor ticks in X
45                             !   One major tick per 4 minor ticks in Y
46                             !   Major tick size is 8 GDU's
50 FRAME                     ! Draws a FRAME around the SOFT CLIP area
60 END

```

GRID Example

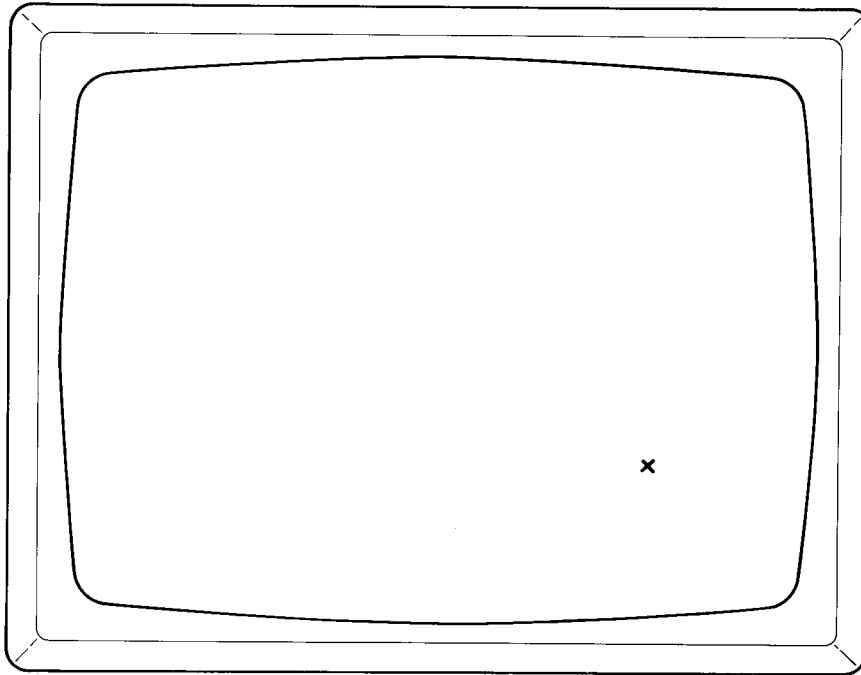


```

1    REM ** This is the GRID example (GRID)
10   PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20   GRAPHICS                 ! Enables the CRT
30   SCALE -10,10,-10,10      ! SCALES the LOCATE rectangle area to 20x20
40   GRID 1,1,0,0,5,5,1      ! DRAWS GRID specifying:
41   ! One minor tick per unit in X
42   ! One minor tick per unit in Y
43   ! Intersect at 0,0
44   ! One major tick per 5 minor ticks in X
45   ! One major tick per 5 minor ticks in Y
46   ! Major tick size is 1 GDU
50   FRAME                   ! Draws a FRAME around the SOFT CLIP area
60   END

```

MOVE Example

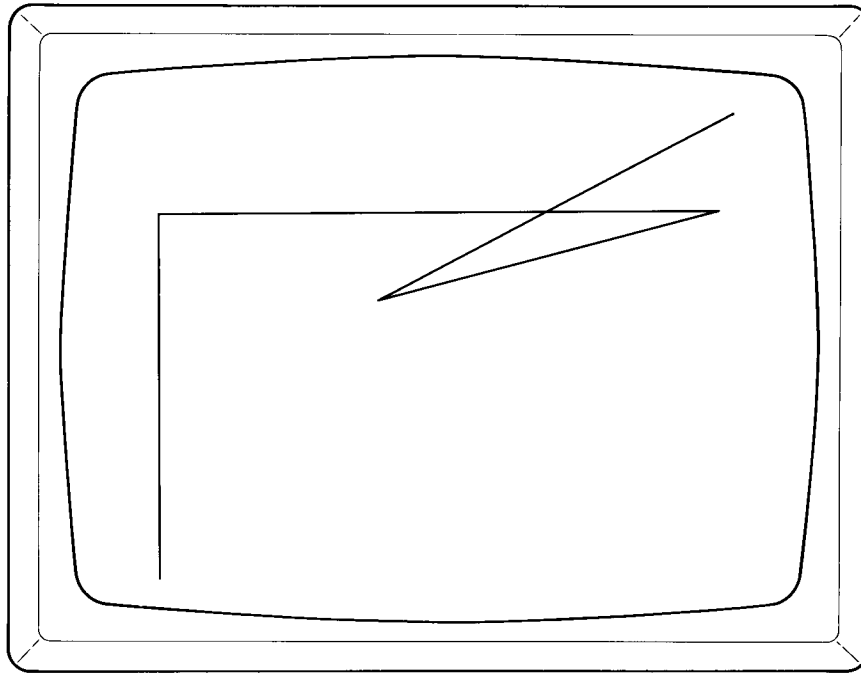


```

1  REM ** This is the MOVE example (MOVE)
10 PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20 GRAPHICS                 ! Enables the CRT
30 FOR Loop=1 TO 25        ! Starts loop
●40 MOVE RND*123,RND*100    ! MOVES to a random coordinate
50 LONG 5                  ! Centers the LABEL
60 LABEL "X"              ! X marks the spot
70 WAIT 750               ! Waits for 3/4 sec
80 GCLEAR                 ! Clears Graphics memory and the CRT
90 NEXT Loop              ! Repeats the loop
100 END

```

DRAW Example

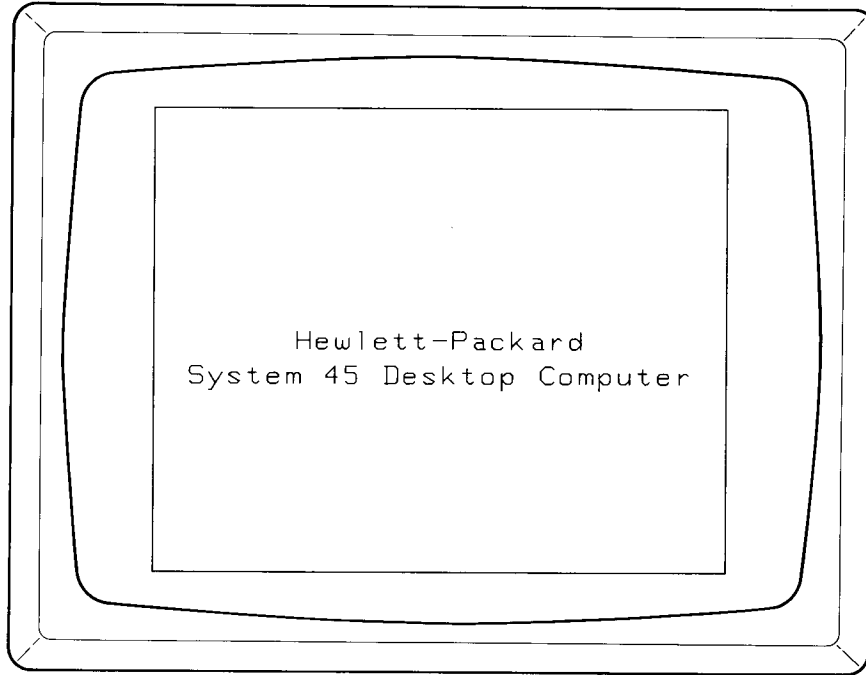


```

1      REM  ** This is the DRAW example (DRAW)
10     PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20     GRAPHICS                ! Enables the CRT
30     MOVE 5,5                ! MOVE to 5,5
●40    DRAW 5,80                ! DRAWS a vertical line to 80 along the Y axis
●50    DRAW 120,80             ! DRAWS a horizontal line to 120 along the X axis
●60    DRAW 50,62              ! DRAWS a diagonal line down to 50,62
●70    DRAW 123,100            ! DRAWS a diagonal line up to 123,100
80     END

```

LABEL Example

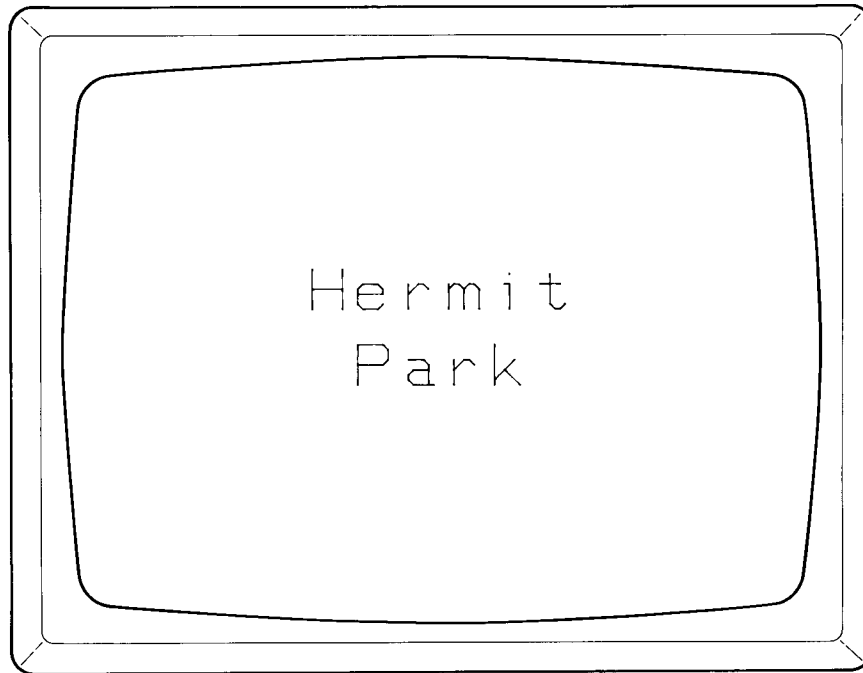


```

1  REM ** This is the LABEL example (LABEL)
10 PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20 GRAPHICS ! Enables the CRT
30 SCALE -1,1,-1,1 ! SCALES the LOCATE rectangle to 2x2
40 FRAME ! FRAMES the SOFT CLIP area
50 CSIZE 7 ! Character SIZE = 7 GDUs
60 LONG 5 ! Centers the LABEL
70 MOVE 0,0 ! MOVES to the center of the plotting area
●80 LABEL "Hewlett-Packard" ! LABELs the text
●90 LABEL "System 45 Desktop Computer"! LABELs the text
100 END

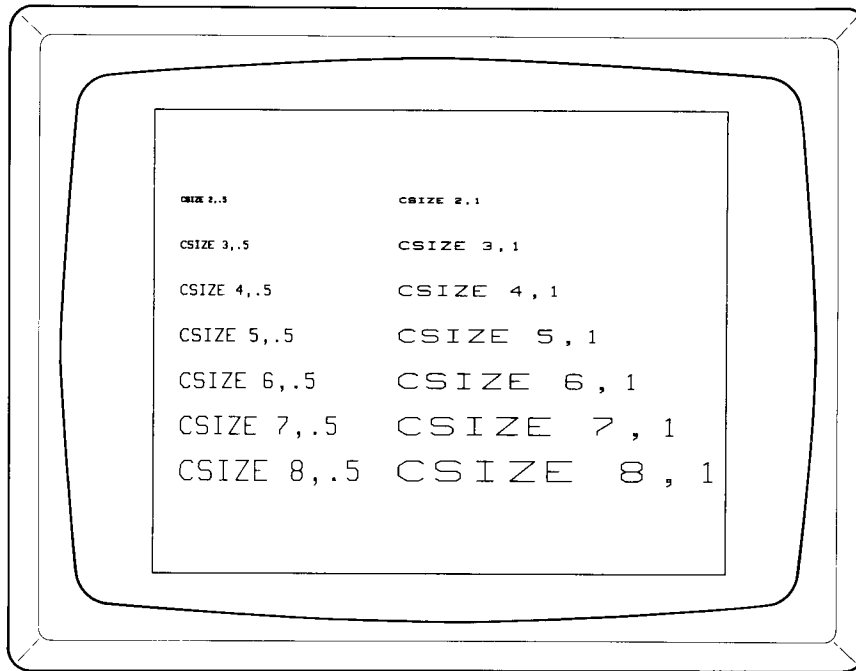
```

LETTER Example



```
1  REM ** This is the LETTER example (LETTER)
10  PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20  GRAPHICS                ! Enables the CRT
30  CSIZE 5,.4              ! Character SIZE = 5 GIU's
31                               ! Aspect ratio = .4
32                               ! Character slant = 0 degrees
●40  LETTER                  ! Sets the LETTER mode
50  END
```

CSIZE Example

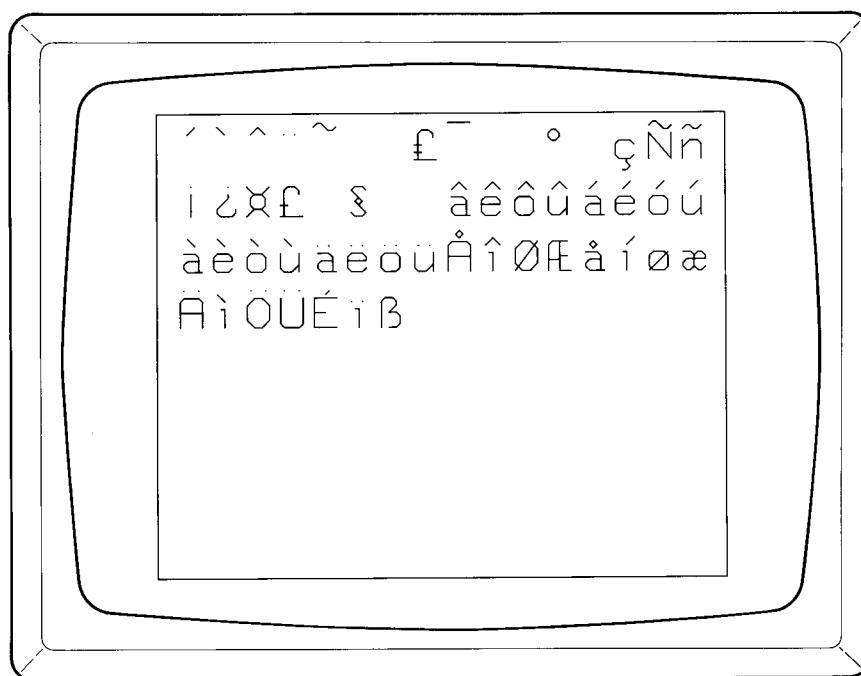
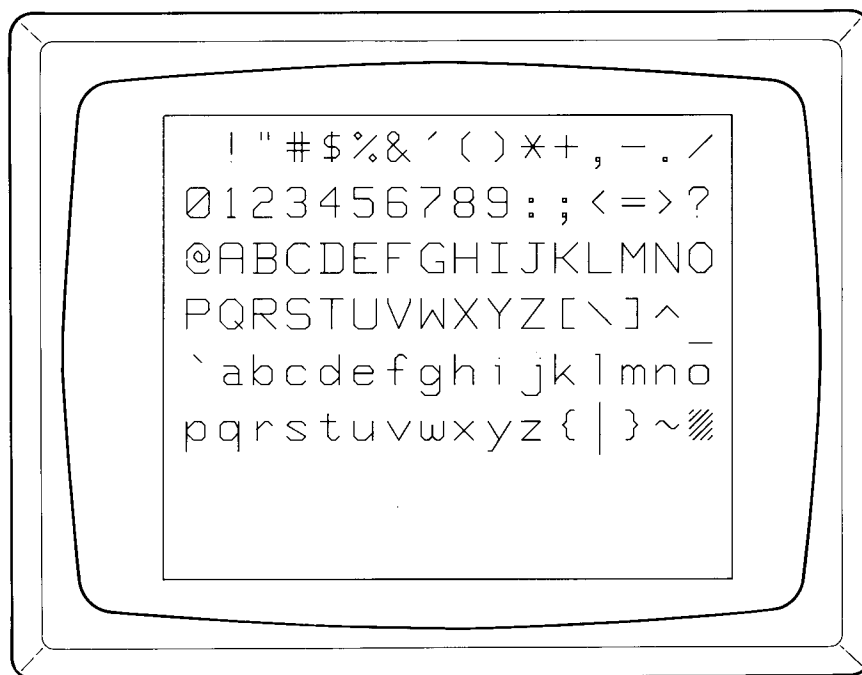


```

1  REM ** This is the CSIZE example (CSIZE)
10  ! This program demonstrates some of the different
20  ! Character SIZES and aspect ratios
30  !
40  PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT as the plotter
50  GRAPHICS                     ! Enables the CRT
60  FRAME                        ! FRAMES the SOFT CLIP area
70  SCALE -.5,10,-10,0          ! SCALES the plotting area to 10.5x10
80  !
90  ! ***** LABEL plotting area with two columns of example Character SIZES
100 FOR Column=1 TO 2           ! Specify one of two columns
110 FOR Size=1 TO 8             ! Specify one of eight Character SIZES
120 CSIZE Size,Column/2         ! Sets Character SIZE and aspect ratio
130 MOVE (Column-1)*4,-Size     ! MOVES PEN to LABEL position
140 LABEL USING "K";"CSIZE ",Size,"",Column/2  !LABELs plotting area
150 NEXT Size                   ! Increments to NEXT Size
160 NEXT Column                 ! Repeats for NEXT Column
170 END

```

Character Set Example

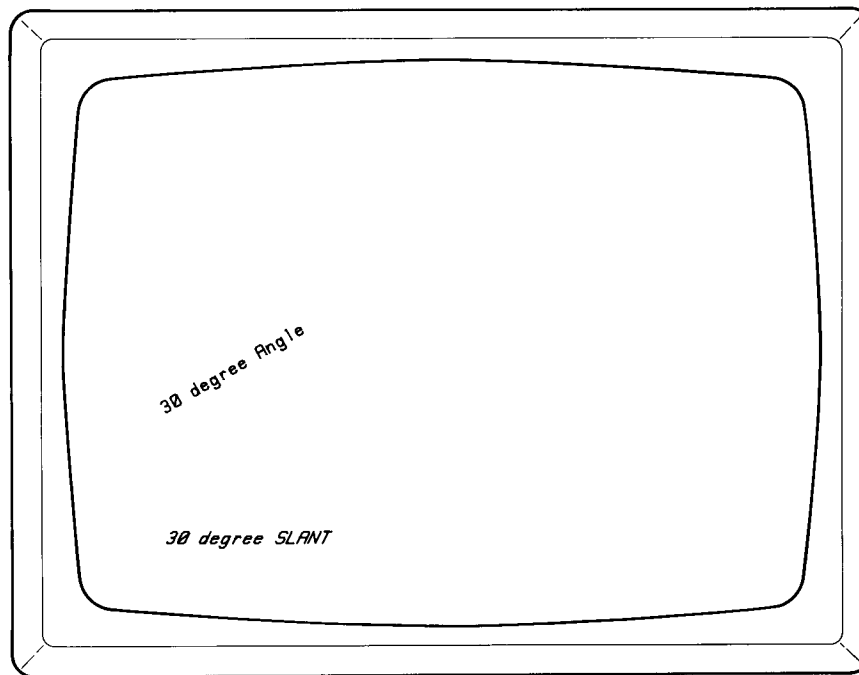


```

1  REM ** This example draws the complete
2  REM ** Graphics character set (CHRSET)
10  ON ERROR GOTO Out          ! Escape for European character set
20  DIM Label$[96]             ! Dimensions string
30  PLOTTER IS 13,"GRAPHICS"    ! Specifies the CRT as the plotter
40  GRAPHICS                   ! Enables the CRT
50  LIMIT 0,184,0,140          ! Defines the CRT LIMITs
60  CSIZE 12                    ! Sets Character SIZE and aspect ratio
70  DATA 32,127,168,222       ! Character sets
80  FOR Complete_set=1 TO 2     ! Initializes the loop
90  FRAME                      ! Draws a FRAME around the SOFT CLIP area
100 MOVE 5,90                  ! MOVES to start position
110 ! ***** Assigns all printable characters to Label$
120 READ Character,Finish       ! READs the characters
130   FOR C=Character TO Finish ! Initializes the loop
140   Character_posit=LEN(Label$) ! Determine next available position
150   Label$[Character_posit+1]=CHR$(C) ! Assigns character to Label$
160   NEXT C                    ! Repeats the loop
170 !
180 ! ***** LABEL characters in groups of 16
190   FOR I=1 TO 96 STEP 16     ! Specify one set of 16 characters
200   LABEL Label$[I,I+15]      ! LABEL 16 characters
210   NEXT I                    ! Repeat until complete
220 Out: !
230 DUMP GRAPHICS              ! Thermal printer output
240 GCLEAR                     ! Clear Graphics memory and CRT
250 Label$=""                  ! Null out Label$
260 NEXT Complete_set           ! Repeats the loop
270 END

```

Slant and Angle Example

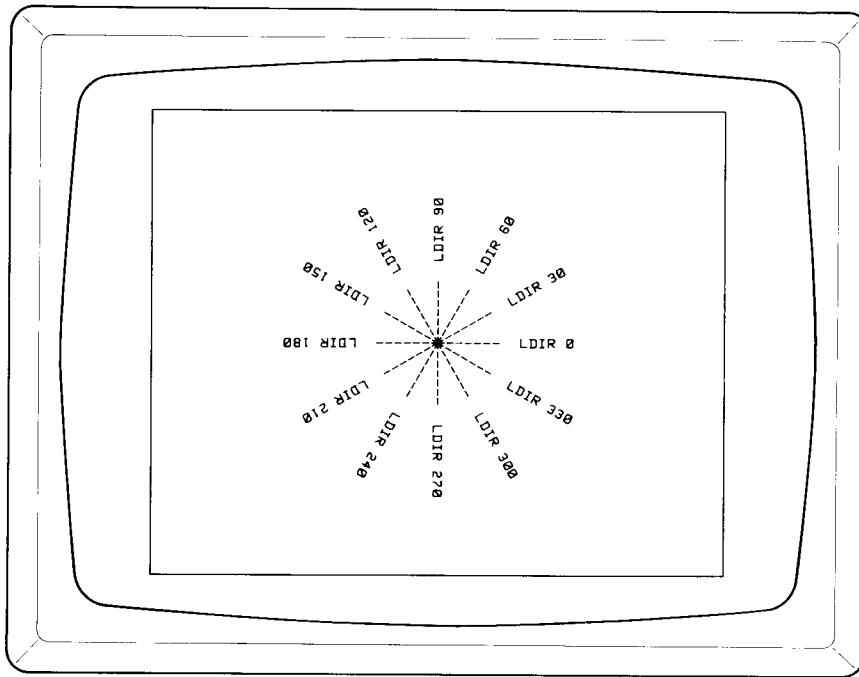


```

1  REM ** This is the SLANT and Angle example (SL_AN)
10 DEG                                ! Sets degrees mode
20 PLOTTER IS 13,"GRAPHICS"           ! Specifies the CRT
30 GRAPHICS                           ! Enables the CRT
40 LIMIT 0,184,0,140                  ! Defines the HARD CLIP area
●50 CSIZE 5,.5,30                     ! Character SIZE =5 GDU's
51                                    ! Aspect ratio =.5
52                                    ! Character SLANT =30 Degrees
60 MOVE 10,20                         ! Positions PEN
70 LABEL "30 degree SLANT"            ! LABELS text
80 MOVE 10,50                         ! Positions PEN
●90 CSIZE 5,.5                        ! Character SIZE =5 GDU's
91                                    ! Aspect ratio =.5
92                                    ! Character SLANT =00 Degrees
●100 LDIR 30                          ! Angle text at 30 degrees
110 LABEL "30 degree Angle"           ! LABELS text
120 END

```

LDIR Example

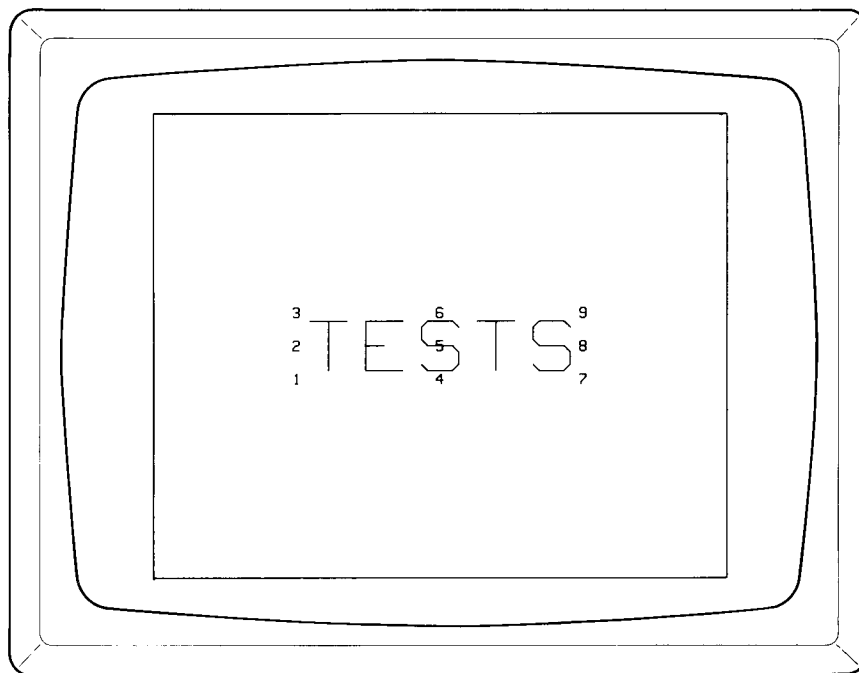


```

10  REM  ** This is the Label DIRECTION example (LDIR)
20  PLOTTER IS "GRAPHICS"      ! Specifies the CRT as the plotter
30  GRAPHICS                   ! Enables the CRT
40  FRAME                      ! FRAMES the SOFT CLIP area
50  DEG                        ! Sets the degrees mode
60  SCALE -10,10,-10,10       ! SCALES the LOCATE rectangle to 20x20
70  LONG 2                     ! Center LABEL at left side
80  FOR Degree=0 TO 330 STEP 30 ! Loop for degrees
● 90  LDIR Degree               ! Specifies a direction
100  MOVE 0,0                  ! MOVES to the center
110  LABEL "----- LDIR";Degree ! LABELs the text
120  NEXT Degree               ! Repeats the loop
130  END

```

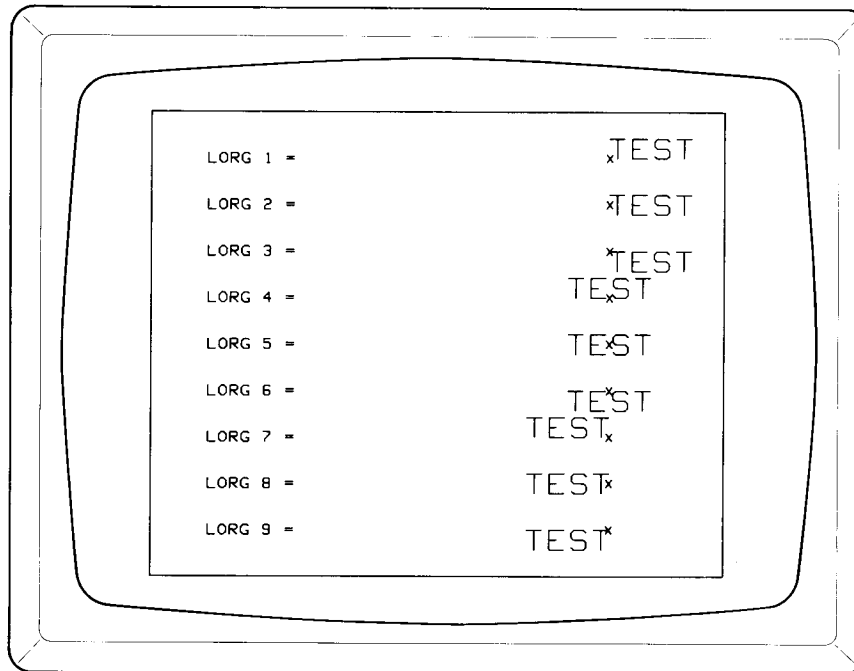
LORG Examples



```

1  REM ** This is another LORG EXAMPLE (LORG1)
10  ! This program labels the word "TESTS" and then labels small
20  ! numbers around the word to show the location of the pen for
30  ! the nine different LORG positions
40  !
50  PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT as the plotter
60  GRAPHICS                      ! Enables the CRT
70  SCALE -2,2,-7,7              ! SCALES the LOCATE rectangle to 4x14
80  FRAME                        ! FRAMES the SOFT CLIP area
90  CSIZE 20                     ! Sets Character SIZE and aspect ratio
●100 LORG 5                      ! Center the LABELs
110 MOVE 0,0                    ! MOVES to the center of the plotting area
120 LABEL USING "K";"TESTS"      ! LABEL the word "TESTS"
130 !
140 ! ***** MOVE and LABEL numbers around the word "TESTS"
150 CSIZE 4                      ! Sets Character SIZE and aspect ratio
160 FOR Position=1 TO 9          ! Specify one of nine positions
170 MOVE INT((Position-1)/3)-1,(Position-1) MOD 3-1 ! Calculate position
180 LABEL USING "D";Position      ! LABEL position number
190 NEXT Position                ! Repeat for NEXT Position
200 END

```

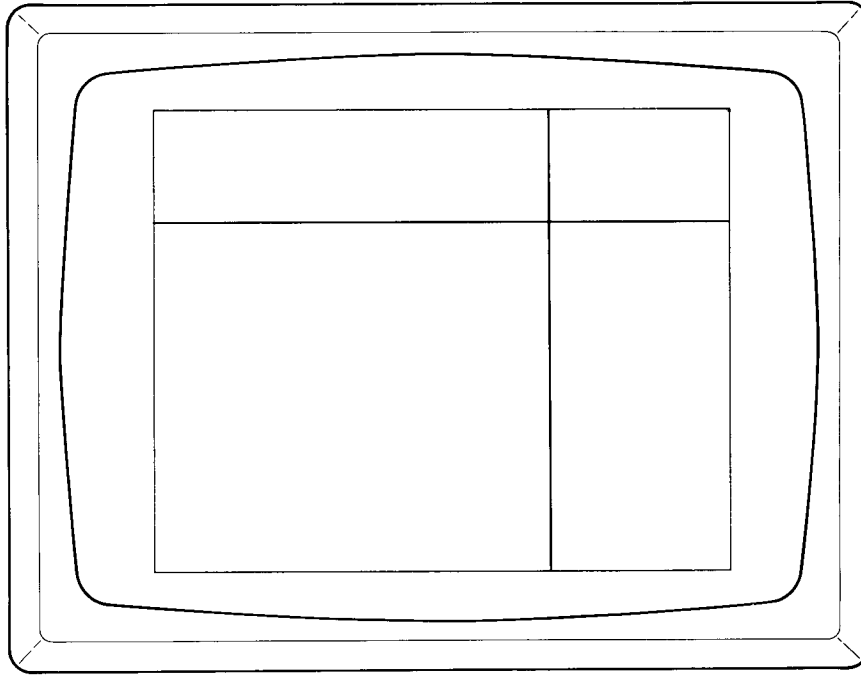


```

1  REM ** This is another LORG example (LORG2)
10  ! This program labels the word "TESTS"
20  ! using each of the LORG parameters; an "X" is drawn
30  ! to indicate the position of the PEN for each LORG
40  PLOTTER IS 13,"GRAPHICS"      ! Specifies the CRT as the plotter
50  GRAPHICS                     ! Enables the CRT
60  SCALE 0,10,-10,0             ! SCALES the LOCATE rectangle to 10x10
70  FRAME                        ! FRAMES the SOFT CLIP area
80  ! ***** LABEL the word "TEST" using each LORG
90  ! ***** the "X" indicates the reference position for the LORG
100 FOR Position=1 TO 9          ! Specify one of nine positions
●110 LORG 2                      ! Specifies LORG for LABELs
120 CSIZE 4                      ! Sets Character SIZE and aspect ratio
130 MOVE 1,-Position            ! MOVE to left LABEL Position
140 LABEL USING "5A,D,2A";"LORG ",Position," =" ! LABEL line
150 !
160 ! ***** MOVE to point, LABEL an "X" to indicate initial PEN Position
●170 LORG 5                      ! Specifies LORG for the "X"
180 CSIZE 3                      ! Sets Character SIZE for the "X"
190 MOVE 8,-Position            ! MOVE to Position for the "X"
200 LABEL "X"                   ! LABEL the "X" to indicate PEN Position
210 !
220 ! ***** MOVE to point, LABEL with the word "Test"
●230 LORG Position              ! Specifies LORG for the word "Test"
240 CSIZE 8                      ! Sets Character SIZE for the word "Test"
250 MOVE 8,-Position            ! MOVE to Position for "X"
260 LABEL "Test"                ! LABEL "Test"
270 NEXT Position               ! Repeat for the NEXT LORG
280 END

```

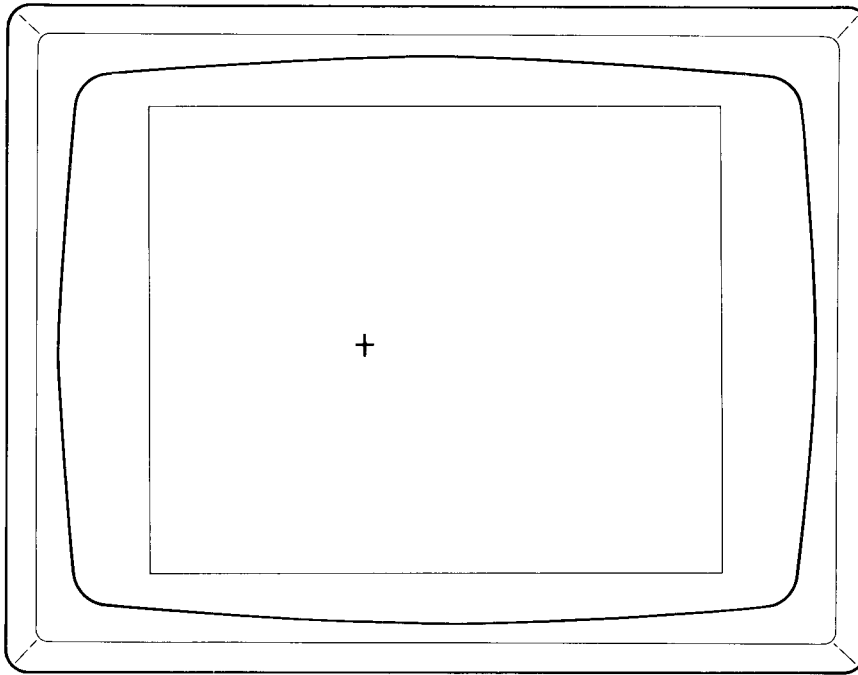
POINTER Examples



```

1  REM ** This is the POINTER example (POINT)
10 PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20 GRAPHICS                 ! Enables the CRT
30 FRAME                   ! FRAMES the SOFT CLIP area
40 SCALE 0,10,0,10        ! SCALES the LOCATE rectangle to 10x10
●50 POINTER 7,6,1         ! MOVES the CURSOR (large type) TO 7,6
60 END

```

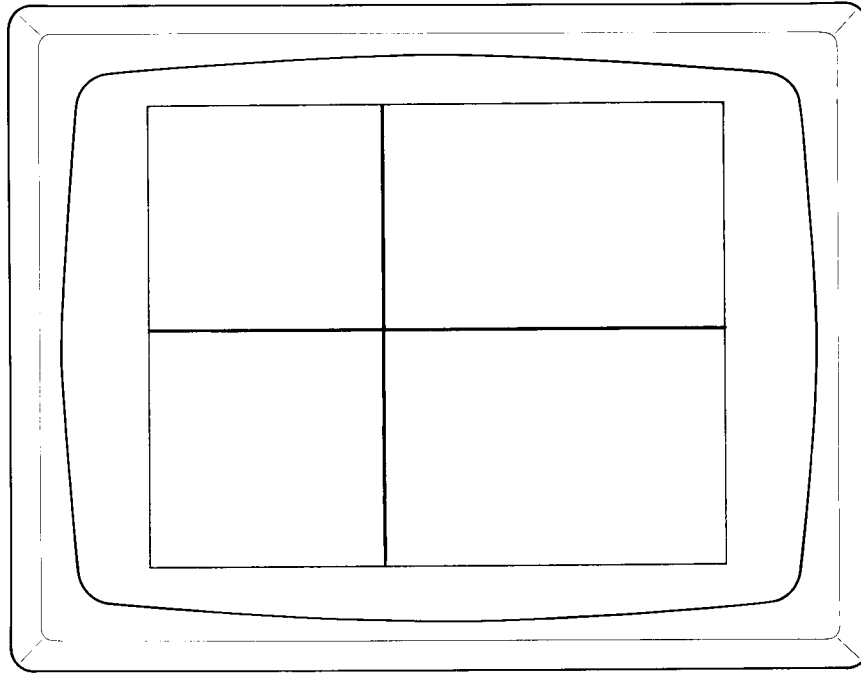


```

1  REM  ** This is another POINTER example (POINT2)
10 PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20 GRAPHICS                 ! Enables the CRT
30 FRAME                    ! FRAMES the SOFT CLIP area
40 SCALE 0,10,0,10         ! SCALES the LOCATE rectangle to 10x10
●50 POINTER 4,5,2           ! MOVES the CURSOR (small, flashing) to 4,5
60 END

```

CURSOR Example

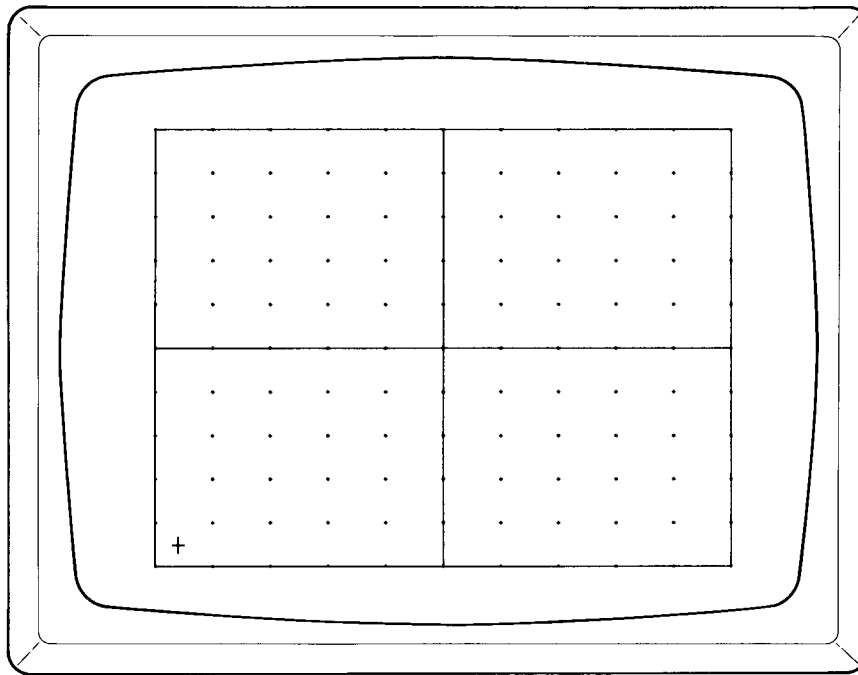


```

1  REM ** This is the CURSOR example (CURSOR)
10 PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20 GRAPHICS                ! Enables the CRT
30 FRAME                   ! FRAMES the SOFT CLIP area
40 PRINTER IS 0            ! Specifies the Thermal Printer
50 POINTER 50,50           ! MOVES the CURSOR to 50,50
●60 CURSOR Xvariable,Yvariable,Pen_status_$ ! Inputs CURSOR position
70 PRINT Xvariable,Yvariable,Pen_status_$   ! PRINTs CURSOR position
80 END

```

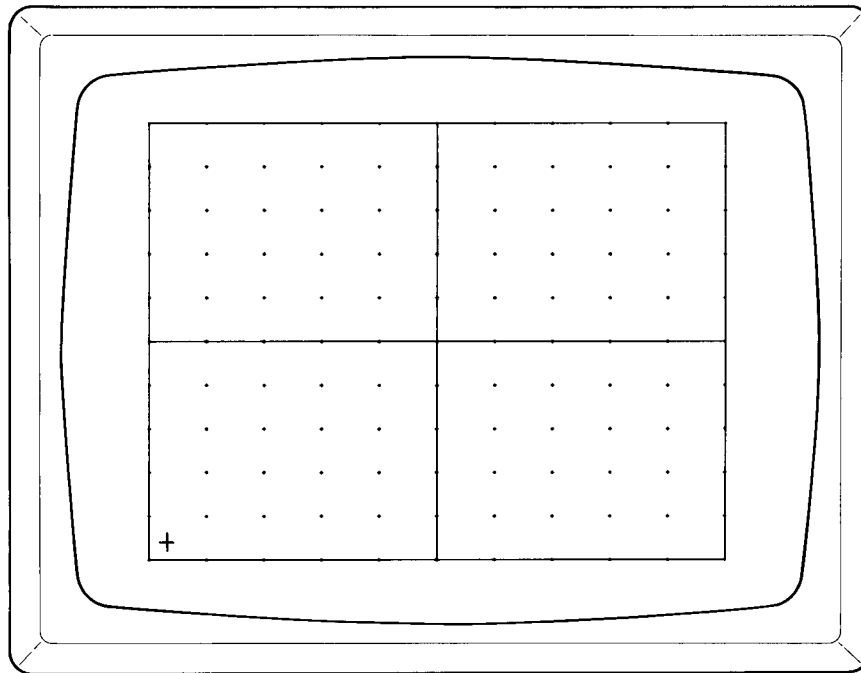
DIGITIZE Examples



```

1  REM  ** This is the digitizing example (GETIT)
2  REM  ** Note that the cursor and the pen are used
3  REM  ** simultaneously in this example, showing that
4  REM  ** they are two different units.
10 PLOTTER IS 13,"GRAPHICS"  ! Specifies the CRT as the plotter
20 GRAPHICS                  ! Enables the CRT
30 LIMIT 0,184,0,140         ! Selects the HARD CLIP limits
40 POINTER 4,4,2             ! Positions blinking CURSOR
50 SCALE 0,10,0,10          ! SCALES LOCATE rectangle to 20x20
60 GRID 1,1,0,0,5,5,1       ! Draws GRID with the PEN specifying:
61                             !   One minor tick per unit in X
62                             !   One minor tick per unit in Y
63                             !   Intersect at 0,0
64                             !   One major tick per 5 minor ticks in X
65                             !   One major tick per 5 minor ticks in Y
66                             !   Major tick size is 1 GDU
70 PRINTER IS 0              ! Selects the Thermal Printer
80 Get_it: !
●90 DIGITIZE X,Y             ! Input a coordinate
100 PRINT X,Y                ! Output the coordinate
110 GOTO Get_it              ! You must press STOP to STOP
120 END

```



```

1  REM ** This is the digitizing example (WHERE)
2  REM ** This also shows that the PEN and CURSOR are different
10 PLOTTER IS 13,"GRAPHICS" ! Specifies the CRT as the plotter
20 GRAPHICS ! Enables the CRT
30 LIMIT 0,184,0,140 ! Selects the HARD CLIP limits
40 POINTER 4,4,2 ! Positions blinking CURSOR
50 SCALE 0,10,0,10 ! SCALES LOCATE rectangle to 20x20
60 GRID 1,1,0,0,5,5,1 ! Draw GRID specifying:
61 ! One minor tick per unit in X
62 ! One minor tick per unit in Y
63 ! Intersect at 0,0
64 ! One major tick per 5 minor ticks in X
65 ! One major tick per 5 minor ticks in Y
66 ! Major tick size is 1 GDU
70 PRINTER IS 0 ! Selects the Thermal Printer
80 Get_it: !
90 WHERE Penx,Peny ! Input pen's coordinate
100 PRINT Penx,Peny ! Output the coordinate
●110 DIGITIZE X,Y ! Input a coordinate
120 PRINT X,Y ! Output the coordinate
130 GOTO Get_it
140 END

```

Glossary

Absolute Plotting ~ Plotting to a coordinate which has its X and Y values specified in the current user units.

Angle ~ The vector at which a line or label is drawn, represented in degrees counter-clockwise from the horizontal.

Anisotropic ~ The X and Y axes units are not displayed as equal to each other.

Axes ~ Plural of axis.

Axis ~ A line drawn within the Cartesian coordinate system along either the horizontal (X) direction or the vertical (Y) direction.

Clipping Area ~ The area which restricts the pen movement whenever lines are drawn in UDUs.

Current Units ~ The mode of X,Y reference which is in effect; may be either GDUs, UDUs or Metric (mm).

Cursor ~ The device which is used to obtain digitizing information.

Digitizing ~ The process of obtaining an X,Y coordinate pair based on the location of the cursor.

GDUs ~ Graphic Display Units. An X,Y reference system which at default defines the CRT to extend from X minimum = 0 to X maximum = 123 127753304, Y minimum = 0 to Y maximum = 100.

Hard Clip ~ The physical limits of the plotting device, beyond which no line can be drawn.

Appendix A

HPGL – The instruction set resident within the HP 9872A Graphics Plotter.

Isotropic – The X and Y axes units are displayed as equal to each other.

Metric Units – A unit of measure mode where everything is referenced in millimetres.

Origin – The coordinate point at which a plotting operation begins.

Pen – The device which is used to draw or plot lines, and to label characters.

Pointer – The method used to position the cursor, or to select a type of cursor.

Plotted Point – The point which has been plotted or drawn to.

Plotting Coordinates – The X,Y coordinate pair which specifies a plotting point.

Plotting Space – The area within which plotting can occur.

Reflected Plot – A plot produced by interchanging either the X min, X max coordinates, the Y min, Y max coordinates or both X and Y coordinate pairs to change the plot.

Relative Plotting – Plotting which specifies plotting from an origin rather than to a specific X,Y coordinate.

Soft Clip – The limits of the plotting device which restricts pen movement for lines drawn in UDU's.

Slant – The angle at which a character is drawn, represented in clockwise degrees from the vertical.

UDUs – User Defined Units. Defined by the program to whatever X and Y units of measure which are convenient.

Syntax Summary

AXES [X tick spacing, Y tick spacing [, X intersection, Y intersection [, X major count, Y major count [, major-tick size]]]]

The **AXES** statement draws a pair of axes with optional tick marks.

CLIP [X min, X max, Y min, Y max]

The **CLIP** statement defines the soft clip boundaries. Omitting the parameters allows any two diagonal corners to be digitized.

CSIZE height[, aspect ratio[, slant]]

The **CSIZE** (character size) statement is used to specify the size, aspect ratio and slant of characters used in labels. The height defaults to approximately 3.3 GDUs (15/4.54). The aspect ratio (width/height) defaults to 9/15. The slant defaults to 0 degrees.

CURSOR X variable, Y variable[, pen status string variable]

The **CURSOR** statement returns the values of the cursor coordinate to the specified variables. Pen status is assigned to the string variable; 0 for up, 1 for down.

DIGITIZE X variable, Y variable[, pen status string variable]

The **DIGITIZE** statement pauses program execution and allows you to reposition the cursor; execution is resumed by pressing the  key, any SFK or the  key. The coordinate of the cursor is assigned to the specified variables. Pen status is assigned to the string variable; 0 for up, 1 for down.

DRAW X parameter, Y parameter

The **DRAW** statement drops the pen and moves it to the absolute X,Y coordinate.

Appendix B

DUMP GRAPHICS [lower bound [, upper bound]

The DUMP GRAPHICS statement copies the CRT graphic display onto the internal thermal printer. Any horizontal area can be copied by specifying its upper and/or lower bound in current units.

EXIT GRAPHICS

The EXIT GRAPHICS statement returns the CRT to the alphanumeric raster from the graphics raster.

FRAME

The FRAME statement draws a box around the current soft clip area.

GCLEAR [paper advance]

The GCLEAR statement clears the CRT and the graphics R/W memory of previously plotted data. The paper advance value specifies how many millimetres of paper to advance on certain plotters; it has no effect on CRT graphics.

GLOAD integer array name (*)

GLOAD transfers data from an integer array to the CRT graphics memory. The first element in the integer array is the load pointer.

GRAPHICS

The GRAPHICS statement sets the CRT to the graphics mode.

GRID [X tick spacing, Y tick spacing[, X intersection, Y intersection[, X major count, Y major count[, major-tick size]]]]

The GRID statement can be used as an alternative to the AXES statement and is used to draw a full screen grid.

GSTORE integer array name (*)

GSTORE transfers data from the CRT graphics memory to an integer array. The first element in the integer array is the store pointer.

`IPLOT X increment, Y increment[, pen control]`

The `IPLOT` statement allows incremental plotting from the last plotted point. The pen control is the same as for the `PLOT` statement.

`LABEL list`

The `LABEL` statement is used like the `PRINT` statement to label.

`LABEL USING image specifier; list`

The `LABEL USING` statement is used like the `PRINT USING` statement to format labels.

`LDIR angle`

or `LDIR run, rise`

The `LDIR` statement specifies the angle at which subsequent labels will be drawn. The angle specifies counter-clockwise rotation of the label from the positive X-axis in current angular units.

`LETTER`

The `LETTER` statement allows you to draw all keyboard alphanumerics and symbols, by typing them in on the keyboard.

`LIMIT [X min, X max, Y min, Y max]`

The `LIMIT` statement defines the hard clip limits (or plotting area). The units are expressed in **millimetres** with the origin at the lower left physical limit. When the parameters aren't included, any two diagonal corner points can be digitized.

`LINE TYPE identifier number [, length]`

The `LINE TYPE` statement selects one of several solid or dashed line types. The range of the identifier number is 1 through 10; 4 is the default length.

LOCATE [X min, X max, Y min, Y max]

The LOCATE statement sets the area that SHOW will fill or SCALE will define. The units are expressed in GDU's. Any two diagonal corner points can be digitized if the parameters are not included. LOCATE also invokes soft clipping at its boundaries.

LORG origin position

The LORG (label origin) statement sets the label origin position which determines where any subsequent labels are drawn relative to the current pen location. The range of the origin position is 1 through 9.

MOVE X parameter, Y parameter

The MOVE statement lifts the pen and moves it to the absolute X,Y coordinates.

MSCALE X offset, Y offset

The MSCALE statement sets **millimetres** as current user units and defines the origin. The origin is offset from the first LOCATE point (X min, Y min) the specified amounts in millimetres.

PDIR angle

or PDIR run, rise

The PDIR statement sets the angle of rotation for relative and incremental plotting. The angle supplied specifies a counter-clockwise rotation from the positive X-axis in current angular units.

PEN pen number

The PEN statement specifies the pen to be used. Pen 0 (zero) specifies return all pens to their holders on the 9872A. Negative pen numbers specify "erase" on the CRT.

PENUP

The PENUP statement lifts the pen.

PLOT X parameter, Y parameter [, pen control]

The PLOT statement provides absolute data plotting and pen control. The pen control defaults to plus one. The pen control parameter is interpreted as follows:

Odd	=drop pen
Even	=lift pen
Positive	=pen change after motion
Negative	=pen change before motion

PLOTTER IS [select code[, HP-IB device address],] "plotter identifier string" [, step size[, # of pens, pen offset, [incremental plotter identifier]]]

The PLOTTER IS statement defines where all plotter operations will be directed. The three plotter identifier strings and their default select codes are:

"GRAPHICS"	13
"9872A"	7,5
"INCREMENTAL"	5

The default values for the INCREMENTAL parameters are:

step size	-	.254 mm
# of pens	-	1
pen offset	-	0 mm
incremental plotter i.d.	-	1

PLOTTER select code[, HP-IB device address] IS OFF

The PLOTTER...IS OFF statement deactivates the plotter set to the specified select code. All plotting statements are executed but the specified plotter does not respond.

PLOTTER select code[, HP-IB device address] IS ON

The PLOTTER...IS ON statement activates the specified plotting device and deactivates all others. That is, all plotting statements are directed only to the specified device. This statement **does not** set the default conditions described under the PLOTTER IS statement.

POINTER X parameter, Y parameter[, cursor type]

The POINTER statement moves the cursor to the specified absolute position and can select one of two types of cursor. An even number specifies a small blinking cross; an odd number specifies full-screen crossed lines. Cursor type defaults to the full-screen crossed lines (cursor type 1).

RATIO

The RATIO function returns a value equal to the ratio of the physical dimension of the hard clip limits. That is, the X dimension divided by the Y dimension.

RPLOT X parameter, Y parameter [, pen control]

The RPLOT statement allows relative plotting from the last absolute plotted point which is used as the origin. The pen control is the same as for the PLOT statement.

SCALE X min, X max, Y min, Y max

The SCALE statement sets user defined units which dimension the LOCATE rectangle.

SETGU

The SETGU statement sets graphic display units (GDUs) as the current units.

SETUU

The SETUU statement sets user defined units (UDUs) as the current units.

SHOW X min, X max, Y min, Y max

The SHOW statement defines an area that is stretched or shrunk equally in X,Y directions to fit within the boundaries defined by the LOCATE statement.

UNCLIP

The `UNCLIP` statement sets the soft clip boundaries equal to the hard clip limits.

`WHERE X variable, Y variable[, pen status string variable]`

The `WHERE` statement returns the coordinate values of the last plotted or moved-to point.

Pen status information is assigned to the string variable; 0 for up, 1 for down.

Graphics Device Information

CRT Graphics Hardware Installation

The Graphics package for the System 45B consists of:

- 98437B Graphics ROM
- 98470B CRT Graphics Memory
- 09845-91050 Graphics ROM Programming Manual

Factory Installed Graphics

If your System 45B was ordered from the factory with the Graphic Option (Opt. 311 and Opt. 700), the hardware (ROM, CRT memory and error message sticker) is already installed for you.

Field Installed Graphics

If you are updating your System 45B with graphics, the hardware has to be installed. The CRT Graphics Memory should be installed by trained HP Service personnel, because installation procedures involve high voltage and a potentially dangerous situation associated with handling CRT's.

You install the Graphics ROM in one of the open slots in the ROM drawer on the right side of the computer. If you are unfamiliar with procedures for installing a ROM, refer to the System 45B Owner's Manual for proper procedures. You can also install the Graphics error message sticker on your option error pull-out card. If you want, the service person will install the Graphics ROM and the error message sticker when the Graphics Memory is installed.

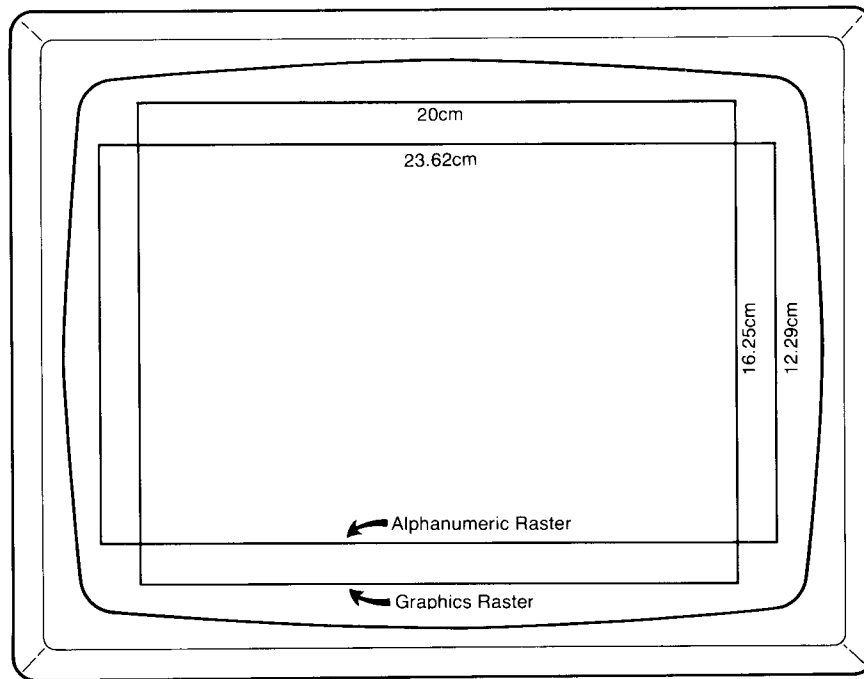
Appendix C

CRT Graphics Specifications

System 45B has two CRT display modes: alphanumeric and graphics. Each mode has its own display area and raster. Normally the CRT is in the alphanumeric mode and without the Graphics ROM and hardware, this is the only mode available to you. When the Graphics ROM and hardware is installed, the graphics mode can be selected.

Display Area

The following figure shows the approximate size and location of the alphanumeric and graphics display areas.



Graphics Resolution

The horizontal and vertical resolution is approximately .04cm. This results in a dot matrix pattern for the display area of 560 usable dots in the horizontal direction and 455 usable dots in the vertical direction.

Line Generation Speed

Straight lines are generated on the CRT at about 200 cm/sec. (The 9872A plotter generates straight lines at about 36 cm/sec.) Curved lines are generated at somewhat slower speeds because your System 45B first computes the points and then plots the curved line.

Graphics Select Code

The select code of the CRT graphics option is internally set at the factory to 13 and cannot be changed. If you are going to use an external plotting device (e.g., HP 9872A) refer to the appropriate Interface Installation and Service Manual and this appendix for information about how to set the select code.

Graphics Memory

The Graphics ROM automatically decreases the available user memory by 90 bytes. This is generally not a concern unless you have a large program or large data base.

The 32K bytes of Read/Write Memory (CRT Graphics Memory) installed in the CRT does not increase your available R/W memory, it only provides a storage location for data which produces the graphics display on the CRT. When using an external display or plotting device, this memory space is not used.

HP 9872A Plotter Installation

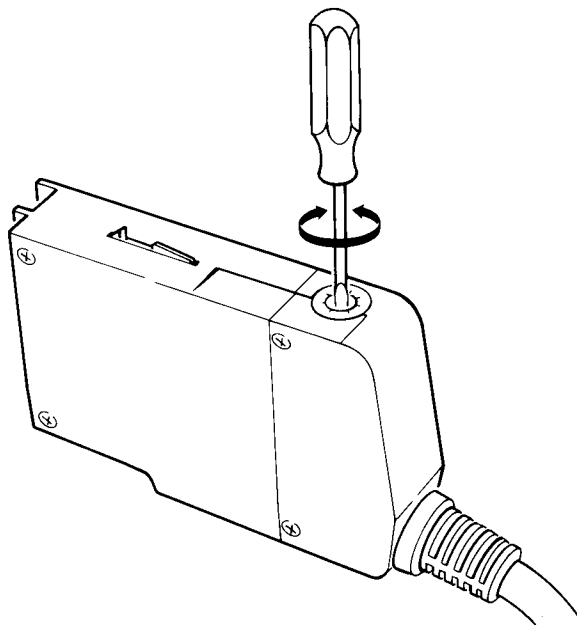
The 9872A Plotter can be used as an external plotter with the System 45B. Refer to the PLOTTER IS statement for information on how to select the 9872A as the active plotter.

Ensure that the interface select code switch and plotter address code switches are set to the proper positions as indicated below.

Interface Select Code

The 9872A Plotter is connected to the System 45B via an HP 98034A HP-IB Interface. Refer to the HP 98034A HP-IB Interface Installation and Service Manual for complete information about the interface and cables.

The interface is preset to a select code of 7 at the factory. To change the setting, rotate the switch (shown following) using a small screwdriver.



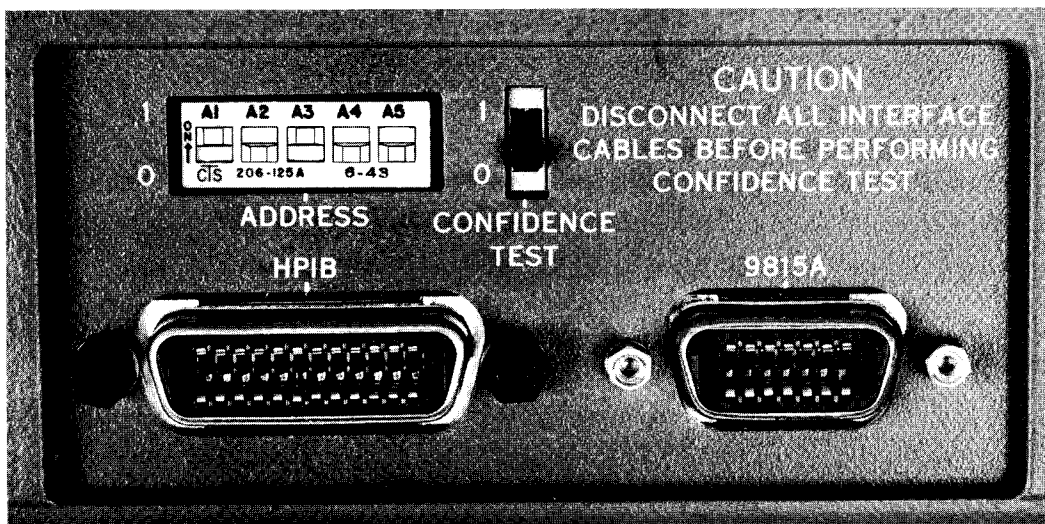
Select Code Switch

Plotter Address Code

Since each HP-IB interface can have as many as 14 devices connected to it, each device must be set to a specific address code.

The plotter can be set to any one of 31 HP-IB addresses ranging from 0 thru 30. Each address can be selected by setting the switches on the plotter back panel (shown next) to the appropriate binary bit positions for the particular address value desired.

The plotter is set to an address code of 05 at the factory. Check your plotter for the proper switch positions shown below.



Plotter Address Switches

The following table lists the switch positions for each address value.

Address Switch Positions

Address Characters		Address Switch Settings					Address Codes	
Listen	Talk	(5)	(4)	(3)	(2)	(1)	decimal	octal
SP	@	0	0	0	0	0	0	0
!	A	0	0	0	0	1	1	1
"	B	0	0	0	1	0	2	2
#	C	0	0	0	1	1	3	3
\$	D	0	0	1	0	0	4	4
%	E	0	0	1	0	1	5	5 ← preset
&	F	0	0	1	1	0	6	6
'	G	0	0	1	1	1	7	7
(	H	0	1	0	0	0	8	10
)	I	0	1	0	0	1	9	11
*	J	0	1	0	1	0	10	12
+	K	0	1	0	1	1	11	13
,	L	0	1	1	0	0	12	14
-	M	0	1	1	0	1	13	15
.	N	0	1	1	1	0	14	16
/	O	0	1	1	1	1	15	17
0	P	1	0	0	0	0	16	20
1	Q	1	0	0	0	1	17	21
2	R	1	0	0	1	0	18	22
3	S	1	0	0	1	1	19	23
4	T	1	0	1	0	0	20	24
5	U	1	0	1	0	1	21	25
6	V	1	0	1	1	0	22	26
7	W	1	0	1	1	1	23	27
8	X	1	1	0	0	0	24	30
9	Y	1	1	0	0	1	25	31
:	Z	1	1	0	1	0	26	32
;	[	1	1	0	1	1	27	33
<	/	1	1	1	0	0	28	34
=	]	1	1	1	0	1	29	35
>	^	1	1	1	1	0	30	36

Incremental Plotter Installation

An incremental plotter can be used as an external plotter with the System 45B, using the HP 98040A Interface. Refer to the PLOTTER IS statement for information on how to select the incremental plotter as the plotter.

Ensure that the interface select code switch is set to the proper position.

Refer to the HP 98040A Interface Installation and Service Manual for information concerning the select code switch and a list of incremental plotters that can be used with this interface.

System 45A and B Graphics Comparison

The System 45B differs from the System 45A in the following statements –

Csize	incorporates a character slant as its third parameter.
Gload	is resident in the Graphics ROM rather than the System 45A Mass Storage ROM.
Gstore	is resident in the Graphics ROM rather than the System 45A Mass Storage ROM.
LDIR	incorporates a run, rise parameter as a means of specifying an angle.
LINE TYPE	The line types are slightly different from the System 45A's line types. A line type 10 with a major tick mark is now incorporated.
PDIR	incorporates a run, rise parameter as a means of specifying an angle.

Note

Hewlett-Packard products which use cathode ray tubes are designed to limit X-radiation levels to values of 0.5 mR/hour or below. This level has been established by the United States Bureau of Radiological Health, and conforms to similar international requirements. All measurements are made using a Victoreen 440 RF/c instrument, and are conducted with the equipment operating under normal as well as worse case control midadjustment.

Subject Index

9872A Plotter Installation 195

a

AXES 60,65,66,67
 Absolute Plotted Point 76
 Plotting 65,66
 Alignment of Text 86,102
 Angle 99
 of Rotation 81,99
 Apparent Motion 81
 Appendices 6
 Appendix A 6,183
 B 6,185
 C 6,193
 Area, Hard Clip 10
 Soft Clip 10
 Aspect Ratio 94

b

Bezier Curve Smoothing 137
 Blinking Cursor 111

c

CHR\$(3) 87
 CLIP 37,38,39,44,47,48,60,74
 CONT 5
 CONT1 13
 CONT2 15
 CONT3 17
 CONT4 21
 CONT5 25
 CONT6 27
 CONT7 29
 CONT8 30,31
 CONT9 33
 CONT10 35
 CONT11 43
 CONT12 45
 CONT13 49
 CONT14 51
 CONT15 54
 CONT16 57
 CONT17 59

CONT18 62
 CONT19 69
 CONT20 71
 CONT21 73
 CONT22 75
 CONT23 89
 CONT24 93
 CONT25 96
 CONT26 101
 CONT27 103
 CONT28 113
 CONT29 115
 CONT30 119
 CONT31 121
 CONT32 126
 CONT33 133
 CONT34 136
 CRT 11
 Graphics Specifications 194
 CSIZE 85,86,87,94
 CURSOR 60,109,110,114
 Changing Soft Clip 48
 Character Cells 94
 Height 94
 Set 172
 Slant 94,97
 Clearing R/W memory 28
 the CRT 28
 Clip Limits, Hard 3
 Soft 3
 Coordinate System 11
 Copying the plot 32
 Current Units 4,37,39
 Cursor 109
 Curve Fitting 137

d

DIGITIZE 42,44,60,109,110,116
 DRAW 34,60,65,66,74
 DUMP GRAPHICS 9,32
 Default Conditions 24
 Difference between IPLOT & RPLOT 80
 Digitizing Operations 110
 from the CRT 116
 with the 9872A 117
 with the Incremental Plotter 117
 Display Area 194
 Dracula Plot 98

e

EXIT GRAPHICS 9,12
 Erasing Characters 91
 Graphics R/W Memory 28
 lines on the CRT 14
 the CRT 28
 Example, AXES 68,164,165
 CLIP 45,153
 CSIZE 95,171
 CURSOR 114,180
 Character Set 172,173
 DIGITIZE 118,181
 DRAW 74,168
 DUMP GRAPHICS 33
 EXIT GRAPHICS 13
 FRAME 30,31,151
 Football Field 146
 GCLEAR 29
 GDU 163
 GLOAD 125,128
 GRAPHICS 13
 GRID 70,166
 GSTORE 125,128
 Histogram 154
 IPLOT 79
 LABEL 88,169
 LABEL USING 88
 LDIR 100,175
 LETTER 92,170
 LIMIT 43,152
 LINE TYPE 17,18,19
 LOCATE 51,157
 LOG/LIN Chart 148
 LORG 104,176,177
 MOVE 72,167
 MSCALE 59
 Menu Selection 140
 Metric Scale 162
 PDIR 82
 PEN 15
 PENUP 35
 PLOT 21,149
 PLOTTER IS 25
 PLOTTER IS OFF 27
 PLOTTER IS ON 27
 POINTER 112,178,179
 Plotting Outside the Hard Clip
 Area 155

RATIO 63
 RPLOT 77
 Reflected Plot 132
 SCALE 54,55,158,159,160,161
 SETGU 61,62
 SETUU 61,62
 SHOW 57,158,159,160,161

UDU 163
 UNCLIP 49,156
 WHERE 120,182

Examples 4,5
 Exchanging Parameters 130

f

FRAME 9,30
 Field Installed Graphics 193
 Full Screen Cursor 111
 Function, RATIO 37,62

g

GCLEAR 9,28
 GDU's 3,7,12,39,44,50,60,61,74,76,78,94
 GLOAD 123,124,125,128
 GL-GS 129
 GRAPHICS 9,12
 GRID 60,65,66,70
 GSTORE 123,124,126,128
 Glossary 183,184
 Graphic Display Units 3,7,12,39
 Graphics Characters 172
 Device Information 192,193,194
 R/W Memory 195
 Resolution 194
 Select Code 195

h

HPGL	134
Hard Clip	74,76,78,87
Area	10
Limits	3,37,39
Units	39
Hard Copy	32
Height, Character	94

i

IPLOT	20,34,60,65,66,78,80
Identifier Strings, Plotter	22
Incremental Plotter Installation	197
Parameters	23
Index	199
Integer Array	124,125
Isotropic Scaling	56

j

Jaggies	14
Justification of Text	86,102

l

LABEL	85,86,87
LABEL USING	85,86,87
LDIR	85,86,87,99
LETTER	85,86,90
LIMIT 37,38,39,40,41,42,44,47,48,50,61,94,130,135	
LINE	19
LINE TYPE	9,16
LITTLE	87
LOCATE	37,38,40,44,48,50,53,56
LORG	85,86,87,102,104,106
Label Origin	102
Labeling Angle	99
Line	95
Line Generation	4
Line Generation Speed	194

m

MOVE	20,34,60,65,72
MSCALE	37,38,40,58,61
Machine Mode	97
Menu Selection	138
Metric Units	3,7,58
Moving the Cursor	90
Pen	72

o

Offsetting the CRT	58
--------------------------	----

p

P1	39,42
P2	39,42
PDIR	65,76,78,81,82
PEN	9,14
PENUP	9,20,34
PLOT	9,20,34,60,66
PLOTTER IS 9,22,23,24,42,44,48,61,94,135	
PLOTTER IS OFF	9,26
PLOTTER IS ON	9,26
POINTER	60,109,110,111
Paper Advance	28
Pen	109
Control	20,76,78
Location	120
Status Variable	114,116
vs Cursor	110
Plotter Address Code	196
Identifier Strings	22
Interface Select Code	195
Plotting Area	41
Boundaries	10
Coordinates	9,10,11
Operations	110
Space	3
Units	60
in GDU's	46
in UDU's	46
Positioning the plot	38,39

r

RATIO 37,41,62,63
 RPLOT 20,34,60,65,66,76,77,80
 Reflected Plots 123,130
 Relative Plotting 65,66
 Resolution Unit 91
 Rotating the Plot 81
 Run and Rise 81,99

s

SCALE 37,38,40,48,50,53,55,61,130
 SETGU 37,41,61
 SETUU 37,41,61
 SHOW 37,38,40,48,50,56,61,130
 Scaling 40
 the Plotting Area 38,52
 Slant, Character 94
 Slant 97
 Slowing the Pen Speed 135
 Soft Clip 74,76,78,87
 Area 10,44
 Limits 3,37,39
 Space 95
 Plotting 3
 Specifying Minor Tick Size 70
 the Cursor 111
 Statement, AXES 67
 CLIP 44
 CSIZE 94
 CURSOR 114
 DIGITIZE 116
 DRAW 74
 DUMP GRAPHICS 32
 EXIT GRAPHICS 12
 FRAME 30
 GCLEAR 28
 GLOAD 124
 GRAPHICS 12
 GRID 70
 GSTORE 124
 IPLOT 78
 LABEL 87
 LABEL USING 87
 LDIR 99
 LETTER 90

LIMIT 41
 LINE TYPE 16
 LOCATE 50
 LONG 102
 MOVE 72
 MSCALE 58
 PDIR 81
 PEN 14
 PENUP 34
 PLOT 20
 PLOTTER IS 22
 PLOTTER IS OFF 26
 PLOTTER IS ON 26
 POINTER 111
 RPLOT 76
 SCALE 53
 SETGU 61
 SETUU 61
 SHOW 56
 UNCLIP 48
 WHERE 120

Statements Using the Cursor 110
 Syntax Guidelines 6
 Summary 185,186,187,
 188,189,190,191
 System 45A and B Graphics Com-
 parisons 198

t

Terminating LABEL 87
 LETTER 91
 Tick Marks 67,70
 Typing Aids 91,92
 on a Plot 90

u

UDU's 3,7,12,39,44,50,53,60,61,74,76,78
 UNCLIP 37,38,39,44,48
 Units, Current 4,39
 GDU 3,7
 Graphic Display 7,12,39
 Metric 3,7,58
 UDU 3,7
 User Defined 7,12,39
 User Defined Units 3,7,12,39
 Using the 9872A Plotter 134

V

Vampires 98

Viewing the plot 12

W

WHERE 20,34,60,109,110,120

Window 47

SALES & SERVICE OFFICES

AFRICA, ASIA, AUSTRALIA

ANGOLA

Empresa Técnica de Equipamentos
Elétricos, S.A.R.L.
R. Barbosa Rodrigues, 42-10T.
Caixa Postal, 6487
Lusanda
Tel: 35515/6
Cable: TELETRA Lusanda

AUSTRALIA

Hewlett-Packard Australia
Pty. Ltd.
31-41 Joseph Street
Blackburn, Victoria 3130
P.O. Box 36
Doncaster East, Victoria 3109
Tel: 896351
Telex: 31-024
Cable: HEWPARD Melbourne
Hewlett-Packard Australia
Pty. Ltd.
31 Bridge Street
Pymble
New South Wales, 2073
Tel: 4496566
Telex: 21561
Cable: HEWPARD Sydney
Hewlett-Packard Australia
Pty. Ltd.
153 Greenhill Road
Parkside, S.A., 5063
Tel: 2725911
Telex: 82536
Cable: HEWPARD Adelaide
Hewlett-Packard Australia
Pty. Ltd.
141 String Highway
Nedlands, W.A., 6009
Tel: 3865455
Telex: 93859
Cable: HEWPARD Perth
Hewlett-Packard Australia
Pty. Ltd.
121 Wollongong Street
Fyshwick, A.C.T., 2609
Tel: 804244
Telex: 62650
Cable: HEWPARD Canberra
Hewlett-Packard Australia
Pty. Ltd.
5th Floor
Teachers Union Building
495-499 Boundary Street
Spring Hill, Queensland 4000
Tel: 2291544
Cable: HEWPARD Brisbane

BANGLADESH

The General Electric Co.
of Bangladesh
Magnet House 72
Dhaka Commercial Area
Motijheel, Dhaka 2
Tel: 252415, 252419
Telex: 734
Cable: GECDAC Dacca

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

BHUTAN

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-8414

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

BHUTAN

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-8414

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

BHUTAN

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-8414

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

BHUTAN

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-8414

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

BHUTAN

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-8414

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

BHUTAN

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-8414

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

BHUTAN

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-8414

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

BHUTAN

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-8414

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

BHUTAN

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-8414

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

BHUTAN

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-8414

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

BHUTAN

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-8414

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

BHUTAN

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-8414

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

BHUTAN

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-8414

BHUTAN

Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3570
TWX: 610-631-2431

ETHIOPIA

Abdella Abdulmalk
P.O. Box 2635
Addis Ababa
Tel: 11 93 40
GUAM
Medical Only
Guam Medical Supply, Inc.
Suite C, Airport Plaza
P.O. Box 8947
Tamuning 96911
Tel: 848-4513
Cable: EARMED Guam

HONG KONG

Schmidt & Co. (Hong Kong) Ltd.
Wing On Centre, 28th Floor
Connaught Road, C.
Hong Kong
Tel: 5-45544
Cable: 74766 SCHMC HX

INDIA

Blue Star Ltd.
Kasturi Buildings
Jamshedji Tata Rd
Bombay 400 020
Tel: 29 50 21
Telex: 011-2156
Cable: BLUEFROST
Blue Star Ltd.
Sahas
414/2 Vir Savarkar Marg
Prabhadevi
Bombay 400 025
Tel: 45 76 87
Telex: 011-4093
Cable: FROSTBLUE

INDONESIA

BERCA Indonesia P.T.
P.O. Box 4964Jkt
Jin Abdul Muis 62
Jakarta
Tel: 349255, 349866
Telex: 46748 BERSIL IA
Cable: BERSAL
BERCA Indonesia P.T.
P.O. Box 174/Sby.
23 Jln. Jimerito
Surabaya
Tel: 42027
Cable: Bercaron

ISRAEL

Electronics Engineering Div.
of Motorola Israel Ltd.
16, Kremenetski Street
P.O. Box 25016
Tel-Aviv
Tel: 38973
Telex: 33569, 34164
Cable: BASTEL Tel-Aviv

JAPAN

Yokogawa-Hewlett-Packard Ltd.
Chuo Bldg., 4th Floor
4-20, Nishinagaya 5-chome
Yokogawa-ku, Osaka-shi
Osaka 532
Tel: 06-304-6021
Telex: 523-3624
Yokogawa-Hewlett-Packard Ltd.
29-21, Takaido-Higashi 3-chome
Suginami-ku, Tokyo 168
Tel: 03-331-6111
Telex: 232-2024 YHP-Tokyo
Cable: YHPMARKET TOK 23 724

Blue Star Ltd.

Blue Star House
11/11A Magarath Road
Bangalore 560 025
Tel: 55668
Telex: 043-430
Cable: BLUESTAR
Blue Star Ltd.
Meekshi Mandiram
xxx/1678 Mahatma Gandhi Rd.
Cochin 682 016
Tel: 32069, 32161, 32282
Telex: 0865-514
Cable: BLUESTAR

Blue Star Ltd.

1-11-1171
Sarajini Devi Road
Secunderabad 500 003
Tel: 70126, 70127
Telex: 015-459
Cable: BLUEFROST
Blue Star Ltd.
234 Kodambakkam High Road
Madras 600 034
Tel: 82056
Telex: 041-379
Cable: BLUESTAR

INDONESIA

BERCA Indonesia P.T.
P.O. Box 4964Jkt
Jin Abdul Muis 62
Jakarta
Tel: 349255, 349866
Telex: 46748 BERSIL IA
Cable: BERSAL
BERCA Indonesia P.T.
P.O. Box 174/Sby.
23 Jln. Jimerito
Surabaya
Tel: 42027
Cable: Bercaron

ISRAEL

Electronics Engineering Div.
of Motorola Israel Ltd.
16, Kremenetski Street
P.O. Box 25016
Tel-Aviv
Tel: 38973
Telex: 33569, 34164
Cable: BASTEL Tel-Aviv

JAPAN

Yokogawa-Hewlett-Packard Ltd.
Chuo Bldg., 4th Floor
4-20, Nishinagaya 5-chome
Yokogawa-ku, Osaka-shi
Osaka 532
Tel: 06-304-6021
Telex: 523-3624
Yokogawa-Hewlett-Packard Ltd.
29-21, Takaido-Higashi 3-chome
Suginami-ku, Tokyo 168
Tel: 03-331-6111
Telex: 232-2024 YHP-Tokyo
Cable: YHPMARKET TOK 23 724

Blue Star Ltd.

Blue Star House
11/11A Magarath Road
Bangalore 560 025
Tel: 55668
Telex: 043-430
Cable: BLUESTAR
Blue Star Ltd.
Meekshi Mandiram
xxx/1678 Mahatma Gandhi Rd.
Cochin 682 016
Tel: 32069, 32161, 32282
Telex: 0865-514
Cable: BLUESTAR

Yokogawa-Hewlett-Packard Ltd.

Nakamo Building
24 Kami Sasajima-cho
Nakamura-ku, Nagoya, 450
Tel: 052-571-5171
Yokogawa-Hewlett-Packard Ltd.
Tangawa Building
2-24-1 Tsuruya-cho
Kanagawa-ku
Yokohama, 221
Tel: 045-312-1252
Telex: 382-3204 YHP YOK

Yokogawa-Hewlett-Packard Ltd.

Mito Mitsui Building
105, 1-chome, San-no-maru
Mito, Ibaragi 310
Tel: 0292-25-7470
Yokogawa-Hewlett-Packard Ltd.
Inoue Building
1348-3, Asahi-cho, 1-chome
Atsugi, Kanagawa 243
Tel: 0462-24-0452
Yokogawa-Hewlett-Packard Ltd.
Hachijuni Building
4th Floor
3-4, Tsukuba
Kumagaya, Saitama 380
Tel: 0485-24-6563

KENYA

Advanced Communications Ltd.
P.O. Box 30070
Nairobi
Tel: 331955
Cable: HEWPACK Nairobi
Medical Only
International Aeradio (E.A.) Ltd.
P.O. Box 19012
Nairobi Airport
Nairobi
Tel: 336055/6
Telex: 22201/22301
Cable: INTAERIO Nairobi

KOREA

Samsung Electronics Co., Ltd.
15th Floor, Daeyongak Bldg.,
25-5, 1-KA
Choong Moo-Ro, Chung-Ku,
Seoul
Tel: (23) 6811, 778-3401/2/3/4
Telex: 22575
Cable: DENTAL Christchurch

Medical Only

International Aeradio (E.A.) Ltd.
P.O. Box 95221
Mombasa

Malaysia

Sales SDN BHD
Suite 2, 21/22
Bangunan Anglisa Raya
Jalan Ampang
Kuala Lumpur
Tel: 2320/27491

MOZAMBIQUE

A.N. Gonçalves, Ltd.
162, 1st Apr. 14 Av. D. Luis
Caixa Postal 107
Maputo
Tel: 27091, 27114
Telex: 6-203 NEGON Mo
Cable: NEGON

NEW GUINEA

Hewlett-Packard Australia
Pty. Ltd.
Development Bank Building
Ground Floor
Ward Strip
Port Moresby, Papua
Tel: 258933
Cable: HEWPACK Wellington

NEW ZEALAND

Hewlett-Packard (N.Z.) Ltd.
4-12 Cruickshank Street
Kilbirnie, Wellington 3
P.O. Box 9443
Courtney Place
Wellington
Tel: 877-199
Cable: HEWPACK Wellington

Philippines

The Online Advanced
Systems Corporation
Rico House
Amorsolo cor. Herrera Str.
Lapasa Village, Makati
P.O. Box 1510
Metro Manila
Tel: 85-35-81, 85-34-91, 85-32-21
Telex: 3274 ONLINE

RHODESIA

Field Technical Sales
45 Kelvin Road North
P.O. Box 3458
Salisbury
Tel: 705231 (5 lines)
Telex: RH 4122

SINGAPORE

Hewlett-Packard Singapore
Pte. Ltd.
1150 Depot Road
Alexandra P.O. Box 58
Singapore 4
Tel: 270-2355
Telex: HPSG RS 21486
Cable: HEWPACK, Singapore

SOUTH AFRICA

Hewlett-Packard South Africa
Pty. Ltd.
Private Bag Wendywood,
Sandton, Transvaal, 2144
Hewlett-Packard Centre
Daphne Street, Wendywood,
Sandton, 2144
Tel: 802-1040/6
Telex: 8-4782
Cable: HEWPACK Johannesburg

ONTARIO

Hewlett-Packard (Canada) Ltd.
1020 Morrison Dr.
Ottawa K2H 8K7
Tel: (613) 820-6483
TWX: 610-563-1636
Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
Mississauga L4V 1M6
Tel: (416) 676-9430
TWX: 610-492-4246
Hewlett-Packard (Canada) Ltd.
552 Newbold Street
London N6E 2S5
Tel: (519) 686-9181

QUEBEC

Hewlett-Packard (Canada) Ltd.
275 Hymus Blvd.
Pointe Claire H9R 1G7
Tel: (514) 697-4232
TWX: 610-422-3022
TLX: 05-821-521 HPCL

MEXICO

Hewlett-Packard Mexicana,
S.A. de C.V.
Av. Periferico Sur No. 6501
Tepic, Jalisco
Mexico 23, D.F.
Tel: 905-676-4900
Telex: 017-74-507
Hewlett-Packard Mexicana,
S.A. de C.V.
Ave. Constitucion No. 2184
Monterrey, N.L.
Tel: 48-71-32, 48-71-94
Telex: 038-410
NICARAGUA
Roberto Terán G.
Apartado Postal 689
Edificio Terán
Managua
Tel: 25114, 23412, 23454, 22400
Cable: ROTERAN Managua

PANAMA

Eléctrico Balboa, S.A.
Apartado 4929
Panama 5
Calle Samuel Lewis
Edificio "Alfa", No. 2
Cuidad de Panama
Tel: 46-2700
Telex: 702-103 Curundu,
Canal Zone
Cable: ELECTRON Panama

NIGERIA

The Electronics
Instrumentations Ltd.
N88/770 Oyo Road
Oluseun House
P.M.B. 5402
Ibadan
Tel: 461577
Telex: 31231 TEIL NG
Cable: THETIEL Ibadan

PAKISTAN

Mushko & Company Ltd.
Osman Chambers
Abdullah Haroon Road
Karachi-3
Tel: 511027, 512927
Cable: COOPERATOR Karachi
Mushko & Company, Ltd.
388, Satellite Town
Rawalpindi
Tel: 41924
Cable: FEMUS Rawalpindi

PHILIPPINES

The Online Advanced
Systems Corporation
Rico House
Amorsolo cor. Herrera Str.
Lapasa Village, Makati
P.O. Box 1510
Metro Manila
Tel: 85-35-81, 85-34-91, 85-32-21
Telex: 3274 ONLINE

RHODESIA

Field Technical Sales
45 Kelvin Road North
P.O. Box 3458
Salisbury
Tel: 705231 (5 lines)
Telex: RH 4122

SINGAPORE

Hewlett-Packard Singapore
Pte. Ltd.
1150 Depot Road
Alexandra P.O. Box 58
Singapore 4
Tel: 270-2355
Telex: HPSG RS 21486
Cable: HEWPACK, Singapore

SOUTH AFRICA

Hewlett-Packard South Africa
Pty. Ltd.
Private Bag Wendywood,
Sandton, Transvaal, 2144
Hewlett-Packard Centre
Daphne Street, Wendywood,
Sandton, 2144
Tel: 802-1040/6
Telex: 8-4782
Cable: HEWPACK Johannesburg

Hewlett-Packard South Africa

(Pty.) Ltd.
P.O. Box 120
Howard Place, Cape Province, 7450
Pine Park Centre, Forest Drive,
Pinelands, Cape Province, 7405
Tel: 53-7955 thru 9
Telex: 57-0006

SRI LANKA

Metropolitan Agencies Ltd.
209/9 Union Place
Colombo 2
Tel: 35947
Telex: 1377METROLTD CE
Cable: METROLTD

SUDAN

Radison Trade
P.O. Box 921
Khartoum
Tel: 44048
Telex: 375

TAIWAN

Hewlett-Packard Far East Ltd.
Taiwan Branch
39 Chung Hsiao West Road
Section 1, 7th Floor
Taipei
Tel: 3819160-9, 3141010
Cable: HEWPACK TAIPEI
Hewlett-Packard Far East Ltd.
Taiwan Branch
68-2, Chung Cheng 3rd. Road
Kachuang
Tel: (07) 242318-Kaohsiung
Analytical Only
San Kwang Instruments Co., Ltd.
20 Yung Siu Road
Taipei
Tel: 3615466-9 (4 lines)
Telex: 22894 SANKWANG
Cable: SANKWANG Taipei

TANZANIA

Medical Only
International Aeradio (E.A.) Ltd.
P.O. Box 861
Dar es Salaam
Tel: 21251 Ext. 265
Telex:

Your Comments, Please...

Your comments assist us in improving the usefulness of our publications; they are an important part of the inputs used in preparing updates to the publications.

In order to write this manual, we made certain assumptions about your computer background. By completing and returning the comments card on the following page you can assist us in adjusting our assumptions and improving our manuals.

Feel free to mark more than one reply to a question and to make any additional comments.

Please do not use this form for questions about technical applications of your system or requests for additional publications. Instead, direct those inquiries or requests to your nearest HP Sales and Service Office.

If the comments card is missing, please address your comments to:

HEWLETT-PACKARD COMPANY
Desktop Computer Division
3404 East Harmony Road
Fort Collins, Colorado 80525 U.S.A.

Attn. Customer Documentation
Dept. 4231

All comments and suggestions become the property of Hewlett-Packard.

System 45B Graphics Reference Card

Graphics Syntax Summary

AXES [X tick spacing, Y tick spacing [, X intersection,
Y intersection [, X major count, Y major count
[, major-tick size]]]]

CLIP [X min, X max, Y min, Y max]

CSIZE height [, aspect ratio [, character slant]]

CURSOR X variable, Y variable [, pen status string variable]

DIGITIZE X variable, Y variable [, pen status string variable]

DRAW X parameter, Y parameter

DUMP GRAPHICS [lower bound[, upper bound]

EXIT GRAPHICS

FRAME

GCLEAR [paper advance]

GLOAD integer array name (*)

GRAPHICS

GRID [X tick spacing, Y tick spacing [, X intersection,
Y intersection [, X major count, Y major count
[, major-tic size]]]]

GSTORE integer array name (*)

IPLOT X increment, Y increment [, pen control]

LABEL list

LABEL USING image specifier; list

LDIR angle or LDIR run, rise

LETTER

LIMIT [X min, X max, Y min, Y max]

LINE TYPE identifier number [, length]

LOCATE [X min, X max, Y min, Y max]

LORG origin position

MOVE X parameter, Y parameter

NSCALE X offset, Y offset

PDIR angle or PDIR run, rise

PEN pen number

PENUP

PLOT X parameter, Y parameter[, pen control]

PLOTTER IS [select code[, HP-IB device address],
"plotter identifier string" [, step size[, # of pens,
pen offset, [incremental plotter identifier]]]]

PLOTTER select code[, HP-IB device address] IS OFF

PLOTTER select code[, HP-IB device address] IS ON

POINTER X parameter, Y parameter[, cursor type]

RATIO

RPLOT X parameter, Y parameter[, pen control]

SCALE X min, X max, Y min, Y max

SETCU

SETUU

SHOW X min, X max, Y min, Y max

UNCLIP

WHERE X variable, Y variable [, pen status string variable]

Default Settings

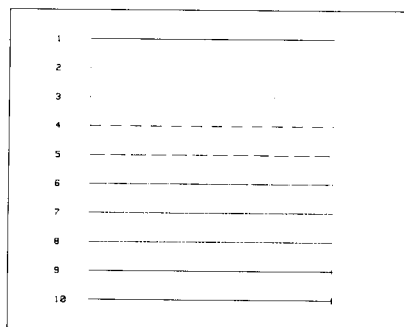
Graphics R/W memory is cleared
Soft clip area = hard clip limits
UDU's = GDU's
UDU's are current units
Pen #1
Line type 1

Characters
Size = 3.3 GDU's
Aspect ratio = 9/15
Slant = 0°
Label origin 1
Label direction horizontal
(left to right)

Pen Control Parameters

Odd = drop pen
Even = lift pen
Positive = pen change after motion
Negative = pen change before motion

Available Line Types



Label Origins

3 6 9
2 T E S T S 8
1 4 7

Graphics Error Messages

ERROR	Description
110	Plotter specifications not recognized.
111	Plotter not previously specified.
112	CRT Graphics hardware not installed.
113	LIMIT specifications out of range.

System 45B Graphics Reference Card

Graphics Syntax Summary

AMES [X tick spacing, Y tick spacing [, X intersection, Y intersection [, X major count, Y major count [, major-tick size]]]]
CLIP [X min, X max, Y min, Y max]
CSIZE height [, aspect ratio [, character slant]]
CURSOR X variable, Y variable [, pen status string variable]
DIGITIZE X variable, Y variable [, pen status string variable]
DRAW X parameter, Y parameter
DUMP GRAPHICS [lower bound[, upper bound]
EXIT GRAPHICS
FRAME
GCLEAR [paper advance]
GLOAD integer array name (*)
GRAPHICS
GRID [X tick spacing, Y tick spacing [, X intersection, Y intersection [, X major count, Y major count [, major-tic size]]]]
GSTORE integer array name (*)
IPLOT X increment, Y increment [, pen control]
LABEL list
LABEL USING image specifier; list
LDIR angle or LDIR run, rise
LETTER
LIMIT [X min, X max, Y min, Y max]
LINE TYPE identifier number [, length]
LOCATE [X min, X max, Y min, Y max]
LORG origin position
MOVE X parameter, Y parameter
MSCALE X offset, Y offset
PDIR angle or PDIR run, rise
PEN pen number
PENUP
PLOT X parameter, Y parameter[, pen control]
PLOTTER IS [select code[, HP-IB device address],
"plotter identifier string" [, step size[, # of pens,
pen offset, [incremental plotter identifier]]]]
PLOTTER select code[, HP-IB device address] IS OFF
PLOTTER select code[, HP-IB device address] IS ON
POINTER X parameter, Y parameter[, cursor type]
RATIO
RPLLOT X parameter, Y parameter[, pen control]
SCALE X min, X max, Y min, Y max
SETGU
SETUU
SHOW X min, X max, Y min, Y max
UNCLIP
WHERE X variable, Y variable [, pen status string variable]

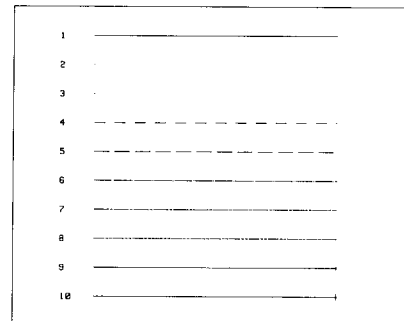
Default Settings

Graphics R/W memory is cleared	Characters
Soft clip area = hard clip limits	Size = 3.3 GDU's
UDU's = GDU's	Aspect ratio = 9/15
UDU's are current units	Slant = 0°
Pen #1	Label origin 1
Line type 1	Label direction horizontal (left to right)

Pen Control Parameters

Odd	=	drop pen
Even	=	lift pen
Positive	=	pen change after motion
Negative	=	pen change before motion

Available Line Types



Label Origins

3 6 9
2 T E S T S 8
1 4 7

Graphics Error Messages

ERROR	Description
110	Plotter specifications not recognized.
111	Plotter not previously specified.
112	CRT Graphics hardware not installed.
113	LIMIT specifications out of range.

Scan Copyright ©
The Museum of HP Calculators
www.hpmuseum.org

Original content used with permission.

Thank you for supporting the Museum of HP
Calculators by purchasing this Scan!

Please to not make copies of this scan or
make it available on file sharing services.