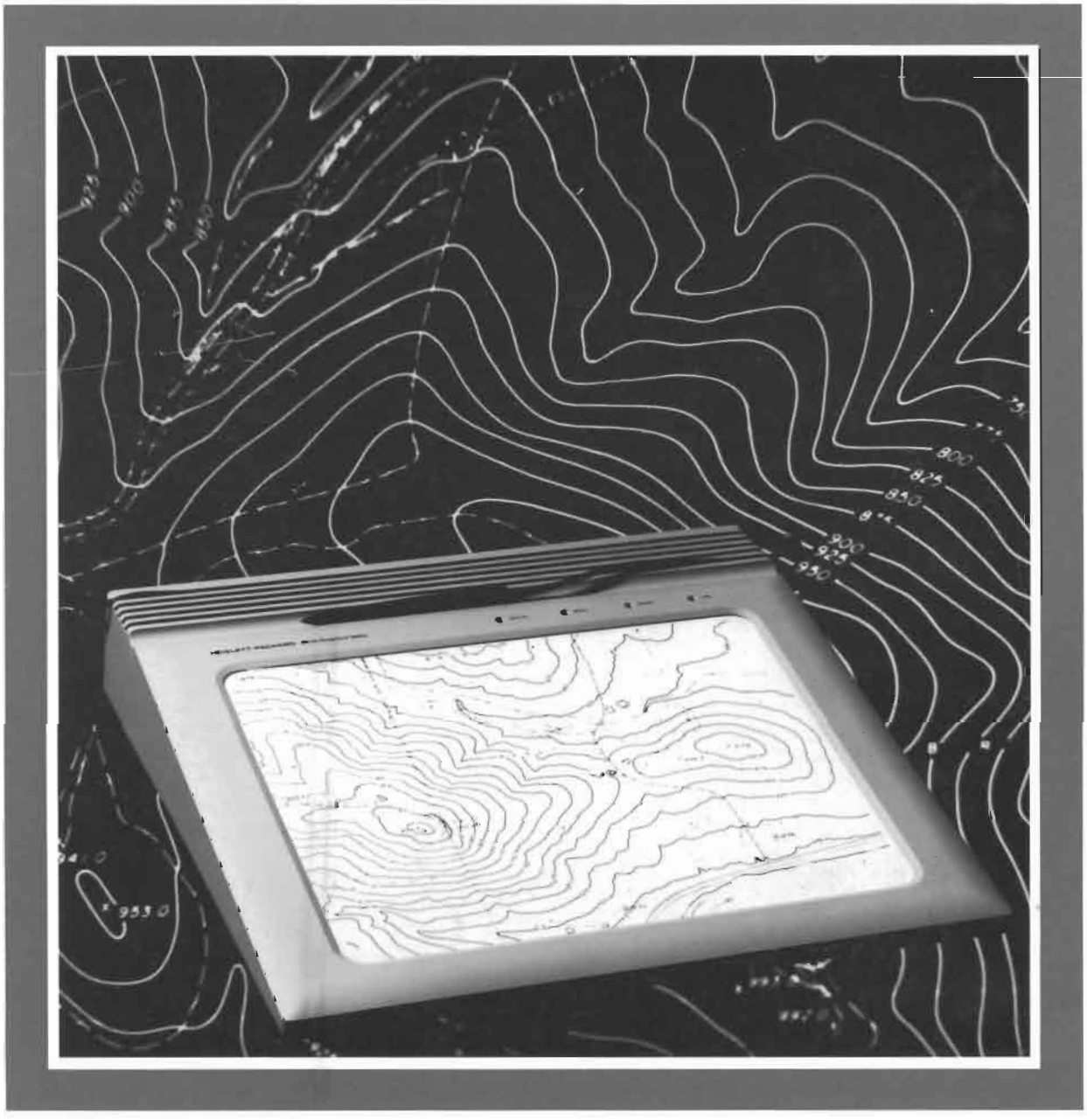
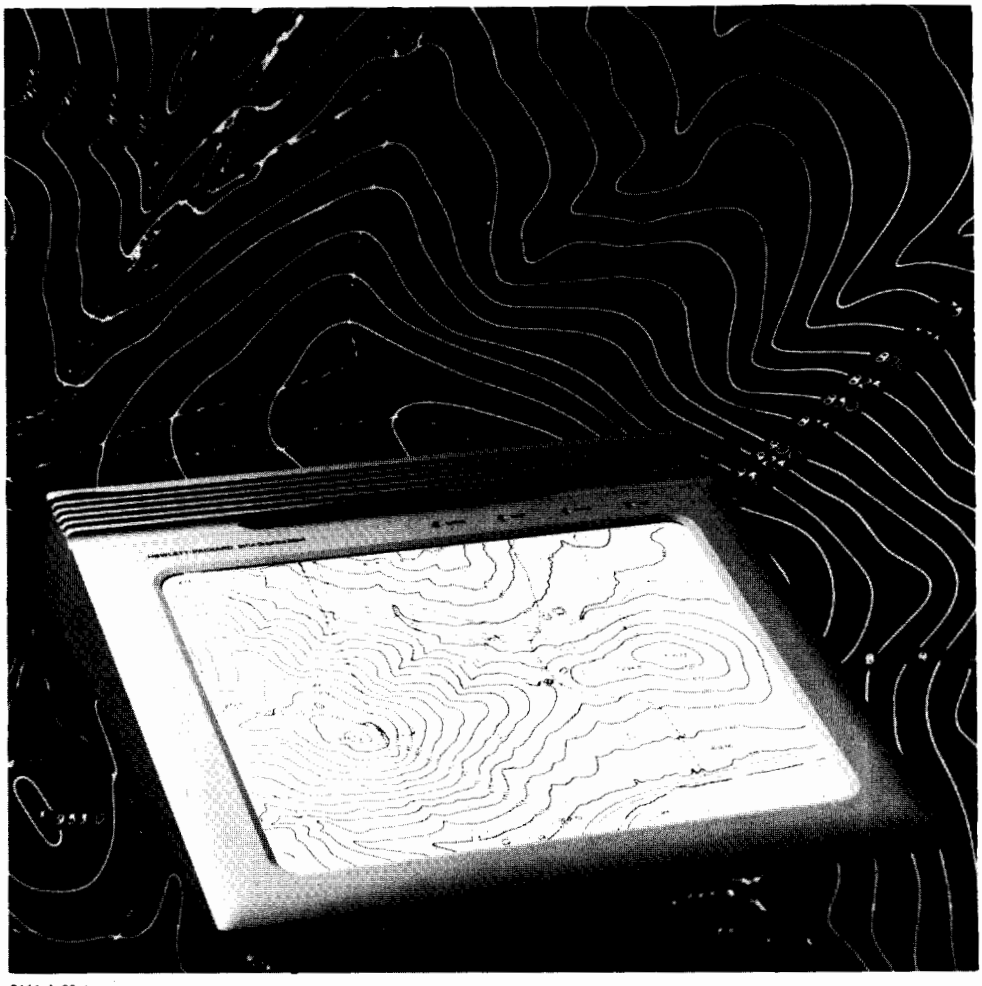
Hewlett-Packard 9111A Graphics Tablet Programming Manual

DESIGNED FOR HP-IB SYSTEMS



Hewlett-Packard 9111A Graphics Tablet Programming Manual

DESIGNED FOR HP-IB SYSTEMS



9111-A-83-1

SERIAL NUMBERS

This manual applies directly to tablets with serial numbers prefixed 2251 and greater.

16399 W. Bernardo Drive, San Diego, California 92127 October 1982

Manual Summary

Chapter 1
Introduction
Contains information concerning manual usage, and a brief description of the graphics tablet and its features.

Chapter 2
Common Graphics Tablet Operations
Discusses operations common to many graphics tablet applications.

Chapter 3
Language Reference
Contains a complete description of each of the 27 Hewlett-Packard Graphics Language (HP-GL) instructions used to program all 9111 tablets. Also included are a description of HP-GL syntax and a table of the instructions.

Chapter 4
HP-IB Interfacing
Summarizes the operation of the graphics tablet with the Hewlett-Packard Interface Bus (HP-IB). Contains examples of sending and receiving ASCII data with various HP computers. Also includes examples of receiving binary default response data.

Chapter 5
The HP 1351S Graphics Display System
Discusses how to use the tablet with the 1351S graphics display system. Includes a complete description of the six instructions designed for use with this system.

Appendix A
An HP-IB Overview
Provides an overview of the Hewlett-Packard Interface Bus (HP-IB).

Appendix B
HP-GL Instruction Summary
Summarizes each of the HP-GL instructions used to program the graphics tablet.

Appendix C
Reference Material
Includes definitions of binary coding and conversions, binary-decimal conversions, two's complement representation, and the tablet's binary default response format. Includes tables of ASCII character codes, tablet default conditions, and error messages. Contains a list of no operation instructions and a diagram of platen digitizing units and dimensions.

Appendix D
Glossary

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Table of Contents

Table of Contents (Continued)

Chapter 3

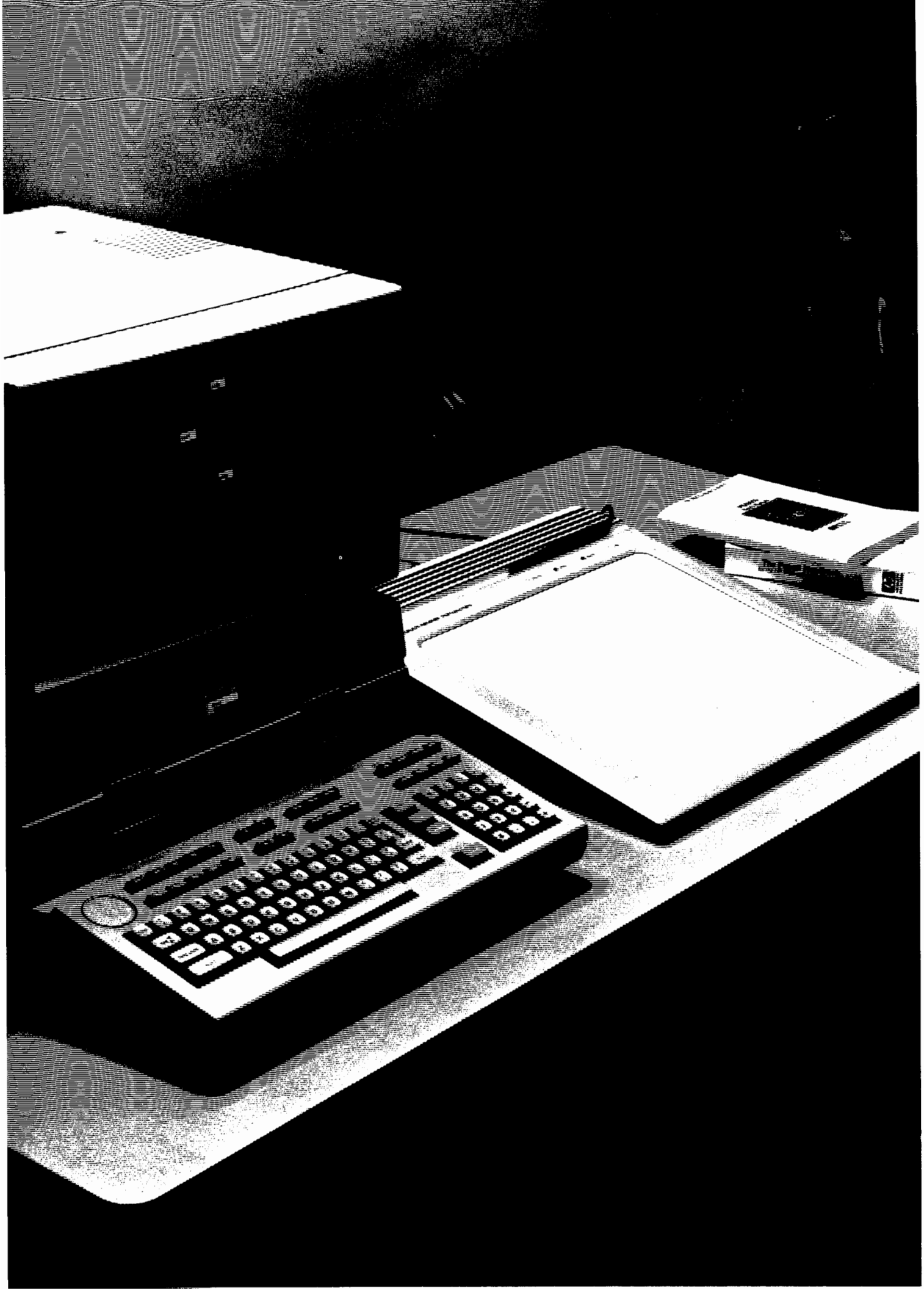
Language Reference (Continued)

Chapter 4

Chapter 5

Table of Contents (Continued)

Table of Contents (Continued)



Chapter 1
Introduction

Chapter 1 tells what is covered in this Programming Manual and what other manuals you may need or find useful. In addition, the chapter contains a brief description of the graphics tablet and its features.

How to Use HP 9111 Documentation

All graphics tablets are shipped with an Operator's Manual (Part No. 09111-90003), this Programming Manual (Part No. 09111-90004), and a Reference Card (Part No. 09111-90007). Depending on the option you ordered, you also may have received a Programming Guide tailored to your computer.

Operator's Manual

The Operator's Manual contains all the information needed to operate, but not program, the graphics tablet.

Programming Manual

This Programming Manual contains the programming and interfacing information for the graphics tablet. The information includes an overview of the tablet and its features; an explanation of common graphics tablet operations; a complete description of the Hewlett-Packard Graphics Language (HP-GL) instructions used to program the tablet; a discussion on using the tablet's Hewlett-Packard Interface Bus (HP-IB) functions; a discussion of how to use the tablet with the HP 1351S graphics display system; and appendices containing an HP-IB overview, a summary of HP-GL instructions, other reference material, and a glossary.

Reference Card

The Reference Card contains a list of the tablet's HP-GL instructions with their parameters, a description of the bits of the tablet's status word, a description of the binary default response format, a list of the tablet's default conditions, and a list of error numbers and their meanings.

Programming Guides

Programming Guides are system tutorials which provide programming examples using a specific HP computer to control basic tablet functions, such as cursor generation and tracking, interactive plotting, digitizing and creating a data file, using menu softkeys, interrupt-driven processing, using audible prompts, using the single and continuous sampling digitizing modes, and using the binary transfer mode.

If you have received a Programming Guide with your option, you may wish to enter and run the sample programs first for a general idea of the tablet's operations before you attempt to read and understand the Programming Manual. If your option does not include a Programming Guide, you may wish to order a Programming Guide for another computer, such as the Series 80 Programming Guide (Part No. 09111-90006).

For Inexperienced 9111 Users

If you have just received your graphics tablet, read the Operator's Manual before attempting to operate the graphics tablet. In addition to telling you how to set up the tablet and connect it to a computer, the manual gives you an understanding of the physical features of the tablet and the ways it can be used. If you are using existing software, the Operator's Manual may be the only manual you need to read.

For Inexperienced HP-GL Programmers

If you are writing your own applications programs for the tablet, and are inexperienced in programming in HP-GL, you will probably find it helpful to read this Programming Manual thoroughly before starting to program the tablet.

For Experienced HP-GL Programmers

If you are writing your own applications programs and are an experienced HP-GL programmer, you may find Appendix B of this manual or the Reference Card most helpful. In the instruction set summary in Appendix B, page numbers for the complete description are listed with each instruction in case you need more details.

Understanding Manual Conventions

Before reading any part of this manual, you should understand the meaning of type styles and number representations used in the text. Words typed in small, boldface type are either switches (**SELF TEST**) or words actually found on the tablet (**MENU**). Numbers are typed using SI (International System of Units) standards to maintain consistency in international publications. Numbers with more than four digits are placed in groups of three, separated by a space, counting both to the left and right of the decimal point (54 321.123 45). Note that the comma used as a decimal marker (European style) or the comma used for the third place marker (U.S. style) are omitted.

Other Relevant Documentation

The actual program statements you use to send data to and from the graphics tablet are highly computer-dependent. For that reason, this manual can only talk about programming concepts in a general way. Your computer's programming manual can help you with the specific statements needed to program the computer.

If your computer has graphics language capabilities to graphics peripheral devices (either through Plotter/Printer ROMs or Graphics ROMs), the graphics ROM manual can help you with any high-level Advanced Graphics Language (AGL) statements that you can use with the graphics tablet.

If your computer has an Input/Output ROM or equivalent read/write capabilities to external devices, the I/O programming manual is the best source of information about your computer's input/output operations and statements and about how to control the HP-IB Interface. It is essential that you understand your computer's input/output operations before attempting to program tablet applications.

A Brief Look at the 9111 Graphics Tablet

The HP 9111 Graphics Tablet is a data entry device that converts a physical location on the tablet's surface into digital values usable by a computer. When the stylus is in proximity to (within approximately ¼ inch) or touching the tablet's surface, the tablet records the X,Y coordinates of that location. The tablet continuously updates position information as long as the stylus remains close enough to the tablet's surface. You can program the rate at which position information is updated (refer to "The Cursor Rate Instruction, CR" in Chapter 3).

You can set either a single-point or a continuous-sampling digitizing mode. These modes are explained in detail in "Digitizing Modes" in Chapter 2. The digitizing modes allow you to control the output of position information but do not stop the continual updating process that occurs as long as the stylus is in proximity to the platen.

You can use one of two data formats to transfer data to and from the tablet: the default binary format or ASCII code. If you only send a computer-dependent read statement, such as READ, INPUT, or ENTER, to the tablet, you will receive data in the default binary format. The default binary format consists of three 16-bit words. If you precede a read statement with an HP-GL (Hewlett-Packard Graphics Language) output instruction, you will receive data in 8-bit ASCII code. Chapter 4, HP-IB Interfacing, explains the default binary and ASCII formats in greater detail.

In addition to the X- and Y-coordinates of the last known stylus position, the tablet can send the following information to the computer: whether the pen is pressed, the number of a selected softkey, the tablet's current status, and the number of the last error, if any.

Programming Features

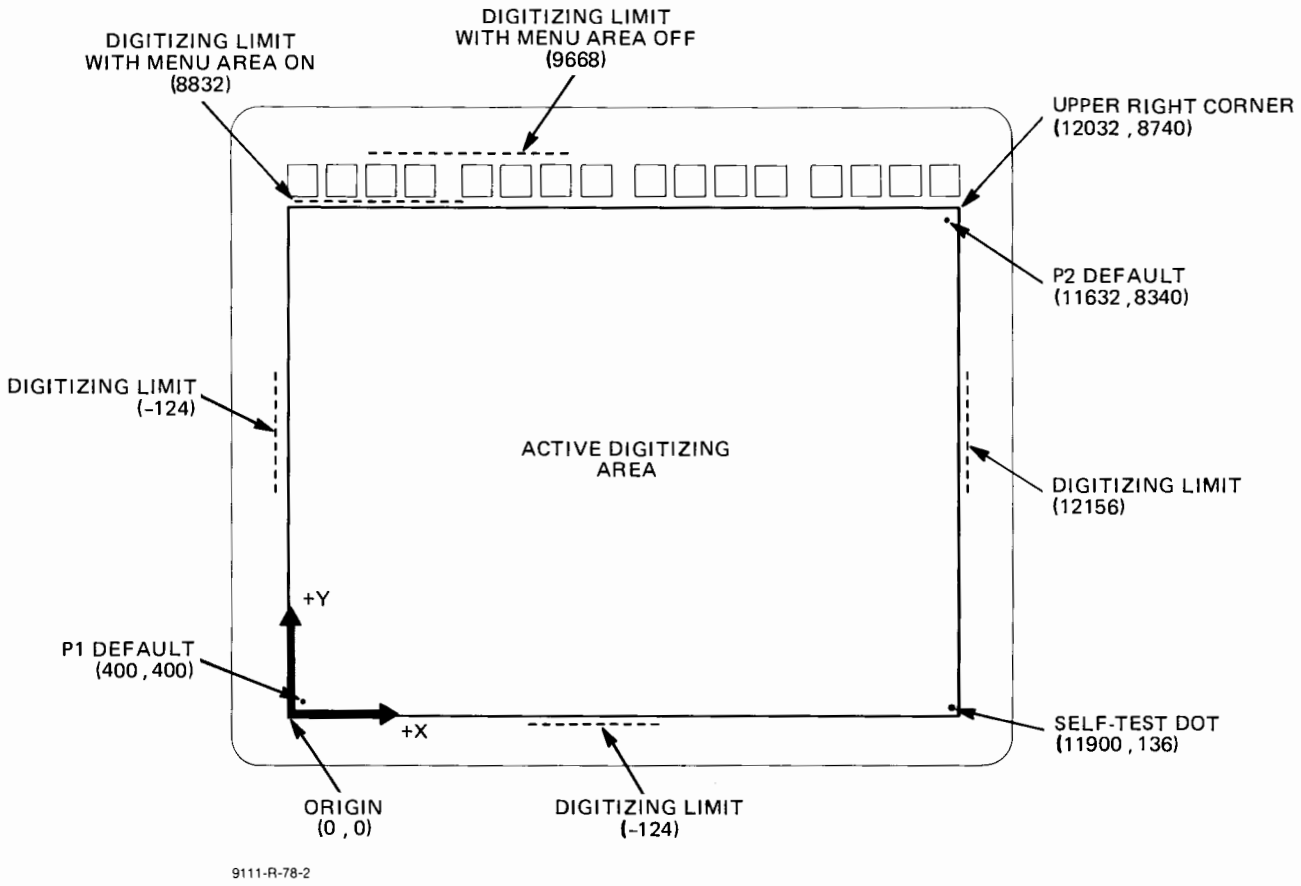
The major programming features of Hewlett-Packard's 9111 are:

- 27 HP-GL (Hewlett-Packard Graphics Language) instructions to provide programming precision and flexibility for all tablet users;
- a programmable data rate of up to 60 points per second to help you achieve fast, smooth cursor tracking;
- a choice between an ASCII data format for simple programming or a binary data format to improve system performance;
- audio feedback programmable in tone, volume, and duration;
- sixteen softkeys, conveniently located across the top of the tablet, to make program control easier by sending a simple, unique code to your computer program; and
- two digitizing modes: single point or continuous sampling.

Physical Description

Refer to the HP 9111 Operator's Manual for a complete description of the physical features of the graphics tablet. Appendix C of this manual contains a detailed diagram showing significant tablet measurements in digitizing units, millimetres, and inches. Following is a basic diagram of the platen boundaries and digitizing limits in digitizing units (DU's).

Platen Boundaries and Digitizing Limits in Digitizing Units



Applications

The three major application areas for the tablet are cursor tracking for interactive graphics design, digitizing existing graphics data, and using a menu to select program commands or other information.

SIGGRAPH Core Input Methods Supported

The graphics tablet can be used for various methods of input that are supported by graphics software packages such as Hewlett-Packard's Graphics/1000-II Advanced Graphics Package 3D (1000-II AGP-3). A discussion follows of input methods that are defined for the SIGGRAPH Core System in the "Status Report of the Graphic Standards Planning Committee," August 1979, and are supported by the 9111 graphics tablet.

The term "device" as used in the SIGGRAPH Core System refers to a method of input or output that is implemented by a physical device. The first set of methods are considered event devices. Event devices are used by the operator to signal an event to the application program.

Button — Returns an integer value depending on the key or button actuated, providing the ability to choose among alternatives. The softkeys on the 9111 perform this function.

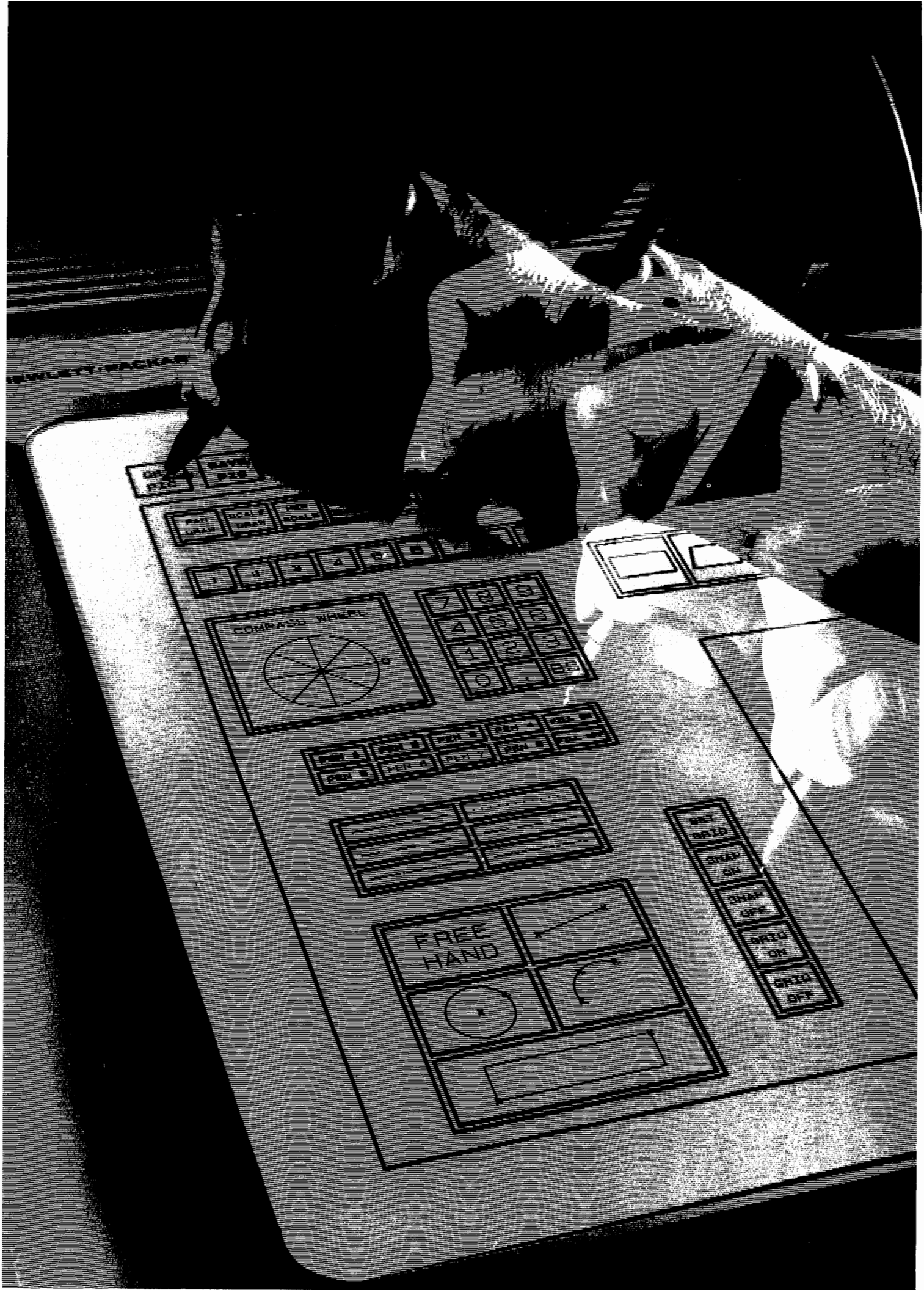
Pick — Returns name and pick identifier of a segment and a primitive within a segment as designated by an operator using devices, such as a terminal cursor or digitizer. For example, the operator can point at the primitives of line, set of lines, or text on the display and have the segment name and pick identifier returned to the application program. Without picking, only the X,Y coordinates are returned to the application program, which then has to decide what the operator is pointing at. The 9111 tablet can be used by an operator to pick segments using a software package such as Graphics/1000-II AGP-3.

Stroke — Provides a series of X,Y coordinates. Returns the sequence of positions composing the stroke and the number of positions in the stroke. Commonly, a stroke begins when the pen switch is pressed. Points are recorded at some rate as long as the switch is closed. The stroke is terminated when the switch opens. The 9111's continuous-sampling mode can provide the stroke function. The stroke function is not supported by the Graphics/1000-II AGP-3 software. As of this manual's publication date, the available software cannot detect all the input points in a stroke when multiple devices are on line.

The next set of functions are considered sampled devices. Sampled devices have values which may be sampled by the application program.

Locator — Returns X,Y coordinate information designated by an operator using a terminal cursor or digitizer. The tablet's stylus functions as a locator.

Valuator — Returns the scalar real value entered by the operator as the relative position of a terminal cursor or digitizer, or the setting of a knob. The value is always in the range of 0 to 1. The tablet's stylus can be used to return X,Y coordinates that can be scaled by an application program.


HEWLETT-PACKARD
COMPASS WHEEL
FREE HAND
SNAP ON
SNAP OFF
GRID ON
GRID OFF

Chapter 2
Common Graphics Tablet Operations

This chapter discusses the following operations that are common to many graphics tablet applications:

- digitizing P1 and P2;
- scaling;
- digitizing modes;
- cursor tracking with picks; and
- interpreting areas on the tablet.

The descriptions of tablet operations in Chapter 2 use specific HP-GL (Hewlett-Packard Graphics Language) instructions. An HP-GL instruction is a two-letter mnemonic that may or may not be followed by one or more numerical parameters. A mnemonic is a code that assists your memory. For example, DF suggests default. The instructions referred to in this chapter are explained in detail in Chapter 3, Language Reference, in this manual.

Many references are made in this chapter to checking bits of the tablet's status word to see whether or not they are set. "The Output Status Instruction, OS" in Chapter 3 contains a table of the eleven bits of the status word and their meanings. A set bit has a value of 1; a bit that is not set has a value of 0.

The input and output statements used to send the HP-GL instructions to the tablet and read data from the tablet are highly computer-dependent. Therefore, tablet operations are discussed in a general way in this manual. A recommended sequence of steps is given for each operation. Programming guides tailored to specific computers give actual program examples for tablet operations.

You might find it useful to skim through this chapter for a general understanding of the concepts, then study the HP-GL instructions in Chapter 3, and then refer back to Chapter 2 for a more thorough understanding of the details.

Digitizing P1 and P2

P1 and P2 are two points on diagonally opposite corners that define a rectangular digitizing area. On the 9111, P1 represents the X,Y coordinates of the lower-left corner of the platen area to be scaled, and P2 represents the X,Y coordinates of the upper-right corner. The default value of P1 is 400, 400 and P2 is 11 632, 8340. The diagrams of the tablet's surface in the

Introduction and Appendix C show the default locations of P1 and P2 and the tablet's digitizing limits. You can digitize anywhere within the tablet's digitizing limits regardless of the location of P1 and P2.

You may wish to change the locations of P1 and P2 by digitizing two points. For example, you might require a specific scale or height/width proportion. This discussion shows you how to digitize two points and then return them to the tablet as P1 and P2. The next discussion explains how to use P1 and P2 for scaling digitizing units into user units.

A recommended sequence for digitizing P1 and P2 follows:

- Use either the initialize instruction, IN, or the default instruction, DF, to set the tablet to known conditions.
- Set the single-point mode using the single-sample mode instruction, SG.
- Prompt the operator to digitize P1.
- Send the output status instruction, OS, to the graphics tablet.
- Read the tablet's status word.
- Check bit 2 of the tablet's status word. (When a point is digitized, bit 2 is set.)
- If bit 2 is not set, keep checking bit 2.
- If bit 2 is set, send the output digitized point instruction, OD, to the tablet, and then read the X,Y coordinates as X_1 and Y_1.
- Prompt the operator to digitize P2.
- Send the OS instruction to the tablet, and read the tablet's status word.
- Check bit 2 of the tablet's status word.
- If bit 2 is not set, keep checking bit 2.
- If bit 2 is set, send the OD instruction to the tablet, and then read the X,Y coordinates as X_2 and Y_2.
- Send the digitizer clear instruction, DC, to the tablet to clear the digitizing mode and last digitized point.
- Send the input points instruction, IP, to the tablet with X_1,Y_1,X_2,Y_2 as parameters. For example, the last two steps could be combined into one instruction as follows for the HP-85, HP-87, and HP 9845 computers:

```
PRINT "DC;IP";X1;",";Y1;",";X2;",";Y2
```

The commas used as separators must be enclosed in quotation marks to be sent to the tablet on the computers mentioned in this example.

The same two instructions could be sent to the tablet as follows for the HP 9826 and HP 9836 computers:

```
PRINT "DC;IP";X1,Y1,X2,Y2
```

For the computers in this example, the commas are not enclosed in quotation marks when sent to the tablet.

Note that in both cases above, PRINT is the output statement that is used. Other typical output statements are WRITE and OUTPUT. Refer to your computer's documentation for the output statement to use and your computer's format for sending parameters that are variables.

Scaling Digitizing Units into User Units

There are two unit systems that can be used to measure point locations in the digitizing area: digitizing units and user units. The digitizing area is divided into digitizing units (DU's). One digitizing unit is 0.025 mm (approximately 0.001 in.) in length. The digitizing area can be scaled into user units. You can define user units to suit a particular application, and they can represent such measurements as months or units sold.

The preceding discussion showed how to set the scaling points P1 and P2 by digitizing. This discussion shows how to use P1 and P2 to scale the tablet's digitizing units into user units. The diagram of "Platen Boundaries and Digitizing Limits" in Chapter 1 shows the locations of P1 and P2, the tablet's origin point, and the tablet's digitizing limits, which are all referred to in the following discussion.

The formulas used in this scaling operation are:

$$X' = aX + b$$
$$Y' = cY + d$$

where X' = the user-unit X-coordinate
Y' = the user-unit Y-coordinate
X = the digitizing-unit X-coordinate
Y = the digitizing-unit Y-coordinate
a = the X scaling factor
c = the Y scaling factor
b = the X offset factor
d = the Y offset factor

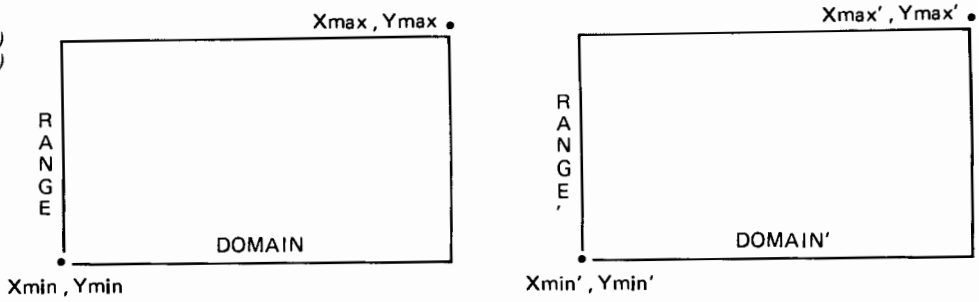
The scaling factors are used to convert the graphics tablet's coordinate system in digitizing units into a coordinate system of user units. The offset factors are needed if you start at an origin other than point 0,0 which is located in the lower-left corner of the platen. The limits of the graphics tablet coordinate system are found by reading P1 and P2 from the graphics tablet. Remember that the values of P1 and P2 can be one of the following:

- the default values of P1 = 400,400 and P2 = 11632,8340. Thus X_{min} = 400, Y_{min} = 400, X_{max} = 11632, and Y_{max} = 8340.
- set by sending a specific value to the tablet with the input points instruction, IP.
- set by digitizing two points that you wish to represent P1 and P2, and then sending those points to the tablet as P1 and P2 using the input points instruction, IP. The preceding discussion shows how to do this.

The range of values for P1 and P2 correspond to the digitizing limits of the graphics tablet. Before you convert digitizing units into user units, you also must know the values of X_{min}, Y_{min}, X_{max}, and Y_{max} in your user-unit coordinate system.

The following diagrams show that the relationships of the values you use to calculate the various elements of the scaling formulas are equivalent.

Digitizing Units (left)
User Units (right)



Using X_{min}, Y_{min}, X_{max}, Y_{max} for the graphics tablet domain and range, and X_{min}', Y_{min}', X_{max}', Y_{max}' for the user-unit domain and range, the following sequence of operations produces a, b, c, and d.

- Send the output points instruction, OP, to the tablet to obtain the current values of P1 and P2.
- Read the values as X_{min}, Y_{min}, X_{max}, Y_{max}.
- Perform the following calculations:

$$\text{Domain} = X_{max} - X_{min}$$
$$\text{Range} = Y_{max} - Y_{min}$$

$$\text{Domain}' = X_{max}' - X_{min}'$$
$$\text{Range}' = Y_{max}' - Y_{min}'$$

$$a = \text{Domain}'/\text{Domain}$$
$$c = \text{Range}'/\text{Range}$$

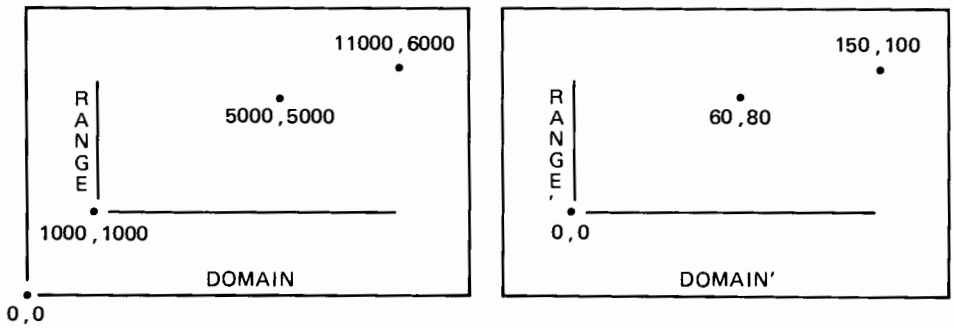
$$b = X_{min}' - X_{min} \times a$$
$$d = Y_{min}' - Y_{min} \times c$$

The scaling and offset factors need to be determined each time user units are changed. Once these factors are defined, you can place the values of P1 and P2 from the graphics tablet into the scaling formulas given in the beginning of this discussion to convert the digitizing units from the tablet into your user units.

Scaling Example

Assume that you have previously sent an IP instruction to the tablet to establish the values of P1 and P2 as 1000,1000 and 11000,6000 respectively. Thus X_{min} = 1000, Y_{min} = 1000, X_{max} = 11000, Y_{max} = 6000. Assume also that you want to scale that area to user units of 0 to 150 in the X-axis and 0 to 100 in the Y-axis. Thus X_{min}' = 0, Y_{min}' = 0, X_{max}' = 150, Y_{max}' = 100. Refer to the following figures.

Digitizing Units (left)
User Units (right)



By inserting the assumed values into the formulas shown earlier, we get the following results:

Domain = 11 000 − 1000 = 10 000
Range = 6000 − 1000 = 5000

Domain′ = 150 − 0 = 150
Range′ = 100 − 0 = 100

a = 150/10 000 = 0.015
c = 100/5000 = 0.02

b = 0 − 1000 × 0.015 = −15
d = 0 − 1000 × 0.02 = −20

Thus if we digitize a point on the tablet with an X,Y coordinate of 5000,5000, we can convert the coordinate to user units as follows:

X′ = 0.015 × 5000 − 15 = 60
Y′ = 0.02 × 5000 − 20 = 80

The point we digitized would have an X,Y coordinate of 60,80 in our user-unit coordinate system as shown in the preceding figures.

Digitizing Modes

There are two modes of digitizing available on the graphics tablet: single point and continuous sampling. This discussion shows how to set both modes. Note that the digitizer clear instruction, DC, clears the digitizing modes.

Single-Point Digitizing

Single-point digitizing is used to digitize one point at a time, such as digitizing points on strip charts or picking items from a menu. The two instructions that implement single-point digitizing are the digitize point instruction, DP, and the single-sample mode instruction, SG.

The DP instruction is primarily included for compatibilty with Hewlett-Packard plotters, and the mode is cleared as soon as a single point is digitized. The SG instruction is designed to alternate with the continuous-sampling mode instruction, CN, described below, and the mode remains in effect until cleared or changed. If you are concerned about plotter compatibility, you may want to use the DP instruction. Otherwise, it is recommended that you use the SG instruction for more efficient programming. Note that the DP instruction is ignored if either the SG or CN instructions are in effect.

A recommended sequence for single-point digitizing follows:

- Use either the initialize instruction, IN, or the default instruction, DF, to set the tablet to known conditions.
- Set the single-sample mode using the single-sample mode instruction, SG.
- Send the output status instruction, OS, to the graphics tablet.
- Read the tablet's status word.
- Check bit 2 of the tablet's status word. (When a point is digitized, bit 2 is set.)

- If bit 2 is not set, keep checking bit 2.
- If bit 2 is set, send the output digitized point instruction, OD, to the tablet, and then read the X,Y coordinates.
- Update your data file with the digitized point.

Continuous Digitizing

Continuous digitizing is used to draw or follow line contours with the tablet's stylus. The continuous-sampling mode instruction, CN, implements continuous digitizing. There are two types of continuous digitizing: switch normal and switch follow. Both types generate streams of data points which must be read individually using the output digitized point instruction, OD.

In the switch-normal mode, the first pen press is interpreted as a "begin digitizing" signal and the second pen press is treated as an "end digitizing" signal. In this mode, the tablet records points as long as the stylus is within approximately ¼ inch of the tablet's surface. When the CN instruction is given, the switch-normal mode is automatically in effect. The switch normal instruction, SN, is used to set the switch-normal mode. Even though it is not necessary to use the SN instruction with the CN instruction when you want the switch-normal mode, it can make your program easier for others to interpret. You must use the SN instruction when you want to change from the switch-follow mode.

In the switch-follow mode, the points are digitized while the pen is pressed (the stylus digitizing switch is open). In this mode, digitizing stops when the pen is lifted. The switch-follow mode is set with the switch follow instruction, SF.

In both modes, the last point sent by an output digitized point instruction, OD, has a pen status of 0. All other points in the continuous-sampling mode have a pen status of 1. The continuous-sampling mode remains in effect until cleared or changed.

A recommended sequence for continuous digitizing follows:

- Use either the initialize instruction, IN, or the default instruction, DF, to set the tablet to known conditions.
- Set the continous-sampling mode using the continuous-sampling mode instruction, CN.
- Set either the switch-normal mode (switch normal instruction, SN) or the switch-follow mode (switch follow instruction, SF). The switch-normal mode is the default continuous-sampling mode.
- Send the output status instruction, OS, to the graphics tablet.
- Read the tablet's status word.
- Check bit 2 of the tablet's status word. (When a point is digitized, bit 2 is set.)
- If bit 2 is not set, keep checking bit 2.
- If bit 2 is set, send the output digitized point instruction, OD, to the tablet, and then read the X,Y coordinates and pen status.

- If the pen status is 1 (all points except the last), add another point in the line to your data file.
- If the pen status is 0 (the last point), terminate the line currently being recorded and set the data file to read the next point as the beginning of a new line.

Cursor Tracking with Picks

It is quite common to use an indicator, or cursor, on a display screen to track the position of the stylus on the tablet's surface. It is also common for an operator to press the stylus on a specific location on a menu to "pick" a menu item. Many cursor-tracking applications control program input by recognizing that the stylus has been pressed in a certain area on a menu.

A recommended sequence for cursor tracking with picks follows.

- Define the CRT area in coordinates that match the tablet.
- Use either the initialize instruction, IN, or the default instruction, DF, to set the tablet to known conditions.
- Set the single-sample mode using the single-sample mode instruction, SG.
- Send the output cursor instruction, OC, to the tablet.
- Read the X,Y coordinates, pen status, softkey value, status word, and error number from the tablet.
- Move the cursor on the CRT to match the new stylus position.
- Check bit 10 of the tablet's status word. (When the pen is pressed, bit 10 is set.)
- If bit 10 is not set, keep checking bit 10.
- If bit 10 is set, check bit 2 of the status word. (When a point is digitized, bit 2 is set.)
- If bit 2 is set, send the output digitized point instruction, OD, to the tablet; read the X,Y coordinates of the digitized point; and go to a routine that interprets the digitized point by calculating its location. Refer to "Interpreting Areas on the Tablet" in this chapter for details about calculating locations.
- If bit 2 is not set, check bit 7 of the status word. (Bit 7 is set when a softkey is selected.)
- In the unlikely event that bit 7 is not set, there is probably some type of hardware error, and you need to perform an error routine.
- If bit 7 is set, send the read softkey instruction, RS, to the tablet; read the softkey value; and perform a routine depending on which softkey is selected.

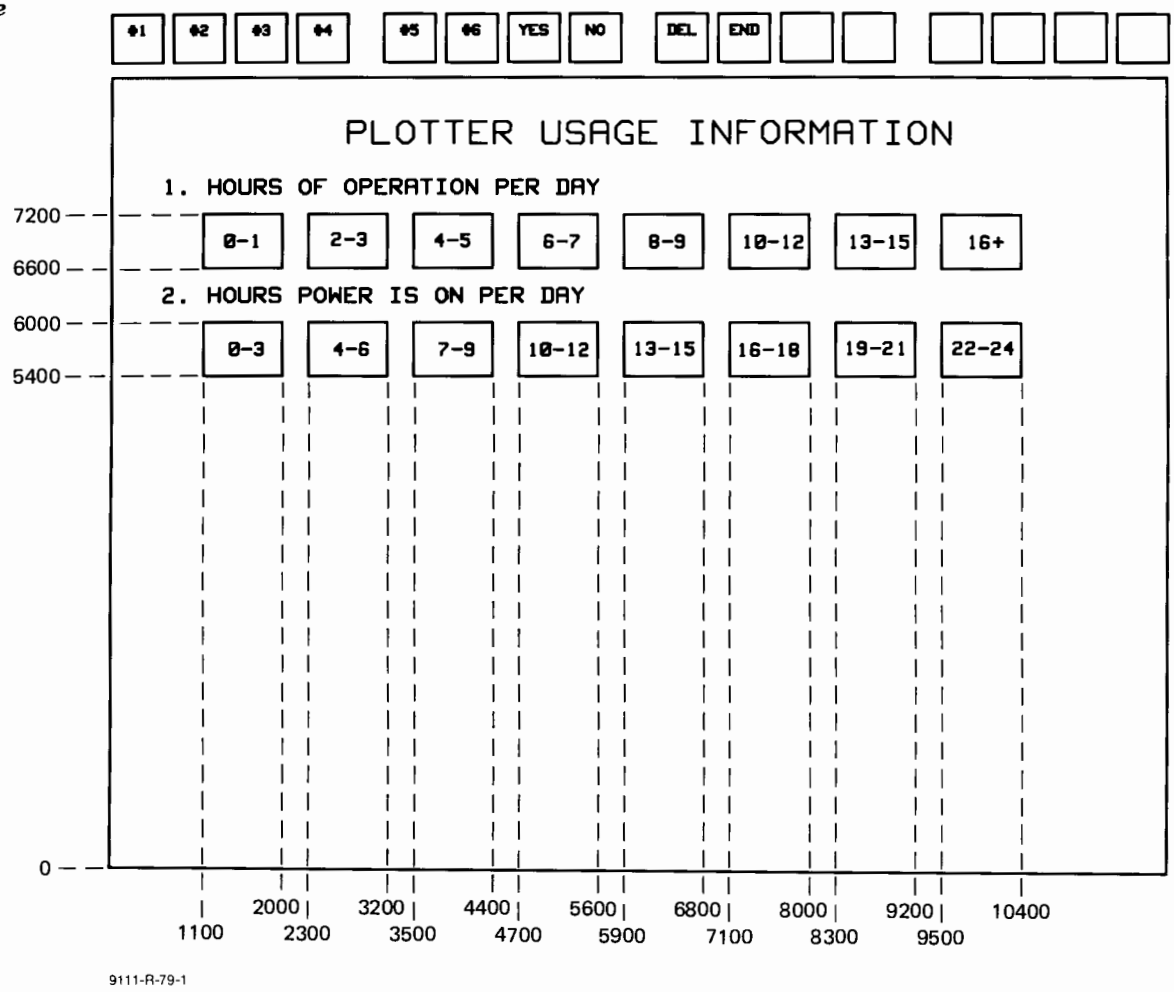
Note that bit 2 is checked before bit 7 in the routine above. That sequence is important because a softkey selection overrides the digitized-point-available response.

Interpreting Areas on the Tablet

In addition to using the softkeys on the tablet's surface to pick menu items, you can also interpret sections of the tablet's active digitizing area as menu items.

Following are two rows of boxes that could be used as menu items on the the tablet's active digitizing area. The X- and Y-axes are marked in digitizing units to show where the boxes are located.

Menu Example



A recommended sequence for interpreting areas on the tablet follows. This sequence assumes that (1) the pen has been pressed, (2) the output digitized point instruction, OD, has been sent to the 9111, and (3) the computer has read the X,Y coordinates of the point. The program can then call an "Interpret Menu" subroutine which compares the X,Y values and takes the desired action.

Interpret Menu

- If Y > 7200 go to an error routine indicating that the pen has been pressed in an invalid area. Otherwise, go on to the next comparison.
- If Y < 7200 or Y > 6600 go to the Row 1 subroutine. Otherwise, go on . . .

- If Y < 6000 or Y > 5400 go to the Row 2 subroutine. Otherwise, go to the error routine.

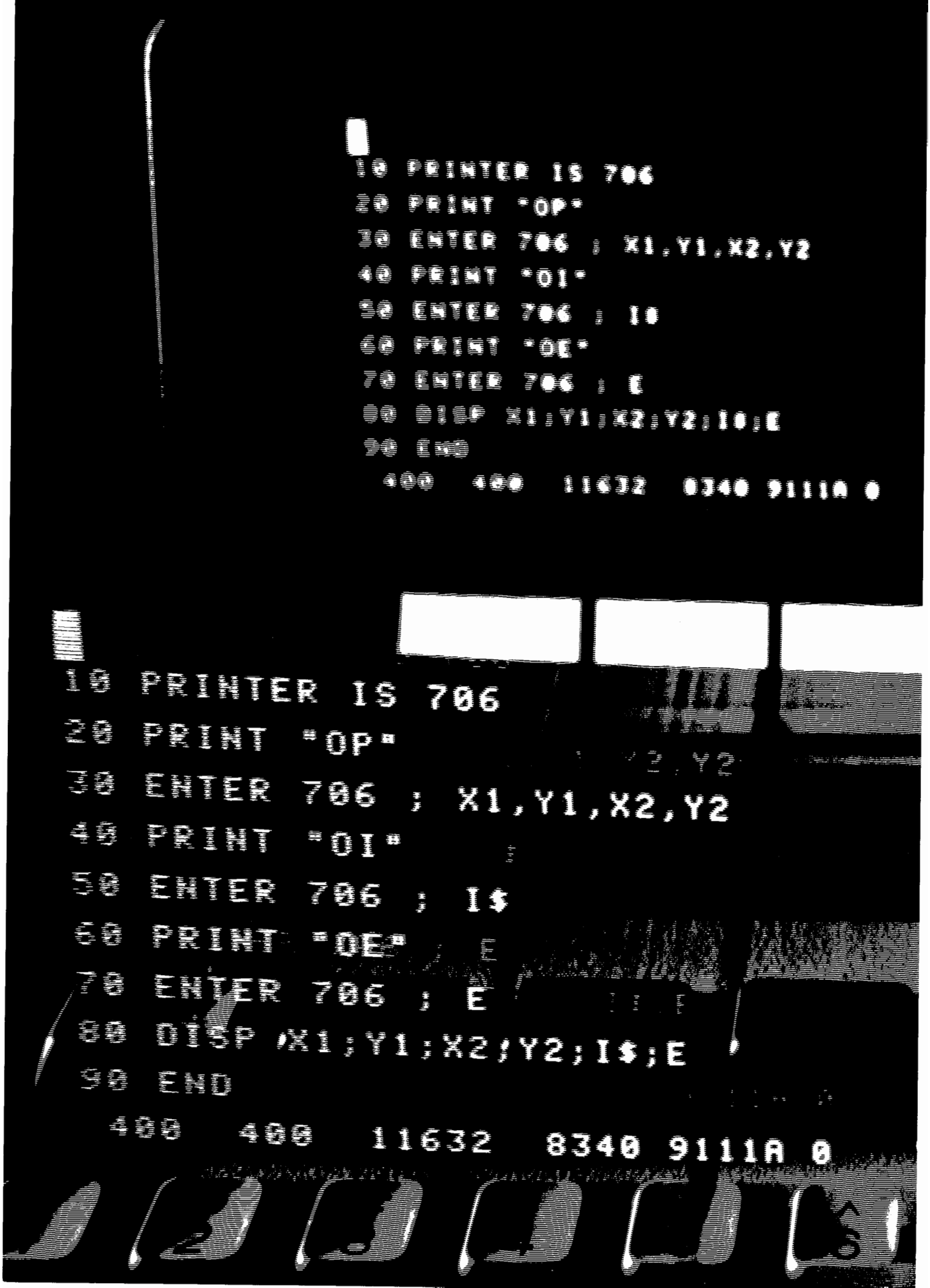
Row 1

- If X < 1100 go to the error routine. Otherwise, go on to the next comparison.
- If X > 1100 or X < 2000 go to the Box 1 subroutine. Otherwise, go on . . .
- If X > 2300 or X < 3200 go to the Box 2 subroutine. Otherwise, go on . . .
- If X > 3500 or X < 4400 go to the Box 3 subroutine. Otherwise, go on . . .
- If X > 4700 or X < 5600 go to the Box 4 subroutine. Otherwise, go on . . .
- If X > 5900 or X < 6800 go to the Box 5 subroutine. Otherwise, go on . . .
- If X > 7100 or X < 8000 go to the Box 6 subroutine. Otherwise, go on . . .
- If X > 8300 or X < 9200 go to the Box 7 subroutine. Otherwise, go on . . .
- If X > 9500 or X < 10 400 go to the Box 8 subroutine. Otherwise, go to the error routine.

Row 2

Same as Row 1.

Note that it is very important to tell the computer what to do if a point is digitized in an undefined area. If you don't, the program stops and cannot be resumed, and the operator may not know what happened or what to do.


10 PRINTER IS 706
20 PRINT "OP"
30 ENTER 706 ; X1,Y1,X2,Y2
40 PRINT "OI"
50 ENTER 706 ; I$
60 PRINT "OE"
70 ENTER 706 ; E
80 DISP X1;Y1;X2;Y2;I$;E
90 END
400 400 11632 8340 9111A 0

Chapter 3
Language Reference

The 9111 Graphics Tablet's Instruction Set

The instruction set that can be used for all 9111 graphics tablets consists of 27 Hewlett-Packard Graphics Language (HP-GL) instructions. This chapter contains a table listing the 27 instructions followed by complete details about each instruction. Included is information about compatibility with other HP graphics devices such as the 9874 digitizer and various plotters.

The instruction descriptions are divided into the following subheadings: DEFINITION, USES, SYNTAX, DEFAULT, OUTPUT, and EXPLANATION. As you read this chapter, keep in mind that the first five subheadings contain concise information intended for quick reference and that all the details can be found under EXPLANATION. Read the EXPLANATION sections for answers to questions that may occur to you in the earlier sections.

Six additional instructions are available for 9111 tablets used with the 1351S graphics display system. Chapter 5 discusses these six instructions and how to use them with the graphics display system.

The HP-IB Interfacing chapter contains general information about the tablet's ASCII and default binary data messages and shows examples of sending ASCII data and receiving both ASCII and default binary data with a variety of computers. Appendix C includes a summary of each instruction. There are 18 additional HP-GL instructions which cause no operation but are included for compatibility with HP plotters and digitizers. These instructions are listed in Appendix C.

HP-GL Syntax

An HP-GL instruction is a two-letter mnemonic, which can be either upper- or lowercase, may be followed by one or more numerical parameters, and ends with a terminator. Spaces and carriage returns (CR) within the instructions are ignored.

Parameters and Separators

Some HP-GL instructions have no parameters and some allow one or more parameters. If more than one parameter is used with an instruction, the parameters must be separated with a comma.

If an instruction has multiple parameters, you can change one parameter without including the parameters that are to remain the same. The graphics tablet allows you to specify just the parameter you want to change by positioning it with commas.

For example, assume that you have previously specified the parameters of the input points instruction, IP, to be as follows:

```
IP 600,600,11000,8000
```

Later you want to change the third parameter from 11 000 to 10 000. This is done as follows:

```
IP,,10000
```

If you send

```
IP,,,
```

all of the current values remain unchanged.

Terminators

An HP-GL instruction, sent to the graphics tablet, can be terminated in one of three ways: line feed (LF), semicolon (;), or asserting EOI. Each type of terminator is explained here. Note that you can use any, but only one, of these terminators for a given instruction. The tablet does not accept more than one terminator per instruction.

1. Line feed (LF) — When the tablet receives a data string followed by a line feed (ASCII decimal 10), the data is interpreted as a complete instruction (two-letter mnemonic with any allowable parameters). Any additional data characters received by the graphics tablet are interpreted as another instruction.

   Many Hewlett-Packard computers automatically generate a carriage return (CR) and line feed (LF) at the end of a line. The tablet ignores the carriage return. If your computer does automatically generate a line feed at the end of a line, that is the only terminator allowed for the last instruction on the line. If your computer does not automatically generate a line feed, you must specify a terminator. If you should suppress the automatic line feed in your output statement for any reason, be sure to include one of the other terminators. Refer to your computer's programming manual for details about automatic line feeds.

2. Semicolon (;) — When the graphics tablet receives a data string followed by a semicolon (ASCII decimal 59), the data is interpreted as a complete instruction (two-letter mnemonic with any allowable parameters). Any additional data characters received by the graphics tablet are interpreted as another instruction. The semicolon is available on terminal keyboards and must be typed in along with the instruction.

   On computers that automatically generate a line feed (LF) at the end of a line, you would only use the semicolon (;) as a terminator to separate several instructions on one line. You would *not* use the semicolon for the last instruction on a line because you would then have two terminators for one instruction. If you send two terminators with one instruction to the tablet, you get unwanted results and the tablet may or may not recognize an error condition.

3. Asserting EOI — The tablet allows the HP-IB EOI (End or Identify) signal line to terminate instructions. If EOI is set true prior to the graphics tablet receiving the last character in a data string, the last character serves its initial function (mnemonic or parameter) as well as acting as a terminator.

Most Hewlett-Packard computers do *not* automatically set EOI true at the end of a data message. However, many HP computers do allow statements that enable you to set EOI true. The HP-IB END command is one way to do this. Refer to your computer's I/O documentation for more information.

Conventions Used to Represent Syntax

The syntax shown under the description of each HP-GL instruction uses the following conventions:

*MN*emonic — The required mnemonic appears in uppercase italics. The two letters can be upper- or lowercase but otherwise must be entered as shown.

parameters — Parameters are indicated by words representing numerical values.

[ ] — All items in square brackets are optional.

terminator — The required terminator can be a line feed (LF), a semicolon (;), or EOI asserted on the last character of the instruction string. Note that the graphics tablet accepts only one terminator per instruction.

Note that the tablet ignores spaces and carriage returns (CR) within an instruction.

Using HP-GL Instructions

You can include HP-GL instructions as part of a program subroutine. Many HP-GL instructions can also be sent directly to the tablet for immediate execution. For example, you might want to send the output error instruction, OE, to identify an error that has caused your program to halt.

The HP-GL instructions are usually sent as a literal string. The instruction strings must be combined with a computer-dependent output statement, such as PRINT, WRITE, or OUTPUT, to be sent to the tablet. Each computer has its own system format conventions relating to instruction strings. An example of sending the OE instruction to the tablet using the HP 9826 computer follows:

```
OUTPUT 706;"OE"
```

Chapter 4, HP-IB Interfacing, contains examples of sending HP-GL instructions for a variety of HP computers. Your computer's documentation contains more detailed information about the output statement to use and how to format the instruction string. Pay particular attention to the effect of quotation marks, commas, and semicolons in sending an instruction string. For example, literal strings are usually sent within quotation marks, commas often specify a free-field format causing extra spaces to be sent, and semicolons often eliminate the extra spaces. Also note how to send variables; they are usually not enclosed in quotation marks. Refer to the IP instruction in Chapter 4 for specific examples of sending an instruction with variables as parameters.

Instruction		Description
BP	[frequency],[duration],[amplitude]	Generates the internal tone.
CN		Sets the continuous-sampling mode.
CR	[points per second]	Specifies the stylus position update rate.
DC		Clears any digitized point and any digitizing mode set by the DP, SG, or CN instructions.
DF		Sets default values.
*DP		Readies the tablet to accept a digitized point.
IM	[E-mask],[S-mask],[P-mask]	Specifies input mask values to set recognized error conditions, service request conditions, and parallel poll response conditions.
IN		Initializes tablet to its power-on state.
IP	[$P1_x$],[$P1_y$],[$P2_x$],[$P2_y$]	Specifies P1 and P2 scaling points.
*OA		Same as OC instruction.
OC		Outputs last known X,Y, pen, softkey, status, and error information.
OD		Outputs last known X,Y, and pen information.
OE		Outputs current error condition.
OF		Outputs digitizing units per millimetre.
OI		Outputs tablet's identity.
*OK		Outputs value for selected softkey; range is from 0 to 32 768.
OP		Outputs scaling points P1 and P2.
OR		Outputs tablet's apparent resolution.
OS		Outputs sum of set bits of status word.
*RC		Same as OC instruction.
RS	[menu enable]	Outputs number of selected softkey; range is from 0 to 16.
SF		Sets switch-follow mode.
SG		Sets single-sample mode.
*SK		Clears any selected softkey values.
SN		Sets switch-normal mode.
TD		Initiates user-interaction self-test.
TP		Simulates a press on the stylus digitize switch.

*These instructions are included in the graphics tablet instruction set primarily for compatibility with other Hewlett-Packard graphics devices.

The Beep Instruction, BP

DEFINITION: The beep instruction, BP, generates the graphics tablet's internal tone. You can select from a set of musical tones and specify their duration and amplitude.

USES: The instruction is used to provide audio feedback to the operator. For example, you might want to acknowledge that a softkey has been selected.

SYNTAX: *BP* [frequency],[duration],[amplitude] terminator
or
BP terminator

DEFAULT: Frequency is 12 Hz, duration is 150 ms, amplitude is 4 on a scale of 0 to 5.

EXPLANATION: There are a few concepts you may wish to keep in mind when programming the beeper.

- A short sequence of tones provides more effective feedback to an operator than an individual beep.
- Selecting a standard sequence of tones to represent each function makes recognition easier for the operator.
- Conventionally, an ascending sequence denotes inquiry, and a descending sequence denotes acknowledgement.

Following are some suggestions for sequences that you may wish to program for different functions:

- a pick tone to indicate that a softkey was selected or a single point digitized:

  BP24,100,5

- an error tone to indicate that the user has digitized an undefined area or didn't answer a question correctly:

  BP5,75,5;BP;BP

- a question tone to indicate that the user needs to answer a question or enter data:

  BP36,30,5;BP40,60;BP43,100

- a finished tone to indicate that an operation has been completed:

  BP31,30,5;BP28,60;BP24,100

You can find references to a "hello" tone in the Operator's Manual and other parts of this manual. The "hello" tone is the short ascending sequence of notes that sound when the tablet is powered-on or initialized.

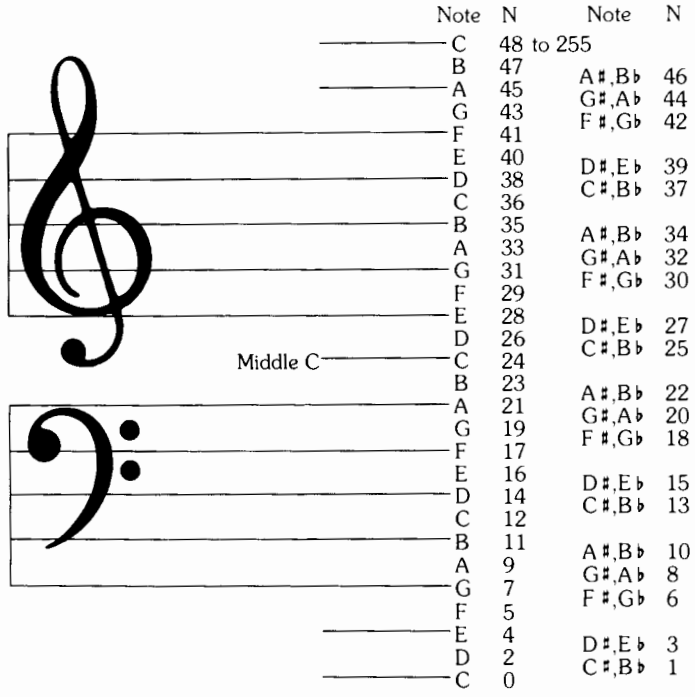
NOTE: Keep in mind, as you design your programs, that the tablet is inactive during a beep. No data can be obtained during a beep, and a pen press may be lost during a long beep. Also, do not attempt to communicate with other devices on the same HP-IB as the tablet during a beep. It is suggested that you program a WAIT statement long enough for the beep to finish after the BP instruction is output.□

The following discussion explains each of the three parameters. If the BP instruction is used without parameters, the tablet sounds using the last specified values for frequency, duration, and amplitude. Parameters outside the ranges specified cause error number 3.

Frequency

The frequency parameter determines the musical tones that are produced. The following diagram shows the parameter values (N) and corresponding notes.

Parameter Values and Corresponding Notes



Values 0 through 48 produce musical notes that are centered about middle C. Zero represents C two octaves below middle C (frequency of 130.81 Hz), and 48 represents C two octaves above middle C (frequency of 2093.0 Hz). The following formula can be used to determine the approximate frequency generated by the values 0 through 48:

frequency in Hz = $1.0595^{N} \times 130.81$ Hz

where the power N = the parameter value (0 through 48) and 130.81 Hz = the frequency of C represented by the value 0.

Parameter values of 49 through 255 are accepted but produce the same frequency as 48.

Duration

The duration parameter specifies how long the tone sounds in milliseconds. The valid values are 1 through 32767. The maximum value of 32767 milliseconds specifies almost 33 seconds.

Amplitude

The amplitude parameter specifies the volume of the tone. The valid values are 0 through 5. A value of 0 specifies no tone and a value of 5 specifies the loudest tone. You may wish to experiment with each value to determine which volume meets your needs.

The Continuous-Sampling Mode Instruction, CN

DEFINITION: The continuous-sampling mode instruction, CN, sets the graphics tablet's continuous-sampling mode.

USES: The instruction is used to follow curving lines, such as contour lines on maps.

SYNTAX: *CN* terminator

DEFAULT: Switch-normal mode (SN).

EXPLANATION: You can use the continuous-sampling mode in one of two ways:

1. You can set the tablet to begin taking points at regular time intervals when the stylus is pressed, and then stop when the stylus is pressed again. The tablet records points as long as the stylus is within approximately ¼ inch of the tablet's surface. This mode of continuous digitizing is called switch normal and is the default mode of digitizing when the CN instruction is specified. If you need to activate the switch-normal mode, use the switch normal instruction, SN.

2. You can specify that the tablet is to take points only when the stylus is pressed down. This mode of continuous digitizing is called switch follow and is activated by using the switch follow instruction, SF.

You need to be aware that you can possibly lose points in the continuous mode if your computer is slow. You can regulate the time intervals between points by using the cursor rate instruction, CR. Refer to that instruction for a discussion of the relationship between the data rate and your computer's speed.

When the tablet receives this instruction, the **DIGITIZE** light on the front panel is lit, and the tablet is ready to take points. When you press the pen tip onto the active area of the platen, the digitize switch within the stylus is activated, the tablet starts storing X,Y coordinates, and bit 2 of the status word is set.

The following sequence is recommended when using the CN instruction.

- Set the continuous-sampling mode (CN).
- Set the digitize-switch mode (SF or SN).
- Check bit 2 of the tablet's status word. (When a point is digitized, bit 2 is set.)
- If bit 2 is set, send the output digitized point instruction, OD, and then read the X,Y coordinates and pen status.
- If bit 2 is not set, keep checking bit 2.

Since the graphics tablet remembers a digitizing-switch mode specified prior to receiving a CN instruction, you can specify switch normal (SN) or switch follow (SF) before or after a CN instruction. Note that you cannot use the digitize point instruction, DP, when the continuous-sampling mode is in effect.

9874 Digitizer Compatibility

The graphics tablet accepts two parameters with the CN instruction so you can use programs written for the 9874 digitizer. The parameters are accepted, but not acted upon. Following is the syntax for the instruction with parameters:

CN [Δt,ΔD] terminator

The elapsed time between samples is represented by Δt, and the distance the cursor is moved between samples is represented by ΔD. Remember that you can regulate the time intervals on the graphics tablet with the cursor rate instruction, CR. The tablet does not have an instruction to regulate distance.

Note that the delta symbol (Δ) is used here, as it is commonly used in mathematics, to denote a change in a variable.

The Cursor Rate Instruction, CR

DEFINITION: The cursor rate instruction, CR, allows you to specify the rate at which the tablet checks the position of the stylus in points per second.

USES: The instruction can be used to control the data rate of the continuous-sampling mode and the rate at which data points are stored in a file or used to update a CRT cursor. However, it is important to understand the relationship between your computer's speed and the effect it can have on the data rate. Refer to the EXPLANATION below for a discussion of this issue.

SYNTAX: *CR* [points per second] terminator

DEFAULT: 60 points per second.

EXPLANATION: From 1 to 60 points per second can be specified as the parameter of the instruction. A value outside this range causes error number 3. A CR instruction with no parameter sets the rate to the default value 60. You do not need to use the instruction at all if you want 60 points per second.

It is important to realize that the speed of your computer, both in input/output and in computation, determines the rate at which data points are input. Some personal computers or desktop computers may limit your data rate to 8 points a second, 15 points a second, or 30 points a second.

If your computer is relatively slow, you could possibly lose points in the continuous-sampling mode. You might want to use the CR instruction to adjust the cursor rate to more closely match the computer's speed.

If you have a high-speed computer, you can make use of the maximum tablet data rate of 60 points per second. The default rate of 60 points per second matches the refresh rate of most CRTs and allows you to achieve fast, smooth cursor tracking.

For specific information about your computer's speed and the refresh rate of your CRT, contact your HP Sales and Support Office.

The Digitizer Clear Instruction, DC

DEFINITION: The digitizer clear instruction, DC, clears the tablet of all digitizing modes.

USES: This instruction is used any time you wish to terminate a digitizing mode either before or after a point is digitized. For example, you might use it to clear a digitizing mode when your program is interrupted from the keyboard rather than by selecting a menu item.

SYNTAX: *DC* terminator

EXPLANATION: The DC instruction does the following:

- clears the digitizing modes set by the continous- and single-sampling mode instructions, CN and SG, and the digitize point instruction, DP;
- clears any digitized point coordinates;
- clears bit 2 of the status word; and
- turns off the **DIGITIZE** light.

The Default Instruction, DF

DEFINITION: The default instruction, DF, sets certain tablet conditions to a known state.

USES: The instruction is used to return the tablet to a known state at the beginning of a program while maintaining the current settings of P1 and P2. This prevents unwanted conditions remaining in effect from previous operations.

SYNTAX: *DF* terminator

EXPLANATION: The default instruction, DF, performs the same functions as the initialize instruction, IN, except that it does not activate the self-test or change the settings of P1 and P2. Following are the conditions of the tablet after execution of a DF instruction.

Conditions Set by Default Instruction

Function	Condition
Data rate	60 points per second
Input masks	
E-mask	7
S-mask	0
P-mask	0
Status word	16 (bit 4 is set)
Softkey area	On
Softkey selection	0
Sampling mode (CN or SG)	None
Switch mode (SN or SF)	SN (switch normal)
Beeper	
Frequency	12 Hz
Duration	150 ms
Amplitude	4
P1 and P2	Not changed from previous setting

The Digitize Point Instruction, DP

DEFINITION: The digitize point instruction, DP, prepares the graphics tablet to recognize the next pen press as a digitized point.

HP Plotter Digitizing Compatibility

USES: If you are concerned about compatibility with digitizing operations on some HP plotters, you would use this instruction to digitize single points. Otherwise, you might prefer to use the single-sample mode instruction, SG, because the mode remains in effect until changed or cleared, allowing more efficient programming.

SYNTAX: *DP* terminator

EXPLANATION: When the tablet receives this instruction, the **DIGITIZE** light on the front panel is lit, and the tablet is ready to take a point. When you press the pen tip onto the active area of the platen, the digitize switch within the stylus is activated, a coordinate point is stored in the tablet, and bit 2 of the status word is set. With the DP instruction, the digitizing mode is cleared as soon as a point is digitized.

The following sequence is recommended when using the DP instruction.

- Use either the initialize instruction, IN, or the default instruction, DF, to set the tablet to known conditions.
- Use the DP instruction to ready the tablet to accept a digitized point.
- Send the output status instruction, OS, to the tablet and read the tablet's status word.
- Check bit 2 of the tablet's status word. (When a point is digitized, bit 2 is set.)
- If bit 2 is set, send the output digitized point instruction, OD, and then read the X,Y coordinates and pen status.
- If bit 2 is not set, keep checking bit 2.

Note that if the continous- or single-sampling modes are in effect (CN or SG instructions), the DP instruction is ignored. Remember that the digitizer clear instruction, DC, can be used to clear a digitizing mode.

The Input Mask Instruction, IM

DEFINITION: The input mask instruction, IM, controls recognized error conditions, service request conditions, and parallel poll response conditions. Service requests and parallel polls are defined in Chapter 4 and Appendix A.

USES: This instruction is used to make a change in any of the three categories mentioned above. For example, error number 7 (inconsistent stylus location data) is not recognized unless you use this instruction to include it. Also, the tablet does not send a service request or respond to a parallel poll unless you use this instruction to specify the conditions. For example, if you are using interrupt-driven processing, you may wish to specify that a service request be sent when a softkey is selected and when a digitized point is available. Refer to Chapter 4, HP-IB Interfacing, for more information about programming with interrupts.

SYNTAX: *IM* [E-mask value],[S-mask value],[P-mask value]
terminator
or
IM terminator

DEFAULT: E-mask value is 7, S-mask value is 0, P-mask value is 0.

EXPLANATION: The input mask instruction, IM, controls:

- the error conditions that are recognized (E-mask),
- the conditions that cause an HP-IB service request message (S-mask), and
- the conditions that cause a positive response to an HP-IB parallel poll (P-mask).

Each of the three masks is explained here and tables are given with the bit values, bit numbers, and meanings. The mask values specified in the IM instruction are the sum of any combination of the bit values for that mask. An IM instruction with no parameters sets the mask values to the default values 7,0,0. Reference is made to the status word in these explanations; "The Output Status Instruction, OS" contains more details about the status word.

Error Mask (E-Mask)

The error mask (E-mask) specifies the error conditions that the tablet recognizes. The E-mask value specified in the IM instruction is the sum of any combination of the bit values shown in the table. Refer to "The Output Error Instruction, OE" for a more detailed description of the causes of the errors.

Error Mask Bits

E-Mask Bit Value	Bit/Error Number	Meaning
0	0	No error.
1	1	Instruction not recognized.
2	2	Wrong number of parameters.
4	3	Invalid parameter value.
64	7	Inconsistent stylus location data.

When one of the errors listed in the table occurs, the bit in the E-mask that corresponds to the error number is tested to determine whether to set the error bit (bit 5) of the status word, turn on the front-panel **ERROR** light, and sound the error tone. If an error number is not included in the E-mask value and thus the error bit is not set, there is no way to determine whether that error occurred. For example, if you specify an E-mask value of 3, an invalid parameter value (error number 3) will never be recognized.

The default E-mask value of 7 specifies that error numbers 1, 2, and 3 are to be recognized (1 + 2 + 4) and that error number 7 is not to be recognized. To include error number 7, specify an E-mask value of 71 (1 + 2 + 4 + 64).

The valid range of E-mask values is 0 through 32767. A value outside this range causes error number 3. If you specify an E-mask that is not a sum of the bit values but is within the valid range, the default value of 7 is used.

Status Mask (S-Mask)

The status mask (S-mask) specifies when HP-IB service requests are to be sent. The S-mask value specified in the IM instruction is the sum of any combination of the bit values shown in the table. Each status mask bit number, except bit number 6, corresponds directly to the same bit number in the status word.

Status Mask Bits

S-Mask Bit Value	Status Bit Number	Meaning
1	0	Always clear.
2	1	Always clear.
4	2	A digitized point is available.
8	3	Initialized. (The power-on self-test is completed.)
16	4	Ready. (The power-on self-test, user-interaction self-test, and beep instruction BP, are completed.)
32	5	Error.
64	6	Not used.
128	7	The pen is pressed in one of the sixteen softkeys (menu boxes).
256	8	The pen is in proximity (within approximately ¼ inch) to the active platen area.
512	9	New cursor information is available.
1024	10	The pen is pressed against the active platen area.

If a bit is included in the S-mask value and the corresponding bit of the status word changes value, a service request (SRQ) is sent and bit 6 of the status word is set. (A logical AND is performed with the bits in the S-mask and the bits in the status word, and then a logical OR of the results of the ANDs.) For example, an S-mask value of 4 specifies that when a digitized point is available, setting bit 2, the service request message is to be sent. In this example, setting the other bits does not cause a service request message to be sent because no other bits are specified by the S-mask value of 4.

NOTE: To avoid potential problems, do not set up two conditions that may occur simultaneously to generate an SRQ, such as bit 2 (digitized point available) and bit 10 (pen pressed). Instead, use mutually exclusive conditions to generate an SRQ, such as bit 2 (digitized point available) and bit 7 (softkey selected).□

The default S-mask value of 0 specifies that no service request message is to be sent. The valid range of S-mask values is 0 through 32767. A value outside this range causes error number 3. If you specify an S-mask that is not a sum of the bit values but is within the valid range, the default value of 0 is used.

Parallel Poll Mask (P-Mask)

The parallel poll mask (P-mask) specifies which conditions cause a positive response (a logical 1) to a parallel poll on the HP-IB interface. The P-mask value specified in the IM instruction is the sum of any combination of the bit values shown in the table. Each parallel poll mask bit number corresponds directly to the same status bit number in the status word.

Parallel Poll Mask Bits

P-Mask Bit Value	Status Bit Number	Meaning
1	0	Always clear.
2	1	Always clear.
4	2	A digitized point is available.
8	3	Initialized. (The power-on self-test is completed.)
16	4	Ready. (The power-on self-test, user-interaction self-test, and beep instruction BP, are completed.)
32	5	Error.
64	6	Service request (SRQ) sent.
128	7	The pen is pressed in one of the sixteen softkeys (menu boxes).
256	8	The pen is in proximity (within approximately ¼ inch) to the active platen area.
512	9	New cursor information is available.
1024	10	The pen is pressed against the active platen area.

If a bit is included in the P-mask value and the corresponding bit of the status word changes value, the tablet sets the parallel poll bit. (A logical AND is performed with the bits in the P-mask and the bits in the status word, and then a logical OR of the results of the ANDs.) For example, a P-mask value of 48 specifies that only bits 4 and 5 (16 + 32) of the status word can cause the tablet to respond to a parallel poll with a logical 1 on the appropriate data line.

The bit on which the graphics tablet responds to a parallel poll and the data line used are determined by the setting of the switch which selects the HP-IB address for the unit. Refer to the 9111 Operator's Manual for information about setting address switches. The following table shows the HP-IB address, the parallel poll bit position, and the data line used in response to the parallel poll. Refer to Chapter 4, HP-IB Interfacing, in this manual for more details about parallel polls.

Parallel Poll Response Bits and Data Lines

HP-IB Address	Parallel Poll Bit Position	HP-IB Data Line Number	
0	7	8	
1	6	7	
2	5	6	
3	4	5	
4	3	4	
5	2	3	
6	1	2	← TABLET PRESET ADDRESS
7	0	1	

The default P-mask value of 0 specifies that the tablet will not respond to a parallel poll. The valid range of P-mask values is 0 through 32767. A value outside this range causes error number 3. If you specify a P-mask that is not a sum of the bit values but is within the valid range, the default value of 0 is used.

The Initialize Instruction, IN

DEFINITION: The initialize instruction, IN, performs the power-on self- test and then sets the graphics tablet to its power-on conditions.

USES: The instruction is used to return the tablet to a known state at the beginning of a program. This prevents unwanted conditions remaining in effect from previous operations.

SYNTAX: *IN* terminator

EXPLANATION: The initialize instruction, IN, is equivalent to switching the tablet off and then on again. The following conditions exist after the graphics tablet is initialized:

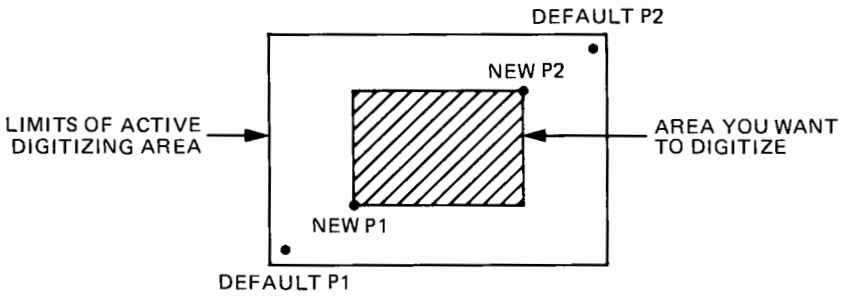
Conditions Set by Initialize Instruction

Function	Condition
Data rate	60 points per second
Input masks	
E-mask	7
S-mask	0
P-mask	0
Status word	24 (bits 3 and 4 are set)
Softkey area	On
Softkey selection	0
Sampling mode (CN or SG)	None
Switch mode (SN or SF)	SN (switch normal)
Beeper	
Frequency	12 Hz
Duration	150 ms
Amplitude	4
P1	400, 400 DU's
P2	11632, 8340 DU's

The Input Points Instruction, IP

DEFINITION: The input points instruction, IP, allows you to set P1 and P2 through program control.

USES: You can use the instruction to move the lower-left and upper-right corners of your digitizing area from the default or current locations of P1 and P2. This is especially useful if the area you want to digitize is considerably smaller than the tablet's active digitizing area as shown in the following diagram.



If you then need to map the digitizing area onto a CRT, you can make more effective use of the CRT space. Refer to Chapter 2, Common Graphics Tablet Operations, for a discussion of "Digitizing P1 and P2," and "Scaling."

SYNTAX: *IP* [$P1_x$],[$P1_y$],[$P2_x$],[$P2_y$] terminator
or
IP terminator

DEFAULT: $P1_x$ is 400, $P1_y$ is 400, $P2_x$ is 11 632, $P2_y$ is 8340.

EXPLANATION: The parameters of the instruction follow.

$P1_x$ is the X-coordinate of P1.

$P1_y$ is the Y-coordinate of P1.

$P2_x$ is the X-coordinate of P2.

$P2_y$ is the Y-coordinate of P2.

The coordinates are specified in the order shown above, and must be in absolute digitizing units. The range of accepted values for each of the four coordinates is from −999 999 to 999 999. Values outside this range cause error number 3 and the current values of P1 and P2 do not change. An IP instruction with no parameters sets P1 and P2 to the default values 400, 400, 11 632, 8340.

Even though the tablet will accept values from −999 999 to 999 999, the actual range of values you are likely to use correspond to the limits of the tablet's active digitizing area.

The digitizing limits with the softkeys enabled are:

$X_{min} = -124$ $X_{max} = 12\,156$
$Y_{min} = -124$ $Y_{max} = 8832$

The digitizing limits with the softkeys disabled are:

$X_{min} = -124$ $X_{max} = 12156$
$Y_{min} = -124$ $Y_{max} = 9668$

Refer to Chapter 1 or Appendix C for diagrams of the tablet's surface with measurements in digitizing units. Refer to "The Read Softkey Instruction, RS" for information about how to enable or disable softkeys.

Note that if you use this instruction with parameters that are variables, it is very important to also send the commas to the tablet. Otherwise, the tablet treats the instruction as IP without parameters and sets P1 and P2 to the default values. Each computer has a format for sending HP-GL instructions to the tablet. Your computer's documentation is the source of information for how to send the instructions.

The following examples show how to send an IP instruction using a variable format on various HP computers. The computers in both examples automatically generate a line feed which terminates the IP instruction. Semicolons are used instead of commas to separate the mnemonic and variables because the the semicolon eliminates excess spaces and the line length might be too long otherwise.

Example for HP-85, HP-87, and HP 9845 Computers

```
PRINT "DC;IP";X1;",";Y1;",";X2;",";Y2
```

Note that for the computers in this example, the commas must be enclosed in quotation marks to be sent to the tablet.

Example for HP 9826 and HP 9836 Computers

```
PRINT "DC;IP";X1,Y1,X2,Y2
```

Note that for the computers in this example, the commas are not enclosed in quotation marks to be sent to the tablet.

The Output Actual Stylus Position Instruction, OA

DEFINITION: The output actual stylus position instruction, OA, works exactly like the output cursor instruction, OC. Refer to "The Output Cursor Instruction, OC" in this manual for an explanation.

HP Graphics Device Compatibility

USES: This instruction is included in the 9111 HP-GL instruction set to allow compatibility with other HP graphics devices that use the instruction, such as the 9874 digitizer, some HP plotters, and graphics CRTs.

The Output Cursor Instruction, OC

DEFINITION: The output cursor instruction, OC, prepares the graphics tablet to output the following information: the X,Y coordinate of the last known stylus location, the pen status, the number of the softkey if one has been selected, the status word value, and the error number, if any.

The output actual stylus position instruction, OA, and the read cursor instruction, RC, operate as described here for the OC instruction.

USES: This instruction is used to tell where the stylus is. It is especially useful in cursor tracking.

SYNTAX: *OC* terminator

OUTPUT: The tablet's data are output to the computer as integers in ASCII in the form:

X,Y,P,M,S,E CR LF

EXPLANATION: Each output parameter is explained below.

X is the X-coordinate of the last known stylus position in digitizing units. The range of values is from −900 to 12900.

Y is the Y-coordinate of the last known stylus position in digitizing units. The range of values is from −900 to 10300.

P is the last known pen status and is one of the following:

0 if the pen is not pressed down, or

1 if the pen is pressed down.

M is the decimal value of the softkey selected. The range of values is from 0 to 16. Zero indicates that no softkey was selected. 1 to 16 correspond to the numbers printed in the softkey boxes on the tablet's surface.

S is the sum of the weighted values of the set bits in the status word. Refer to "The Output Status Instruction, OS" for a table of the weighted values and their meanings.

E is the number corresponding to the last error, if any. Refer to "The Output Error Instruction, OE" for a table of error numbers and their causes.

The OC instruction must be followed by a computer-dependent input statement, such as READ, INPUT, or ENTER, to obtain the tablet's output data. You don't have to read all the output parameters. However, you must read all the parameters preceding the last parameter you want. For example, you can read X,Y or X,Y,P or X,Y,P,M but not X,Y,M.

Note that negative coordinates are output with a minus sign and are caused by a stylus location to the extreme left or bottom of the platen. Even though larger ranges are listed above for the coordinate values, the actual range of values you are likely to receive are restricted to the limits of the tablet's active digitizing area. The digitizing limits with the softkeys enabled are:

$X_{min} = -124$ $X_{max} = 12156$
$Y_{min} = -124$ $Y_{max} = 8832$

The digitizing limits with the softkeys disabled are:

$X_{min} = -124$ $X_{max} = 12156$
$Y_{min} = -124$ $Y_{max} = 9668$

Refer to Chapter 1 or Appendix C for more complete diagrams of the tablet's surface with measurements in digitizing units. Refer to "The Read Softkey Instruction, RS" for information about how to enable or disable softkeys.

This instruction does not require that a digitize mode be set or that the stylus be pressed. It is not necessary that the instruction interact with bit 2 of the status word.

NOTE: The timing may be different between the parameters returned as output from the OC instruction. For example, assume you have digitized a softkey. The first time you receive output, the pen status may have changed to 1 (pen pressed). However, the softkey value may not be reflected until the next time you receive output.□

The Output Digitized Point Instruction, OD

DEFINITION: The output digitized point instruction, OD, prepares the graphics tablet to output the X- and Y-coordinates and pen status associated with the last digitized point.

USES: This instruction is used to send the coordinates and pen status to the computer after each point is digitized.

SYNTAX: *OD* terminator

OUTPUT: The stylus position and status are output to the computer as integers in ASCII in the form:

X,Y,P CR LF

EXPLANATION: Each output parameter is explained below.

X is the X-coordinate of the digitized point in digitizing units. The range of values is from −120 to 12152.

Y is the Y-coordinate of the digitized point in digitizing units. The range of values is from −120 to 9115.

P is the pen status and is one of the following:

−1 if the OD instruction is received by the graphics tablet and a digitizing mode (CN, SG, or DP) is not set,

0 for the last digitized point in the continuous-sampling mode (CN), or

1 for all points digitized in the single digitizing modes (SG or DP) and all except the last point in the continuous mode (CN).

The OD instruction must be followed by a computer-dependent input statement, such as READ, INPUT, or ENTER, to obtain the tablet's output data. You don't have to read all the output parameters. However, you must read all the parameters preceding the last parameter you want. For example, you can read X or X,Y or X,Y,P but not X,P.

Note that negative coordinates are output with a minus sign and are caused by a stylus location to the extreme left or bottom of the platen.

The following sequence is recommended when using the OD instruction.

- Set the continuous- or single-sampling mode (CN or SG).
- Check bit 2 of the tablet's status word. (When a point is digitized, bit 2 is set.)

- If bit 2 is set, send the OD instruction, and then read the X,Y coordinates and pen status.
- If bit 2 is not set, keep checking bit 2.

When the graphics tablet receives the CN, SG, or DP instruction, the **DIGITIZE** light on the front panel is lit, and the tablet is ready to take a point. When you press the pen tip onto the active area of the platen, the digitize switch within the stylus is activated, a coordinate point is stored in the tablet, and bit 2 of the status word is set. Once bit 2 is set, and the OD instruction is sent to the tablet, the computer can read the X,Y coordinates and pen status.

NOTE: If the OD instruction is received by the graphics tablet and bit 2 of the status word is not set, the tablet takes control of the HP-IB control lines and stops further data communication until bit 2 is set. System I/O communication is halted until a point is digitized. It is recommended that you not send an OD instruction until you know that bit 2 is set.□

If the OD instruction is received by the graphics tablet and a digitizing mode (CN, SG, or DP) is not in effect, error number 1 is generated, and the following is output:

0,0,−1 CR LF

The Output Error Instruction, OE

DEFINITION: The output error instruction, OE, prepares the graphics tablet to output the number corresponding to the last error, if any.

USES: This instruction is used to determine the cause of errors.

SYNTAX: *OE* terminator

OUTPUT: error number CR LF

EXPLANATION: There are three types of errors that can occur on the 9111:

- normal operation errors,
- user-interaction self-test errors, and
- hardware errors.

These three errors are explained in the following paragraphs, along with tables that list the error numbers and corresponding causes.

When the tablet recognizes an error, bit position 5 of the status word is set, the **ERROR** light on the front panel is lit, and the error tone sounds. Some of the normal operation errors can be masked so they are not recognized (refer to the discussion of "Normal Operation Errors" that follows). All of the user-interaction self-test and hardware errors are recognized when they occur.

All of the recognized error numbers can be read by the OE instruction if the tablet is functionally capable. When the graphics tablet receives the OE instruction, bit position 5 is cleared and the **ERROR** light is turned off. The hardware errors also cause a combination of front-panel indicator lights to be displayed (refer to the discussion of "Hardware Errors" that follows).

In addition to being included in a program, the OE command can be executed immediately so you can determine the cause of the error right away. The OE command must be followed by a computer-dependent input statement, such as READ, INPUT, or ENTER, to obtain the error number.

Normal Operation Errors

The following errors may occur during normal operation of the tablet. Error numbers 1 through 3 are user programming errors. Error 7 is indicative of hardware failure or an exceptionally noisy environment. Error 100 is caused by hardware and firmware errors.

Normal Operation Errors

Error Number	Cause
0	No error.
1	The HP-GL instruction is not recognized because: • the two-letter mnemonic is invalid; or • the entire instruction exceeds 45 characters, excluding blanks and carriage returns; or • the OD instruction has been sent to the tablet without either the single (SG) or continuous (CN) digitizing mode being in effect. (Refer to the SG and CN instructions for information about setting up these modes.)
2	The HP-GL instruction contains an invalid number of parameters.
3	The HP-GL instruction contains an invalid parameter.
7	The stylus location data is inconsistent. This generally indicates an exceptionally noisy environment or a hardware failure. In order to generate this error, four points in succession must fail to be consistent. No points are output the first three times the data fails this consistency test.
100	A system error, such as stack overflow or an interrupt problem, has occurred during normal use of the machine.

Error numbers 1 through 7 can be masked so they are not recognized. Error numbers 1 through 3 are normally recognized when they occur. Error number 7 is not recognized unless you use the input mask instruction, IM, to include it. Refer to the IM instruction for details about masking errors. Error 100 cannot be masked.

User-Interaction Self-Test Errors

These errors can occur during the self-test activated by the **SELF TEST** switch on the rear of the tablet, or by the test digitizer instruction, TD.

Error Number	Cause
50	An illegal proximity signal is detected when the self-test is requested. This error occurs if the stylus tip is in close proximity to, or touching, the platen when you initiate the self-test. It could also indicate a faulty proximity testing circuit.
51	An illegal pen press is detected when the self-test is requested. This error occurs if the stylus tip is *not* in close proximity to the platen but the pen is pressed when you initiate the self-test. The pen switch could be stuck or a lead in the cable shorted. Try replacing the cartridge, because a bent cartridge may bind in the stylus.
52	A pen press is detected before proximity to the platen is detected. This error occurs after the "hello" tone sounds and can be caused by a pen press off the platen. The platen could be dead, or the center cable conductor open.
53	A position error is indicated. You might have missed the self-test dot, or there could be a problem in the platen or stylus mechanism.

Hardware Errors

These errors can be generated only during the power-on or user-interaction self-tests. Remember that the power-on self-test is also performed when the initialize instruction, IN, is sent to the tablet. When one of these conditions is detected, the tablet halts the test and displays a combination of front-panel indicator lights to indicate the error. The status of the lights for each error condition is included in the following table. The error numbers can also be read with an OE instruction if the tablet is functionally capable.

If a hardware error occurs, write down the error number and refer to the 9111 Operator's Manual, "Shipment," for instructions on how to obtain servicing assistance.

Error Number	Cause	Indicator Light Status		
		DIGITIZE	MENU	ERROR
101	Processor Register or Flag Error	Off	Off	On
102	ROM Checksum Error	Off	On	Off
103	RAM Test Error	Off	On	On
104	I/O Port Error	On	Off	Off
105	Interface Chip Error	On	Off	On
106	Phase Counter Error	On	On	Off
107	Interrupt Mask	On	On	On
108	153 Hz Clock Interrupt	On	On	On
109	RAM Timer Interrupt	On	On	On
110	HP-IB Chip Interrupt	On	On	On

The Output Factor Instruction, OF

DEFINITION: The output factor instruction, OF, prepares the graphics tablet to output the number of digitizing units per millimetre in both the X- and Y-axes.

USES: This instruction is used to obtain the number of digitizing units per millimetre. The factor is used by software packages to calculate the number of digitizing units in a given length.

SYNTAX: *OF* terminator

OUTPUT: The graphics tablet always outputs the following:

40,40 CR LF

EXPLANATION: A digitizing unit is 0.025 mm in length. Therefore, there are 40 digitizing units per millimetre (40 × 0.025 mm = 1 mm).

The OF instruction must be followed by a computer-dependent input statement, such as READ, INPUT, or ENTER, to obtain the tablet's output data.

The Output Identification Instruction, OI

DEFINITION: The output identification instruction, OI, prepares the graphics tablet to output a device identifier.

USES: This instruction is used by software packages to identify the input device that is on-line.

SYNTAX: *OI* terminator

OUTPUT: The graphics tablet always outputs the following character string:

9111T CR LF

EXPLANATION: The T after 9111 indicates that your tablet includes the instructions that are discussed in Chapter 5, The HP 1351S Graphics Display System. Your tablet has all the capabilities of a 9111A tablet plus the additional six instructions in Chapter 5. If you do not use the additional instructions, program your software to look only for the 9111 part of the identifier.

The OI instruction must be followed by a computer-dependent input statement, such as READ, INPUT, or ENTER, to obtain the tablet's output data.

The Output Key Instruction, OK

DEFINITION: The output key instruction, OK, prepares the graphics tablet to output a value corresponding to the selected softkey.

9874 Digitizer Compatibility

USES: This instruction is used for program compatibility with the 9874 digitizer. The read softkey instruction, RS, is the preferred instruction to use with the graphics tablet.

SYNTAX: *OK* terminator

OUTPUT: softkey value CR LF

EXPLANATION: Following is a table of the softkeys selected and the corresponding output values.

Softkey Values

Softkey Selected	Output Value	Softkey Selected	Output Value
1	1	9	256
2	2	10	512
3	4	11	1024
4	8	12	2048
5	16	13	4096
6	32	14	8192
7	64	15	16384
8	128	16	32768

The OK instruction must be followed by a computer-dependent input statement, such as READ, INPUT, or ENTER, to obtain the tablet's output data.

When a softkey is selected, bit 7 of the status word is set and the **MENU** light is lit. When the tablet receives the OK instruction, bit 7 of the status word is cleared. Use the set key instruction, SK, to clear the softkey value and turn off the **MENU** light.

The Output Points Instruction, OP

DEFINITION: The output points instruction, OP, prepares the graphics tablet to output the current coordinates of the scaling points, P1 and P2, in digitizing units.

USES: This instruction is used in scaling the graphics tablet to other devices such as a CRT or plotter.

SYNTAX: *OP* terminator

OUTPUT: After an OP instruction is received, the plotter outputs the coordinates of P1 and P2 in digitizing units as up to five integers in ASCII in the following form:

$P1_x$,$P1_y$,$P2_x$,$P2_y$ CR LF

EXPLANATION: Each output parameter is explained below.

$P1_x$ is the X-coordinate of P1.

$P1_y$ is the Y-coordinate of P1.

$P2_x$ is the X-coordinate of P2.

$P2_y$ is the Y-coordinate of P2.

The range of values for each of the four coordinates is from −999999 to 999999. However, the actual values you are likely to obtain correspond to

the limits of the tablet's digitizing area. The digitizing limits with the softkeys enabled are:

$X_{min} = -124$ $X_{max} = 12\,156$
$Y_{min} = -124$ $Y_{max} = 8832$

The digitizing limits with the softkeys disabled are:

$X_{min} = -124$ $X_{max} = 12\,156$
$Y_{min} = -124$ $Y_{max} = 9668$

Remember that the default values of P1 and P2 are 400 , 400 and 11 632 , 8340 and that you can change these values with the IP instruction. Refer to Chapter 1 or Appendix C for diagrams of the tablet's surface with measurements in digitizing units. Refer to "The Read Softkey Instruction, RS" for information about how to enable or disable softkeys.

The OP instruction must be followed by a computer-dependent input statement, such as READ, INPUT, or ENTER, to obtain the tablet's output data.

The Output Resolution Instruction, OR

DEFINITION: The output resolution instruction, OR, prepares the graphics tablet to output its resolution in millimetres for both the X- and Y-axes. The resolution is the smallest distance that the tablet can distinguish between two digitized points.

USES: This instruction is used by software packages to determine the tablet's resolution for scaling purposes.

SYNTAX: *OR* terminator

OUTPUT: The graphics tablet always outputs the following:

.025,.025 CR LF

EXPLANATION: The apparent resolution of the graphics tablet is 0.025 mm. However, the actual resolution is 0.100 mm. The tablet rounds all X,Y coordinate values to the nearest increment of 0.100 mm, but then outputs these values as digitizing units of 0.025 mm. A digitizing unit of 0.025 mm allows compatibility with other Hewlett-Packard digitizers and plotters.

The OR instruction must be followed by a computer-dependent input statement, such as READ, INPUT, or ENTER, to obtain the tablet's resolution.

The Output Status Instruction, OS

DEFINITION: The output status instruction, OS, prepares the graphics tablet to output the sum of the bit values that are set in the status word.

USES: The instruction is most commonly used to determine when the user has digitized a point or selected a softkey. This is accomplished by checking bit 2 or bit 7 of the status word.

SYNTAX: *OS* terminator

OUTPUT: Upon receipt of the OS instruction, the internal eleven-bit status word is converted to an integer that is output in ASCII in the form:

status CR LF

A table under EXPLANATION shows each bit, the value, why the bit is set, and how the bit is cleared.

EXPLANATION: The status word is the primary mechanism for communicating the inner workings of the tablet to the user. The status word contains eleven bits, with nine of the bits representing significant conditions within the tablet.

The status word provides the following information that is useful in programming digitizing applications and in debugging programs:

- whether there is a digitized point available;
- the error number, if any;
- whether a service request (SRQ) is sent;
- which softkey is selected, if any;
- whether the pen is in proximity to the tablet;
- whether the cursor position is updated; and
- whether the pen is pressed.

The following table shows the bits that are set by each of the conditions listed above.

Status Word

Bit Value	Bit Position	Meaning	How Bit Is Cleared
1	0	Always clear.	
2	1	Always clear.	
4	2	A digitized point is available.	OD, DC, DF, IN
8	3	Initialized. (The power-on self-test is completed.)	OS, DF
16	4	Ready. (The power-on self-test, user-interaction self-test, and beep instruction BP, are completed.)	Initiating the self-tests or beep instruction BP.
32	5	Error.	OE, DF, IN
64	6	Service request (SRQ) sent.	Sending status byte message in response to a serial poll, or if service is no longer required.
128	7	The pen is pressed in one of the sixteen soft-keys (menu boxes).	OK, RS, DF, IN

Status Word (Continued)

Bit Value	Bit Position	Meaning	How Bit Is Cleared
256	8	The pen is in proximity (within approximately ¼ inch) to the active platen area.	Remove the pen tip from the active platen area.
512	9	New cursor information is available.	Read data in default binary transfer mode; DF, IN, OC.
1024	10	The pen is pressed against the active platen area.	Lift the pen.

At power-on, the status is 24, the sum of 8 (initialized) and 16 (ready). Bit 3 is cleared after execution of the OS instruction, and the status becomes 16 (ready).

The OS instruction must be followed by a computer-dependent input statement, such as READ, INPUT, or ENTER, to obtain the tablet's output data.

NOTE: In response to a serial poll, the graphics tablet sends a status byte containing the lower eight bits (0 to 7) of the current status word. To ensure consistent status information, it is best to use either serial polls or OS instructions in a program, not both. Refer to Chapter 4, HP-IB Interfacing, for more information about serial polls.□

The Read Cursor Instruction, RC

DEFINITION: The read cursor instruction, RC, works exactly like the output cursor instruction, OC. Refer to "The Output Cursor Instruction, OC" in this manual for an explanation.

HP Graphics Device Compatibility

USES: This instruction is included in the 9111 HP-GL instruction set to allow compatibility with other HP graphics devices that use the instruction, such as the 9874 digitizer, some HP plotters, and graphics CRTs.

The Read Softkey Instruction, RS

DEFINITION: The read softkey instruction, RS, prepares the graphics tablet to output the number of the selected softkey.

USES: The instruction is used to determine which softkey is selected or to expand the active digitizing area of the platen by disabling the softkeys.

SYNTAX: *RS* [menu enable] terminator

DEFAULT: The softkeys are enabled (RS1) unless this instruction is used to disable them (RS0).

OUTPUT: softkey value CR LF

EXPLANATION: The menu enable parameter can be one of the following:

1 enables the menu or softkey area of the platen, or
0 disables the menu or softkey area of the platen.

A value other than 1 or 0 causes error number 3, and the instruction is executed with the softkeys enabled. The RS instruction with no parameters is executed with the softkeys enabled.

By using the softkeys, you don't have to use a routine like that in "Interpreting Areas on the Tablet" in Chapter 2, which looks at the exact X,Y coordinates digitized to see if they fall within certain boundaries. Following is a table of the softkeys selected and the corresponding output values. The output value is the same with either parameter.

Softkey Values

Softkey Selected	Output Value	Softkey Selected	Output Value
None	0	9	9
1	1	10	10
2	2	11	11
3	3	12	12
4	4	13	13
5	5	14	14
6	6	15	15
7	7	16	16
8	8		

The RS instruction must be followed by a computer-dependent input instruction, such as READ, INPUT, or ENTER, to obtain the tablet's output data.

The following sequence is recommended when using the RS instruction.

- Check bit 7 of the tablet's status word. (When a softkey is selected, bit 7 is set.)
- If bit 7 is set, send the read softkey instruction (RS1), and then read the softkey value.
- If bit 7 is not set, keep checking bit 7.

When a softkey is selected, bit 7 of the status word is set and the **MENU** light is lit. When the tablet receives the RS instruction, the softkey value is cleared, bit 7 of the status word is cleared, and the **MENU** light is turned off.

The Switch Follow Instruction, SF

DEFINITION: The switch follow instruction, SF, specifies that the tablet is to take points at regular time intervals when the stylus remains pressed. The instruction is used in the continuous-sampling mode, which is activated by the CN instruction.

USES: This instruction is used for drawing curved lines or tracing pictures placed on the tablet's surface. It also can be used when you want to put an ink cartridge in the stylus, draw a freehand sketch on paper placed on the tablet, and then digitize the drawing.

SYNTAX: *SF* terminator

EXPLANATION: Each point must be read individually. First send the output digitized point instruction, OD, to the tablet, and then read the X,Y coordinates and pen status. The last point sent in this mode always has a pen status of zero. The pen status for all other points in this mode is always one. Refer to "The Continuous-Sampling Mode Instruction, CN" for more details.

The Single-Sample Mode Instruction, SG

DEFINITION: The single-sample mode instruction, SG, sets the graphics tablet's single-sample mode.

USES: This instruction is used to digitize single points. For example, you can digitize strip charts, pick items from a menu, or draw straight lines in this mode.

SYNTAX: *SG* terminator

EXPLANATION: When the tablet receives this instruction, the **DIGITIZE** light on the front panel is lit, and the tablet is ready to take a point. When you press the pen tip onto the active area of the platen, the digitize switch within the stylus is activated, a coordinate point is stored in the tablet, and bit 2 of the status word is set.

Use the output digitized point instruction, OD, to send the X,Y coordinates and pen status to the computer.

The Set Key Instruction, SK

DEFINITION: The set key instruction, SK, clears any previously digitized softkey value, clears bit 7 of the status word, and turns off the **MENU** light on the front panel.

USES: This instruction is used to clear the graphics tablet between softkey selections but it does not store the softkey value. It is used in situations where you don't need to store the softkey value. The RS instruction with a parameter of 1 (RS1) stores the softkey value as well as performing the functions of the SK instruction.

SYNTAX: *SK* terminator

9874 Digitizer Compatibility

EXPLANATION: The graphics tablet accepts one parameter with the SK instruction so that you can use programs written for the 9874 digitizer. The parameter is accepted, but not acted upon. Following is the syntax for the instruction with a parameter:

SK [any value] terminator

The Switch Normal Instruction, SN

EXPLANATION: The switch normal instruction, SN, sets the tablet to begin taking points at regular time intervals when the stylus is pressed, and then stop when the stylus is pressed again. This instruction is used in the continous-sampling mode, which is activated by the CN instruction.

USES: This instruction is especially useful in tracing fine detail because you don't have to keep the stylus pressed down.

SYNTAX: *SN* terminator

EXPLANATION: Each point must be read individually. First send the output digitized point instruction, OD, to the tablet, and then read in the X,Y coordinates and pen status. The last point sent in this mode always has a pen status of zero. The pen status for all other points in this mode is always one. Refer to "The Continuous-Sampling Mode Instruction, CN" for more details.

The Test Digitizer Instruction, TD

DEFINITION: The test digitizer instruction, TD, activates the graphics tablet's user-interaction self-test.

USES: This instruction is useful when it is more convenient to send an instruction from the computer to activate the self-test than it is to use the **SELF TEST** switch on the rear of the tablet.

SYNTAX: *TD* terminator

EXPLANATION: After all the lights flash and you hear the "hello" tone, press the pen on the self-test dot marked on the lower-right corner of the platen. When the self-test dot is digitized, the "hello" tone sounds again and the test is complete. Otherwise, an error condition is generated. The 9111 Operator's Manual contains more information about the self-test. "The Output Error Instruction, OE" describes error numbers and their possible causes.

If any instruction is received by the graphics tablet while it is waiting for the digitized point, the self-test is aborted.

The Take Point Instruction, TP

DEFINITION: The take point instruction, TP, simulates the press of the digitize switch in the stylus. A point is digitized at the stylus location whether or not the stylus is actually pressed.

USES: This instruction can be used to force a point to be digitized in the single-sample mode or to terminate digitizing a string of data in the continuous-sampling, switch-normal mode.

SYNTAX: *TP* terminator

EXPLANATION: Remember that the single-sample mode is activated by the SG instruction, and the continuous-sampling, switch-normal mode is activated by the CN and SN instructions.

Chapter 4
HP-IB Interfacing

This chapter defines the 9111 graphics tablet's implementation of the Hewlett-Packard Interface Bus (HP-IB). The major areas discussed in this chapter are summarized below.

- "HP-IB Implementation on the 9111 Graphics Tablet" lists the HP-IB functions implemented on the tablet.
- "HP-IB Operations" discusses such topics as addressing the tablet; the talk-only mode; ASCII code and the default binary response; clearing the tablet; aborting bus transactions; sending service requests; handling service requests by performing interface status checks, interrupts, serial polls, and parallel polls; and asserting EOI.
- The Sending and Receiving sections include examples of sending and receiving data with a variety of computers.

This chapter assumes the user has a working knowledge of the HP-IB; however, if you wish to refresh your memory on HP-IB structure, refer to "Appendix A — An HP-IB Overview" in this manual. This chapter also assumes that you are familiar with the input and output operations of your computer. Refer to your computer's I/O documentation for more information.

HP-IB Implementation on the 9111 Graphics Tablet

The HP-IB conforms to IEEE 488-1978 specifications, and direct interconnection of the HP-IB is via a connector on the rear panel. The HP-IB functions implemented in the graphics tablet are as follows:

1. Source Handshake (SH1)
2. Acceptor Handshake (AH1)
3. Talker (T5 — Basic Talker, Serial Poll, Talk Only Mode, Unaddress If My Listen Address); and No Extended Talker (TE0)
4. Listener (L4 — Basic Listener, Unaddress If My Talk Address); and No Extended Listener (LE0)
5. Service Request (SR1)
6. No Remote/Local (RL0)
7. Parallel Poll (PP2 — Local Configuration)
8. Device Clear (DC1)
9. No Device Trigger (DT0)
10. No Controller (C0)

Addressing the Graphics Tablet

The graphics tablet is set to an HP-IB address code of 06 at the factory. This corresponds to a listen character of & and a talk character of F. The tablet functions as a talker or listener depending on the instructions it receives from the computer. In order to communicate effectively with the tablet, it is important that you completely understand the addressing protocol of your computer.

Computers with High Level I/O Statements

Many computers provide higher level input/output (I/O) statements that are used to specify device addresses on the HP-IB. In these cases, the addressing protocol (unlisten, talk, listen) is a function of the computer's internal operating system and need not be of concern to you.

Computers with No High Level I/O Statements

On computers with no high level I/O statements, addressing devices on the HP-IB is accomplished using mnemonics, such as CMD, which serve as the "bus command."

When bus commands are necessary, a typical addressing sequence is

<Unlisten Command> <Talk Address> <Listen Address>

This sequence is made up of three major parts which serve the following purposes:

1. The unlisten command is the universal bus command with a character code of ?. It unaddresses all listeners. After the unlisten command is transmitted, no active listeners remain on the bus.
2. The talk address designates the device that is to talk. A new talk address automatically unaddresses the previous talker.
3. The listen addresses designate one or more devices that are to listen. A listen address adds the designated device as listener along with other addressed listeners.

This basic addressing sequence simply states who is to talk to whom. The unlisten command (?) plays a vital role in this sequence. It is important that a device receive only the data that is intended for it.

When a new talk address is transmitted in the addressing sequence, the previous talker is unaddressed. Therefore, only the new talker can send data on the bus and there is no need to routinely use an untalk command in the same manner as the unlisten command.

Changing the Tablet's Address

The graphics tablet's address can be changed using the five rear-panel address switches. The use of the address switches is explained in detail in the 9111 Operator's Manual.

Using the Talk-Only Mode

The graphics tablet can be placed in talk-only mode by setting the rear-panel address switches to 31. In talk-only mode, the tablet does not use

the HP-IB addressing protocol. Instead, it responds to all initiated data transfers by sending the default binary response, which is the stylus location and status word in six bytes. The default binary response is discussed in more detail later in this chapter.

The talk-only mode can be useful for debugging purposes. By putting the tablet in talk-only mode and a printer in listen-only mode, you can print out the tablet's data for analysis.

Operating Note

If your program stops for no apparent reason and no further processing is possible, it could be because an automatic handshaking operation between the computer and tablet has not been completed for some reason. Often you can continue processing by doing a "soft" reset on your computer. A "soft" reset is accomplished on many Hewlett-Packard computers by pressing the **RESET** key on the computer once. Some computers have an input/output statement such as SET TIMEOUT that allows you to program for such a condition.

Data Messages

There are two types of data messages sent by the graphics tablet: ASCII code and the default binary response. If you only send a computer-dependent read statement, such as READ, INPUT, or ENTER, to the tablet, you will receive data in the default binary format (three 16-bit words). If you precede a read statement with an HP-GL output instruction, you will receive data in eight-bit ASCII code.

ASCII code is commonly used on Hewlett-Packard and other computers. Available software is likely to use ASCII code. It provides a standard correspondence between binary codes that are easily understood by the computer and alphanumeric symbols that are easily understood by humans. However, ASCII requires more computer time for processing and provides a larger volume of data. To see how much more data is generated in ASCII code than in the default binary response, compare the ASCII code and default binary response format illustrations shown later in this section.

The default binary data is more difficult to interpret, but its use enables more efficient data communications. Data volume and host computer processing are both minimized. Before using the default binary response, it is important to understand whether and how your computer can handle the data and what kind of software support is available for this type of binary data.

A description of ASCII code and the default binary response follows.

ASCII Code

When you send a specific output instruction to the tablet, such as the output status instruction, OS, or the read softkey instruction, RS, data is returned to the computer in eight-bit ASCII code. ASCII code is sent with a trailing line feed (LF) and carriage return (CR) with EOI true during the line feed. Appendix C contains a table of ASCII codes. Asserting EOI is explained later in this chapter.

Following is an example of output the tablet could send in response to an output cursor instruction, OC.

ASCII Output

X-Coordinate				
1	2	0	3	2
00110001	00110010	00110000	00110011	00110010

,	Y-Coordinate			
,	8	3	4	0
00101100	00111000	00110011	00110100	00110000

,	Pen	,	Softkey	,
,	1	,	0	,
00101100	00110001	00101100	00110000	00101100

Status Word				,
1	8	1	6	,
00110001	00111000	00110001	00110110	00101100

Error	CR	LF
0	CR	LF
00110000	00001101	00001010

EOI true

The tablet interprets messages it receives as eight-bit ASCII code. It ignores spaces and carriage returns (CR) and terminates a message when it receives a line feed (LF), a semicolon (;), or a character with EOI set true.

Default Binary Response

The transfer of default binary data is initiated by a read statement from your computer. Refer to the examples of reading binary data found in the end of this chapter. The default binary response is output by the tablet unless you precede a read statement with an HP-GL output instruction. The default binary response is available using any of the bus addresses and is the only output available when the tablet is in the talk-only mode.

The default binary response consists of six bytes of data. The first two bytes are the binary representation of the stylus' X-position. This is followed by two bytes of Y-position data and two bytes of the tablet's current status. The sixth byte in the sequence is always sent with EOI true to terminate the message.

Following is an example of default binary response data containing essentially the same information that took 23 bytes to send in ASCII code in the preceding example.

Default Binary Output

X-Coordinate	Y-Coordinate	Status Word
12032	8340	1816
00101111 00000000	00100000 10010100	00000111 00011000

EOI true

Each two bytes is a two's complement binary number with the most significant bit first. Appendix C contains an explanation of two's complement representation. Each read statement from the computer initiates an output of the six-byte default binary data. For most efficient timing, the computer's read cycles should approximately match the cursor update cycles. Refer to your computer's I/0 documentation for information about the timing of read operations. Refer to "The Cursor Rate Instruction, CR" for more information about setting the cursor rate.

NOTE: When using the binary data transfer mode, do *not* place a 9876 printer on the same HP-IB as the 9111 tablet. You can have a 9876 printer on the same HP-IB when using HP-GL.□

Clearing the Graphics Tablet

There are three bus commands that can be used to clear the graphics tablet: DCL, SDC, and IFC. The effect of each of the commands on the tablet is explained here. Refer to HP-IB documentation or to the "IEEE Standard Digital Interface for Programmable Instrumentation" for information about programming HP-IB commands.

The computer can set all devices on the HP-IB system to a predefined or initialized state by sending the device clear command, DCL. The computer can also set selected devices to a predefined or initialized state by sending a selected device clear command, SDC, along with the addresses of the devices. The basic difference is that devices respond to the SDC command only if they are addressed to listen, whereas the DCL command clears all devices on the bus.

Upon receipt of either a DCL or SDC command, the graphics tablet clears the current digitizing mode, clears any digitized point currently stored, and resets the default parameters. Refer to "The Default Instruction, DF" for a list of the default parameters.

The interface clear command, IFC, is used by the computer to override all bus operations and return the bus to a known quiescent state. When the IFC command is sent to the graphics tablet, the data transaction is halted immediately. The interface clear command *does not* reset the tablet to default parameters.

Sending Service Requests

The active controller is always in charge of the order of events on the HP-IB. If a device on the bus has some information which the controller should be aware of, it can use the service request line to ask for the controller's attention. For example, the tablet might assert service request (SRQ) to tell the computer that its stylus was pressed by the operator and a reading is ready to be taken. This represents a request (*not* a demand),

and it is up to the computer to determine when and how it services the tablet. However, the tablet continues to assert SRQ until it has been satisfied.

The S-mask parameter of the input mask instruction, IM, is used to specify which status word conditions send the service request message. Unless you change the S-mask value from the default setting of 0, the tablet cannot send a service request (refer to "The Input Mask Instruction, IM").

Bit position 6 of the status word is set to 1 when any of the conditions designated by the S-mask are true. Bit position 6 is set to 0 when a status byte message is sent in response to a serial poll or if service is no longer required. Until bit position 6 is reset to 0, no additional service request messages are possible.

Handling Service Requests

Once a service request is received, the computer must locate the device which requested service and determine the reason for the device's request. There are several methods for handling service requests, and they may be used alone or in combination. The following methods are discussed here:

- Interface Status Check
- Interrupt
- Serial Poll
- Parallel Poll

Each of these methods requires high-level input and output statements, such as those available with I/O ROMs on many Hewlett-Packard computers.

Interface Status Check

The HP-IB interface allows you to check the interface status to see if a service request (SRQ) has been received. The disadvantage of this method is that you need to constantly check the interface status. Refer to your computer's I/O documentation for more information about an interface status check.

Interrupt

An interrupt is a signal that suspends ordinary operation of a computer so that some immediate need can be handled. This can be a faster and simpler method of processing, as it eliminates the need for the computer to constantly check the interface status for an SRQ or the status word of the tablet to see if a digitized point or softkey is available.

When the graphics tablet generates a service request, it interrupts the HP-IB interface, which can cause an end-of-line branch in the operation of the program. When the interrupt is received, a suitable interrupt response must be generated. This response is called a service routine. The following outline shows the sequence of events necessary to program for interrupts.

Refer to your computer's I/O documentation for the specific instructions to use in programming these steps.

- Specify where the service routine for an event is and whether the routine is a subroutine or just a program segment.
- Set SRQ-generating conditions in the graphics tablet using the S-mask parameter of the input mask instruction, IM.
- Establish an interrupt mask for SRQ in the interface control register of the computer.
- Enable an interrupt from the interface to ready the computer to respond to previously defined conditions.
- Specify a service routine as follows:
  - — Determine why the tablet generated the SRQ by checking bits in the tablet's status word. You have several choices about how to do this, including doing a serial poll, sending the output status instruction, OS, or sending the output cursor instruction, OC.
  - — Perform desired operations, depending on which condition generated the SRQ.
  - — Enable the interrupt again.
- Return to the program at the point where the interrupt occurred.

Serial Poll

Serial polling is the method of sequentially determining which device connected to the HP-IB has requested service. Only one instrument is checked at a time. Refer to your computer's I/O documentation for information about how to use a serial poll in a program.

A status byte message is sent by the tablet whenever a serial poll message is received. The status byte message consists of the lower eight bits (0 to 7) of the current status word in the tablet. "The Output Status Instruction, OS" describes the status word in detail.

NOTE: In response to an output status instruction, OS, the graphics tablet sends a byte containing information from all 11 bits of the tablet's status word. To ensure consistent status information, it is best to use either serial polls or the OS instruction in a program, not both.□

Remember that the graphics tablet sends a service request only if the conditions specified in the S-mask are satisfied. "The Input Mask Instruction, IM" contains the details about setting the status conditions.

The computer must issue special statements to initiate and terminate a serial poll. Therefore, a serial poll cannot be executed when the tablet is in the talk-only mode.

NOTE: If two serial polls are done within one millisecond of each other, then it is possible for the second one to show bit 6 (RQS) incorrectly set. This condition normally corrects itself within one cursor rate update cycle (1/cursor rate seconds). However, if the beeper is active, the correction can take the duration of the beeper operation for bit 6 to be cleared.□

Parallel Poll

Parallel polling is a method of simultaneously checking status on up to eight instruments on the HP-IB. When the computer initiates a parallel poll, each device returns a status bit via an assigned data line. Refer to your computer's I/O documentation for information about how to use a parallel poll in a program.

Parallel polling can only be done to tablets with an address of 0 through 7. Tablets with address settings from 8 through 30 cannot respond to a parallel poll.

The following table shows the HP-IB address, the parallel poll bit position, and the data line used in response to the parallel poll.

Parallel Poll Response Bits and Data Lines

HP-IB Address	Parallel Poll Bit Position	HP-IB Data Line Number	
0	7	8	
1	6	7	
2	5	6	
3	4	5	
4	3	4	
5	2	3	
6	1	2	← TABLET PRESET ADDRESS
7	0	1	

The P-mask parameter of the input mask instruction, IM, is used to specify which status word conditions result in a logical 1 response to a parallel poll. The P-mask bit value must be changed from the default value of 0 for the tablet to respond to a parallel poll. Refer to "The Input Mask Instruction, IM" for details.

Positive responses to parallel polls continue to occur until all bits of the status word included in the P-mask value have been reset to 0. "The Output Status Instruction, OS" describes the status word in detail. The tablet does not respond to a parallel poll in talk-only mode.

Asserting EOI

Normally, data messages sent over the HP-IB are sent using the standard ASCII code and are terminated by the ASCII line feed character (decimal 10). However, the EOI (End or Identify) line can be used to mark the end of the data message. When the listeners detect that the EOI line is true, they recognize that the byte on the data lines is the last one of the data message.

The graphics tablet asserts EOI with the last character of any data message, whether binary default response or ASCII, to terminate the data message. EOI can also be used to terminate HP-GL instructions sent to the tablet. Refer to HP-IB documentation or to the "IEEE Standard Digital Interface for Programmable Instrumentation" for more information about asserting EOI. END is an HP-IB message that can be used in the talking or listening functions to assert EOI. The EOI line is also used during an identify (parallel poll) sequence.

Sending and Receiving ASCII Data

Computer-to-Tablet

Transmitting ASCII data from a computer to the graphics tablet is typically accomplished using I/O statements such as WRITE, PRINT, or OUTPUT. The following examples of sending program data to the tablet from various computers are only intended to illustrate the necessity for understanding the I/O statement protocol implemented by your computer.

Each of these examples causes the graphics tablet to sound a tone. In a program, this tone is typically used to indicate that a point is successfully digitized. Each of the computers used in these examples automatically sends a line feed (LF) at the end of the output statement which acts as a terminator for the HP-GL instructions.

HP-85, HP-86, HP-87, HP 9826, and HP 9836 BASIC Example

```
10 PRINTER IS 706
20 PRINT "BP20,100,3;BP22;BP24;
   BP26"
30 END
```

NOTE: To use the above example on the HP-86, you must change Line 10 so that the PRINTER IS statement uses a select code other than 7. For example, you can substitute the following for Line 10 to run the program on the HP-86:

```
10 PRINTER IS 606
              |
              └──Select code
```

In order to interface the tablet to the HP-86, you may need to change the HP-IB interface select code to avoid duplication of the HP-86 printer/disc interface select code. Electrical damage may result if you have two interfaces set to the same select code. The select code for the HP-86 printer/disc interface is set to 7 and cannot be changed. To operate the tablet with an HP-86, you must have an HP-IB Interface card (HP 82937A), which may also be preset to 7 but can be changed. You must be sure the select code in the interface card is a number other than 7; for example, 6. Refer to the section about connecting the tablet to the HP-86 in the Series 80 Programming Guide for more complete information.□

HP 9845 BASIC Example

```
10      PRINTER IS 7,6
20      PRINT "BP20,100,3;BP22;BP24;BP26"
30      END
```

Refer to the HP 1000 FORTRAN example of receiving ASCII data from the tablet for the steps necessary to create, compile, load, and run a FORTRAN program. In the WRITE statements in the following program, 41 is the LU (logical unit) number assigned to the tablet.

```
0002            PROGRAM BEEPS
0003            WRITE (41,30)
0004   30       FORMAT ("BP20,100,5")
0005            WRITE (41,60)
0006   60       FORMAT ("BP22")
0007            WRITE (41,90)
0008   90       FORMAT ("BP24")
0009            WRITE (41,110)
0010   110      FORMAT ("BP26")
0011            END
```

Tablet-to-Computer

Outputting data from the tablet to the computer is typically accomplished using I/O statements such as READ, INPUT, or ENTER. The following examples of obtaining output data from the tablet using various computers are only intended to illustrate the necessity for understanding the I/O statement protocol implemented on your computer. The following examples require an I/O ROM or external I/O capability. Each of these examples causes the tablet to output the current positions of P1 and P2 and the tablet identifier string to the computer.

HP-85, HP-86, HP-87, HP 9826, and HP 9836 BASIC Example

```
10 PRINTER IS 706
20 PRINT "OP"

30 ENTER 706 ; X1,Y1,X2,Y2
40 PRINT "OI"
50 ENTER 706 ; I$
60 DISP X1,Y1,X2,Y2,I$
70 END
```

NOTE: To use the above example on the HP-86, you must change Line 10 so that the PRINTER IS statement uses a select code other than 7. For example, you can substitute the following for Line 10 to run the program on the HP-86:

```
10 PRINTER IS 606
              └── Select code
```

You must also change Lines 30 and 50 to ENTER 606 . . . . Refer to the note with the computer-to-tablet example for an explanation of why you need to change the select code.□

The default positions of P1 and P2 and the tablet's identification are displayed on the computer's screen:

```
 400                    400
 11632                  8340
9111T
```

HP 9845 BASIC Example

```
10    PRINTER IS 7,6
20    PRINT "OP"
30    ENTER 706;X1,Y1,X2,Y2
40    PRINT "OI"
50    ENTER 706;I$
60    DISP X1;Y1;X2;Y2;I$
70    END
```

The default positions of P1 and P2 and the tablet's identification are displayed on the computer's screen:

```
400  400  11632  8340 9111T
```

The following example shows you what information to enter at a terminal so you can create, compile, load, and run a FORTRAN program on the HP 1000. The listing shows the information you must enter in color and the information that is displayed to you by the computer in black. Explanations appear after the listing for those lines with numbers on the right.

HP 1000 FORTRAN Example

```
WAITING FOR INPUT
:RU,EDITR                                                              (1)
SOURCE FILE?
/O                                                                     (2)
/ FTN4                                                                 (3)
          PROGRAM ASCII                                                (4)
          DIMENSION IPTS(12),I(3)
          WRITE (1,99)                                                 (5)
   99     FORMAT (" FORTRAN PROGRAM TO OUTPUT P1, P2, AND IDENTIFIER")
          WRITE (56,100)                                               (6)
   100    FORMAT ("OP")
          READ (56,101)IPTS
   101    FORMAT (12A2)
          WRITE (56,102)
   102    FORMAT ("OI")
          READ (56,103) I
   103    FORMAT (3A2)
          WRITE (1,104)IPTS,I
   104    FORMAT (12A2,3A2)
          END
EOF
/EC &ASCII                                                             (7)
:RU,FTN4,&ASCII::13,1,%ASCII::13                                       (8)

 PAGE 0001  FTN.   9:58 AM  THU.,  2  DEC., 1982

0001  FTN4

  FTN4 COMPILER: HP92060-16092 REV. 2026 (800423)

  **  NO WARNINGS **  NO ERRORS **   PROGRAM = 00114      COMMON = 00000


 $END FTN4: NO DISASTRS   NO ERRORS   NO WARNINGS
:RU,LOADR,,%ASCII::13,1                                                (9)
  ASCII             32012 32173

  .DIO.             32174 32462  24998-1X331 REV.2140 810505
     .    .    .    .    .    .    .    .    .    .    .
     .    .    .    .    .    .    .    .    .    .    .
  .PACK             40514 40624  92084-1X076 REV.2121 750701

       5  PAGES RELOCATED
       5  PAGES REQUIRED

   LINKS:BP   PROGRAM:BG   LOAD:TE   COMMON:NC

    /LOADR:ASCII  READY AT  9:58 AM   THU.,  2  DEC., 1982
    /LOADR:$END
```

HP 1000 FORTRAN Example (Continued)

```
:RU,ASCII                                                    (10)
 FORTRAN PROGRAM TO OUTPUT P1, P2, AND IDENTIFIER
400,400,11632,8340      9111T                                (11)
```

Following are the explanations corresponding to the numbered lines in the preceding listing.

(1) This command runs the EDITR program so you can create or edit your source program.

(2) Source file 0 indicates that you are creating a new source file. If you are editing an existing source file, enter the file name here.

(3) Once you enter this line to tell the computer what version of FORTRAN you are using, you can enter the remainder of the program.

(4) ASCII is the program name.

(5) 1 is the LU (logical unit) number for the terminal.

(6) 56 is the LU for the tablet in this example.

(7) This command allows you to end and create a new source file called &ASCII. If you are editing an existing program, you can use the ER command to end and replace the existing source file.

(8) This command compiles the source program. The source file name is &ASCII, the disk cartridge number where the source file is stored is 13, the LU where the listing is to be printed is 1 (terminal), the relocatable file name is %ASCII, and the disk cartridge number where the relocatable file is stored is 13.

(9) This command loads the relocatable code. The relocatable file name is %ASCII, the disk cartridge number where the relocatable file is stored is 13, and the LU where the listing is to be printed is 1 (terminal).

(10) This command runs the program called ASCII.

(11) The program output consists of the default positions of P1 (400,400), P2 (11632,8340), and the tablet identifier (9111T).

Receiving Binary Default Response Data

Tablet-to-Computer

Sample programs to input X,Y coordinate and status information from the 9111, using the default binary transfer are included here for several computers. Your computer's manuals that contain information about HP-IB interfacing and I/O programming are the best source for detailed information about using the default binary transfer with your computer. The following examples require an I/O ROM or external I/O capability.

HP-85, HP-86, HP-87, HP 9826, HP 9836, and HP 9845 BASIC Example

```
10 ENTER 706 USING "#,W,W,W";X,Y,P
20 DISP X,Y,P
30 END
```

NOTE: To use the above example on the HP-86, you must change Line 10 so that the ENTER statement uses a select code other than 7. For example, you can substitute the following for Line 10 to run the program on the HP-86:

```
10 ENTER 606 USING "#,W,W,W";X,Y,P
           └── Select code
```

Refer to the note with the computer-to-tablet example in "Sending and Receiving ASCII Data" for an explanation of why you need to change the select code.□

The X,Y coordinates of the current stylus position and the status word are displayed on the computer's screen:

```
10028    7628    1816
```

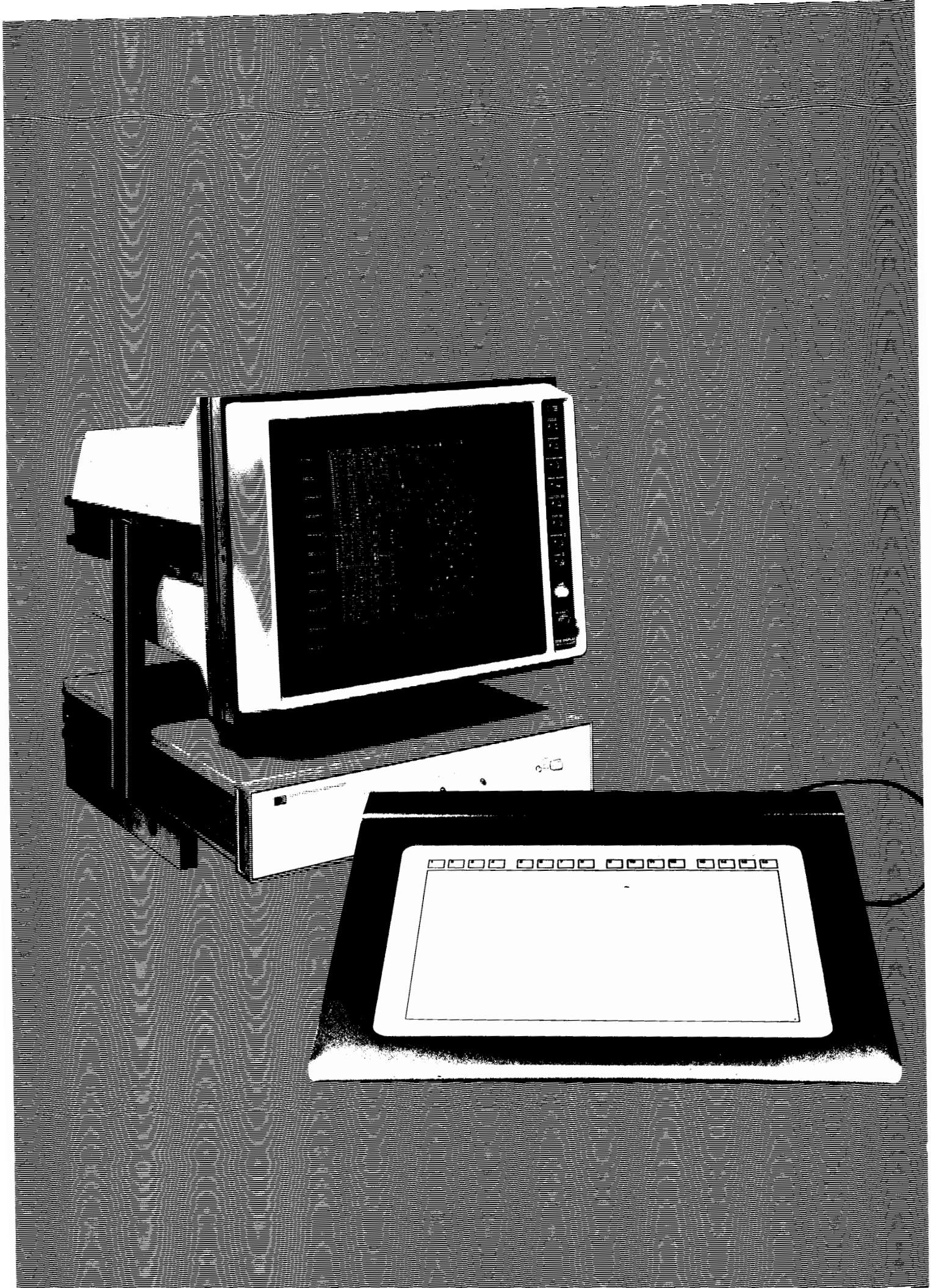
Refer to the HP 1000 FORTRAN example of receiving ASCII data from the tablet for the steps necessary to create, compile, load, and run a FORTRAN program. In the call to the system utility REIO in the program below, the LU (logical unit) of the tablet is 41. The 2100B is added to the LU of the device in order to handle binary data input without terminating on a carriage return (CR) or line feed (LF). The 3 indicates three words of data to be input to the array I. The WRITE statement sends the X, Y, and status information to a device at LU 1 (normally a terminal).

HP 1000 FORTRAN Example

```
0001   FTN4
0002           PROGRAM BINRY
0003           INTEGER I(3)
0004           EXTERNAL REIO
0005   10      CALL REIO (1,41+2100B,I,3)
0006           WRITE (1,50) I(1),I(2),I(3)
0007           GOTO 10
0008   50      FORMAT (3I7)
0009           END
```

The X,Y coordinates of the current stylus position and the status word are displayed on the terminal's screen:

```
2956        7712        1304
```



Chapter 5
The HP 1351S Graphics Display System

Chapter 5 describes how to use the 9111 graphics tablet with the Hewlett-Packard 1351S graphics display system. In addition to the HP-GL (Hewlett-Packard Graphics Language) instruction set, the graphics tablet contains six additional instructions that are designed for use with the 1351S graphics display system. This chapter explains these instructions and their use in detail.

What Is the Graphics Display System?

The HP 1351S graphics display system consists of a graphics generator (1351A) and one of a series of large screen displays (13XX). The tablet accepts special instructions from a host computer which allow it to talk through the graphics generator directly to the display. Once the host computer initiates the instructions, it is free for other tasks. The direct path from tablet to display increases performance, speeds up responses, and minimizes computer overhead, allowing a single computer to support several graphics workstations efficiently.

What Other Documentation Do I Need?

This chapter only discusses the features of the graphics tablet that are for use with the 1351S system. To understand how to program and operate the entire system, refer to the remainder of this 9111 Programming Manual, the 1351A Operating and Programming Manual, and Operating and Service Manuals for the 13XX series CRT display and the 1351A graphics generator.

What Are the Special Features?

A high-performance, interactive graphics system results from the addition of a graphics tablet to a graphics display system. When you use the 9111 tablet with the 1351S graphics display system, you have all the capabilities of a 9111 tablet used with other systems, plus you can do the following:

- Customize the cursor by defining it as a special character or by selecting any size cross-hair cursor from a single dot to the full screen.
- Stretch (rubber band): the corner of a rectangle, one line from an endpoint, or two lines from a common endpoint.

- Specify that a single stretched, or rubber-band line will always be horizontal with no vertical movement, or vice versa.
- Take advantage of the tablet's maximum data rate of 60 updates per second because data transfer from the tablet to the generator is fast.
- Send X,Y coordinates automatically scaled to the resolution of the display.

General Programming Information

This part of the chapter discusses the following general programming issues that may help you in programming the tablet with the graphics display system:

- the programming sequence for all enable echo (EE) instructions;
- additional programming considerations when using EE instructions;
- assigning the tablet as the talker and the generator as the listener on the Hewlett-Packard Interface Bus (HP-IB);
- generating interrupts to the computer on the HP-IB; and
- using the tablet softkeys.

Each EE instruction is explained in detail later in this chapter. The input and output statements used to send instructions to the tablet and control HP-IB operations are highly computer-dependent. Therefore, these operations are discussed in a general way in this chapter. Refer to the programming manuals for your generator and computer for details about which statements perform each operation.

Programming Sequence

The recommended sequence for programming all of the enable echo (EE) instructions follows.

1. Place the necessary user-supplied data in generator memory locations 0 through 4.
2. Set the pen in the generator (beam on or off).
3. Send the appropriate EE instruction to the tablet.
4. Configure the HP-IB so that the tablet is the talker and the graphics generator is the listener.
5. Enable a service request on the HP-IB to cause the computer to handle an interrupt from the tablet. The most common conditions for which you would send a service request from the tablet are that the stylus has been pressed, a digitized point is available, or a softkey has been selected. Chapter 4 of this manual explains more about service requests and interrupts.

Additional Considerations

The tablet updates information in generator memory locations 1 through 4. After a figure is drawn on the display screen and before a new figure is started, the current figure must be stored elsewhere in the generator memory if it is to remain on the display. If the current figure is not stored

elsewhere, the tablet replaces the information in memory locations 1 through 4 with coordinates for the new figure, and the current figure is erased. Any information you wish to keep should be stored in a file or in consecutive memory locations in the generator where it cannot be erased or written over.

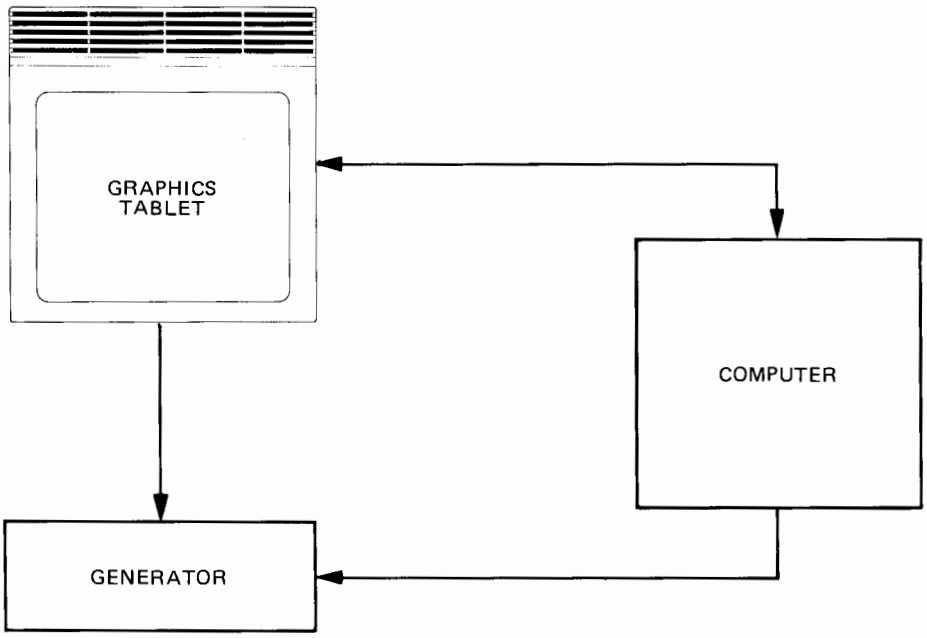
Before starting a new function, it is good programming practice to erase the generator memory file into which the tablet has been storing data. For example, if you have been drawing a rectangle which uses memory locations 0 through 4, and start drawing rubber-band lines which only use memory locations 0 through 2, the leftover rectangle data in locations 3 and 4 may be displayed with the rubber-band line.

Assigning a Talker and Listener

Since the tablet is designed to talk directly to the generator via the HP-IB, it is necessary to configure the tablet as the bus talker and the graphics generator as the listener before transfer of data can occur. Following is the necessary sequence of operations.

1. Tell all the instruments on the bus to UNLISTEN using the ASCII character ?.
2. Set the tablet as the talker. The tablet has a factory-set HP-IB address of 06.
3. Set the graphics generator as the listener. The generator has a factory-set address of 18.

After the bus is configured as shown above, the computer is free to carry on other types of operations until needed by the tablet. When the tablet needs attention, it generates a service request causing the computer to resume responsibility as the bus talker. The following diagram illustrates the computer's interaction with the tablet and generator.



Computer Interaction with Assigned Listener and Talker

When the tablet is assigned as the talker and is sending information directly to the generator, the computer is free to carry on other tasks, or to

talk to other instruments on select codes different from that of the tablet and graphics generator. The computer cannot update the generator at the same time that the tablet is doing so. If the system depends on the computer to update dynamic items on the display screen, the computer cannot do this while the tablet is interacting with the generator. Consequently, when control returns to the computer, the display screen picture may have changed radically.

Generating Interrupts

Once the graphics tablet is transferring data directly to the graphics generator, the computer is not needed for interaction until a service request is generated on the HP-IB. When the service request is generated, the computer takes control of the bus and routes all data from the tablet through itself to the generator or hardcopy device, if any.

Suppose that you are moving the cursor to a point on screen and want that point to define the start of a rectangle when you press down on the stylus. At that time, several things need to happen:

1. The tablet must stop sending data to the generator.
2. The computer must assume control of the HP-IB.
3. The computer tells the tablet to output the digitized point when the stylus is pressed.
4. The computer reads the digitized point and uses it as the fixed corner of the rectangle.
5. The computer stores the digitized point from the tablet in memory locations 0 and 4 in the generator.
6. The computer sends a rubber-band rectangle instruction, EE4, to the tablet. (Refer to details of the EE4 instruction later in this chapter.)
7. The bus is configured with the tablet as talker and the generator as listener to allow the tablet to send the moving points of the rectangle to generator memory locations 1 through 3.
8. The interrupt is enabled, allowing the next service request to be generated.

A service request or interrupt is necessary anytime computer interaction is needed. Examples of functions that need computer interaction are reading a digitized point from the tablet, doing a calculation before plotting, or performing a certain function based on a softkey selection.

Using the Softkeys

The softkeys on the tablet can be used for a number of functions with the graphics display system, as they can for any other system. The tablet interprets a softkey selection by number when the pen is pressed in the corresponding box. Your program can find out from the tablet which one of the softkeys was selected. Depending on the number returned, your program can then go to a routine that relates the softkey number to the program function. For example, you can use softkey 1 to define a rectangle, softkey 2 to generate a single rubber-band line, softkey 3 to create a symbol cursor, etc.

Tablet Instructions Used with the Graphics Display System

The following discussion concerns the six enable echo (EE) instructions available for use with the HP 1351S graphics display system. In order to understand and effectively use these six instructions, you must have a thorough knowledge of how to program the 1351A graphics generator. Refer to the generator's Operating and Programming manual for an explanation of how to program the generator.

The following table lists the instructions and their descriptions.

Instruction	Description
EE0	Moves a previously defined ASCII-character cursor around the screen.
EE1	Generates single or double rubber-band lines.
EE2,X,Y	Forces a single rubber-band line to always be horizontal.
EE3,X,Y	Forces a single rubber-band line to always be vertical.
EE4,X,Y	Produces a rubber-band rectangle.
EE5,S	Draws a cross-hair cursor of a specified size.

The Symbol Cursor Instruction, EE0

DEFINITION: The symbol cursor instruction, EE0, causes the graphics tablet to output the X,Y coordinates of the current stylus position to the display.

USES: The instruction is used to move a previously defined ASCII-character cursor around the screen. You can create a cursor to suit your own application.

SYNTAX: EE0

OUTPUT: The tablet sends the following instruction string to the graphics generator:

FL1,:PA(X),(Y);LF

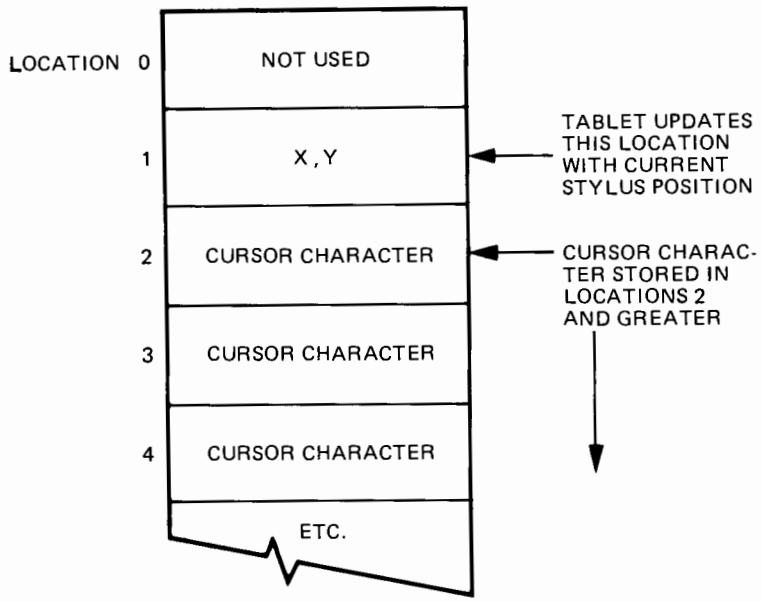
where X and Y are the coordinates of the current stylus position. Refer to the Graphics Generator Operating and Programming Manual for an explanation of the FL (find location) and PA (plot absolute) instructions. Note that LF is a line feed.

EXPLANATION: When the symbol cursor instruction, EE0, is given, the tablet must be configured as the HP-IB talker and the generator as the listener. The tablet then enters coordinates directly into the generator memory (location 1) at 60 times per second without computer interaction.

The cursor must be entered into location(s) 2 and greater in the generator memory before the EE0 instruction is sent to the tablet. The cursor can be created from the 1351A Modified ASCII Character Set as listed in Appendix E of the 1351A Operating and Programming Manual. The size of the characters in the cursor is controlled by the character size instruction, CS, for the generator.

It is possible to make a cursor an entire word, a string of numbers, or a symbol. Each letter, number, or ASCII symbol requires one location in memory. For example, to use the word "banana" as a cursor requires eight locations in memory: one location for each letter of "banana," one location for the current stylus position, and one unused location (location 0). Refer to the following diagram of the generator memory configuration.

Generator Memory



It is also possible to place characters on top of one another on the display screen. To do so, store a backspace command (decimal 8) in memory locations between each of the characters, or use characters which do not cause the generator to advance the beam to the next character starting point.

A double-bright cursor character can be generated by duplicating the cursor characters in consecutive memory locations in the generator. To draw a cross-hair cursor (+), use the vertical tick mark (decimal 11) with the horizontal tick mark (decimal 12) overlaid. The tick mark characters do not advance the generator beam.

The working bounds of the generator are automatically changed to 12,1008 in both the X- and Y-directions when using the EE0 instruction. This buffer keeps cursor characters from being distorted on the display screen when using characters up to size 2. Consequently, a point on the display screen which is in the X,Y range of 1 through 11 or 1008 through 1020 cannot be reached with the symbol cursor.

After putting the pen down (display beam on) and drawing a cursor character, remember to lift the pen again (display beam off) or a line will be drawn from the old cursor position to the new. This happens because the generator searches through its memory sequentially (refer to the preceding memory diagram).

With the EE0 instruction, the updated cursor position is stored in location 1. The generator receives a plot absolute instruction, PA, from the tablet along with the X,Y coordinates of the current stylus position and moves the beam to that point on the screen. The generator then goes to location

2, where the cursor character is stored, and draws that character. If there are more cursor characters in locations 3 and greater, they are drawn at the same point.

When the generator has drawn all of the cursor characters, the pen must be lifted (display beam off) before the generator starts at location 1 again, or a line will be drawn from the current point on the screen to the next updated point.

Recommended Programming Sequence

Following is a recommended programming sequence for generating a symbol cursor.

- Using generator instructions, tell the generator to:
  - — Erase file 1 to clear out any extraneous information from the preceding function.
  - — Find location 1 where the tablet will put the coordinates of the current stylus position.
  - — Place the pen in the "up" position to prevent the generator from drawing a line from the cursor starting point (0,0) to the current stylus position.
  - — Select the size of the cursor with the character size command.
  - — Enable the pen.
  - — Draw the cursor character(s) at the coordinate point in location 1 (provided by the tablet).
  - — Raise the pen so a line will not be drawn between the current point and the next updated point.
- Send the EE0 instruction to the tablet to initiate the function.
- Go to a "talk" subroutine that assigns the generator as listener and the tablet as talker and starts the data transfer from the tablet to the generator.
- Data transfer stops when the stylus is pressed. You can set up an interrupt routine to check for the press of the stylus and ask for the location of that digitized point. Chapter 4, HP-IB Interfacing, contains more information about interrupts.

The Rubber-Band Line Instruction, EE1

DEFINITION: The rubber-band line instruction, EE1, generates single and double rubber-band lines.

USES: The instruction allows you to experiment with different lengths and locations of lines and endpoints without using the computer's memory and time to constantly calculate and draw each separate line or endpoint.

SYNTAX: EE1

OUTPUT: The tablet sends the following instruction string to the graphics generator:

FL1,:PA(X),(Y);LF

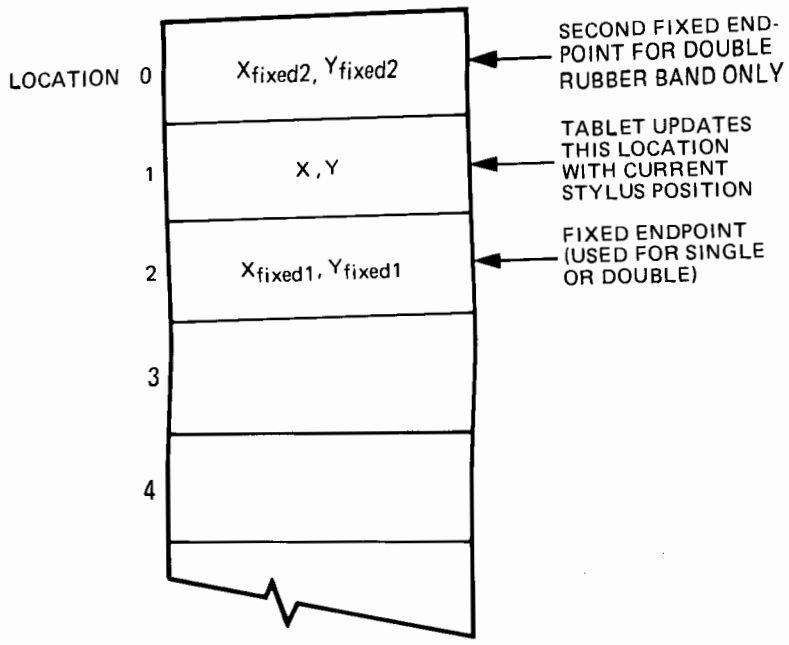
where X and Y are the coordinates of the current stylus position. Refer to the Graphics Generator Operating and Programming manual for an explanation of the FL (find location) and PA (plot absolute) instructions. Note that LF is a line feed.

EXPLANATION: Following are explanations for creating single and double rubber-band lines and for adding an alpha cursor to either.

Single Rubber-Band Line

A single rubber-band line has a fixed point at one end while the other end moves with the stylus. The fixed endpoint of the line is set the first time the stylus is pressed on the tablet's surface. The programmer must store the fixed endpoint in location 2 of the generator memory and leave memory location 0 empty. Refer to the following diagram of the generator memory configuration.

Generator Memory



As the stylus is moved away from the fixed point, a line is drawn from the current stylus position to the fixed endpoint. The EE1 instruction causes the moving endpoint to be stored in location 1 of generator memory. This continues until the stylus is pressed the second time, fixing the second endpoint. A second single rubber-band line can be started at this second point, or a completely different line can be started by pressing the stylus a third time.

Double Rubber-Band Line

A double rubber-band line has two fixed ends, both connected to a single moving point. The first pen press defines one fixed end, and the second pen press fixes the second end. The programmer must store the first fixed endpoint in location 2 of the generator memory and the second fixed end-point in location 0 (refer to the memory diagram). Now the stylus can move the mobile point around these two fixed ends. The EE1 instruction causes the moving point to be stored in location 1 of the generator memory.

Adding an Alpha Cursor

It is possible to add an alpha cursor to the end of a single rubber-band line by putting cursor characters in locations 2 and 3 of the generator

memory. An alpha character can be placed in the middle of a double rubber-band line by placing characters in locations 2 and 3 and writing the second fixed endpoint into location 4.

Recommended Programming Sequence for Single Rubber-Band Lines

The following programming sequence is recommended for generating a single rubber-band line.

- Call a "cursor" subroutine that has been programmed to allow the user to see where the pen is positioned on the display screen and where the line will begin when the stylus is pressed. The "cursor" subroutine should call a "pen press" subroutine to obtain the X,Y coordinates of the location where the stylus is pressed.
- Tell the generator to find a location, called "M" for this example, and write the X,Y coordinates of the first pen press there with the beam off. This stores the information in a location where it will not be written over when a new line starts.
- Increment the location pointer "M" by 1.
- Tell the generator to go to memory location 0 and write the X,Y coordinates there with the beam off. This is the fixed endpoint of the line.
- Send the EE1 instruction to the tablet to initiate the function.
- Go to a "talk" subroutine that assigns the generator as listener and the tablet as talker and returns the X,Y coordinates after the stylus is pressed. You can use an interrupt routine to check for the press of the stylus, which indicates the end of the data transfer.
- Tell the generator to find location "M" and write the X,Y coordinates there with the beam on.

Recommended Programming Sequence for Double Rubber-Band Lines

The following programming sequence is recommended for generating a double rubber-band line.

- Call a "cursor" subroutine that has been programmed to allow the user to see where the pen is positioned on the display screen and where the line will begin when the stylus is pressed. The "cursor" subroutine should call a "pen press" subroutine to obtain the X,Y coordinate of the location where the stylus is pressed.
- Store the coordinates of the first pen press in an array to save them for use later in the subroutine.
- Call the "cursor" subroutine to obtain the second fixed point of the line.
- Erase file 1 (the memory locations the tablet uses), set the generator memory pointer to location 0, and place the first fixed endpoint there with the beam off.
- You can place information in location 1 of generator memory to prevent a momentary line from appearing between point 0,0 and the X,Y point in location 2 until the tablet can update location 1 with a set of stylus position coordinates.

- Place the second endpoint coordinates in location 2 with the beam on. This causes a line to be drawn between the first and second endpoints. Then store the second endpoint in an array for later use.
- Send the EE1 instruction to the tablet to initiate the function.
- Go to a "talk" subroutine that assigns the generator as listener and the tablet as talker and returns the X,Y coordinates after the stylus is pressed. You can use an interrupt routine to check for the press of the stylus, which indicates the end of the data transfer.
- Tell the generator to find a location, called "M" for this example, and write the X,Y coordinates of the first endpoint there with the beam off. This stores the information in a location where it will not be written over when a new line starts.
- Write the second endpoint in generator memory location M + 1 with the beam on. This completes the redrawing of the line beginning at location M.
- Increment the location counter so that another figure will not be drawn over this line.

The Forced Horizontal Line Instruction, EE2

DEFINITION: The forced horizontal line instruction, EE2, forces a single rubber-band line, which has one end fixed and the other end moving with the stylus, to always be horizontal, with no vertical movement.

USES: The instruction is useful in applications such as drafting where each point must lie in a vertical or horizontal plane.

SYNTAX: EE2,X,Y

where X and Y are the coordinates of the fixed endpoint.

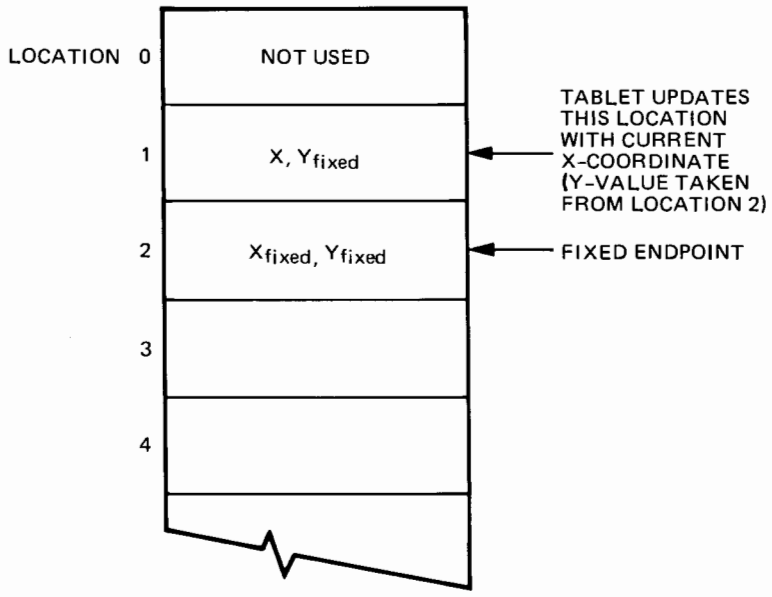
OUTPUT: The tablet sends the following instruction string to the graphics generator:

FL1,:PA(X),(Y);LF

where the X-value is used to establish the moving end of the line. Refer to the Graphics Generator Operating and Programming manual for an explanation of the FL (find location) and PA (plot absolute) instructions. Note that LF is a line feed.

EXPLANATION: A forced horizontal line has the same Y-value, no matter where on the tablet's surface the pen is moved. One end of the line is fixed when the pen is first pressed on the tablet's surface. The programmer must send the coordinates of that point to the tablet as the X,Y coordinates of the EE2 instruction.

All subsequent movement of the stylus produces a line with the same Y-value as the fixed endpoint. This means that the only value that changes in the generator memory is the X-value sent by the tablet to memory location 1. The Y-coordinate of the stylus position on the tablet is taken to be the same as that provided by the programmer in location 2 of the generator memory. Refer to the following diagram of the generator memory configuration.



When redrawing a forced horizontal line in the generator memory, the programmer is responsible for fixing the Y-coordinates. That is, you must use the Y-value of the first endpoint for the second endpoint as well.

The Forced Vertical Line Instruction, EE3

DEFINITION: The forced vertical line instruction, EE3, forces a single rubber-band line, which has one end fixed and the other end moving with the stylus, to always be vertical, with no horizontal movement.

USES: The instruction is useful in applications such as drafting where each point must lie in a vertical or horizontal plane.

SYNTAX: EE3,X,Y

where X and Y are the coordinates of the fixed endpoint.

OUTPUT: The tablet sends the following instruction string to the graphics generator:

FL1,:PA(X),(Y);LF

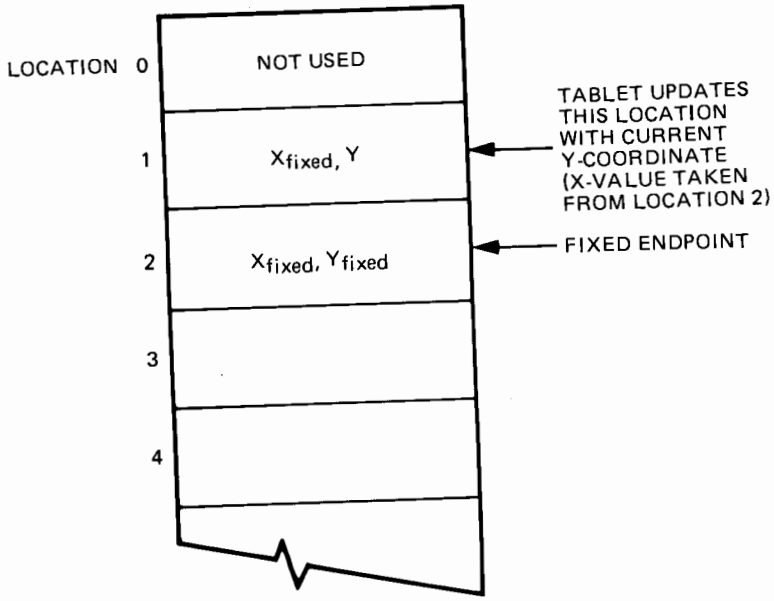
where the Y-value is used to establish the moving end of the line. Refer to the Graphics Generator Operator and Programming manual for an explanation of the FL (find location) and PA (plot absolute) instructions. Note that LF is a line feed.

EXPLANATION: A forced vertical line has the same X-value, no matter where on the tablet's surface the pen is moved. One end of the line is fixed when the pen is first pressed on the tablet's surface. The programmer must send the coordinates of that point to the tablet as the X,Y coordinates of the EE3 instruction.

All subsequent movement of the stylus produces a line with the same X-value as the fixed endpoint. This means that the only value that changes in the generator memory is the Y-value sent by the tablet to memory location 1. The X-coordinate of the stylus position on the tablet is taken to

memory configuration.

Generator Memory



When redrawing a forced vertical line in the generator memory, the programmer is responsible for fixing the X-coordinates. That is, you must use the X-value of the first endpoint for the second endpoint as well.

The Rubber-Band Rectangle Instruction, EE4

DEFINITION: The rubber-band rectangle instruction, EE4, produces a rubber-band rectangle.

USES: The instruction allows you to experiment with three different moving corners of a rectangle with one fixed endpoint without using the computer's memory and time to constantly calculate and draw separate rectangles in separate locations.

SYNTAX: EE4,X,Y

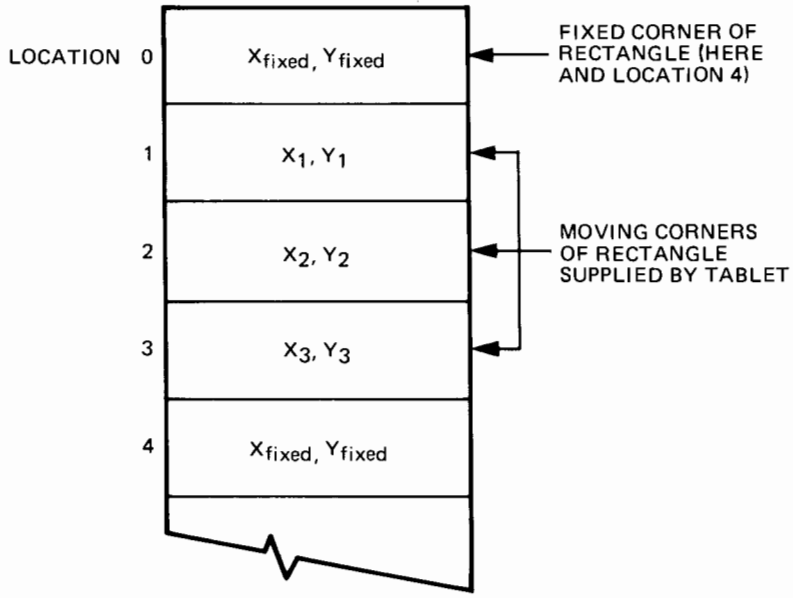
where X and Y are the coordinates of the rectangle's fixed corner.

OUTPUT: The tablet sends the following instruction string to the graphics generator:

FL1,:PA(X1),(Y1);(X2),(Y2);(X3),(Y3);LF

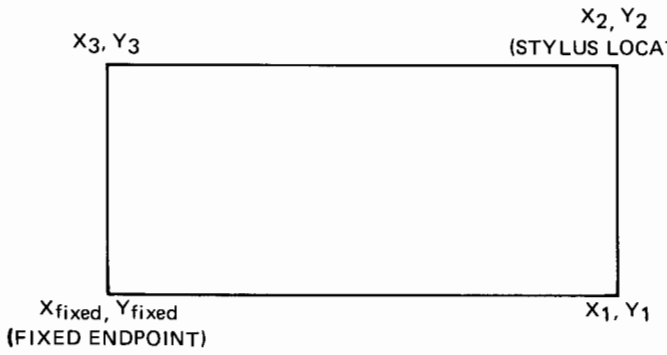
where $(X_1,Y_1),(X_2,Y_2),(X_3,Y_3)$ are the three moving corners of the rectangle. Refer to the Graphics Generator Operating and Programming manual for an explanation of the FL (find location and PA (plot absolute) instructions. Note that LF is a line feed.

EXPLANATION: A rectangle is defined by 5 points, the first and last having the same coordinates. The first and last points form the fixed corner of the rectangle. They are established by pressing the pen on the tablet's surface, and they must be placed in memory location 0 and 4 of the generator memory by the programmer. Refer to the following diagram for the generator memory configuration.



Generator Memory

The tablet calculates the three moving corners (points) based on movement of the stylus and places them into locations 1 through 3 of the generator memory. The corner opposite the fixed corner is the location of the stylus. Refer to the following diagram locating the corners of a rubber-band rectangle.



Corners of a Rubber-Band Rectangle

Recommended Programming Sequence

The following programming sequence is recommended for a rubber-band rectangle.

- Call a "cursor" subroutine that has been programmed to allow the user to see where the pen is positioned on the display screen and where the rectangle will begin when the stylus is pressed. The "cursor" subroutine should call a "pen press" subroutine to obtain the X,Y coordinates of the location where the stylus is pressed.
- Write the X,Y coordinates of the location where the stylus was pressed into locations 0 and 4 of the generator memory. These are the start and stop points of the rectangle.
- Store X and Y in an array so they are not lost when the tablet sends new values the next time the stylus is pressed.

- Send the EE4 instruction to the tablet to initiate the function. Include the coordinates of the first press of the stylus as the parameters of the instruction to establish the fixed corner of the rectangle.
- Go to a "talk" subroutine that assigns the generator as listener and the tablet as talker and returns the X,Y coordinates after the stylus is pressed. You can use an interrupt routine to check for the press of the stylus, which indicates the end of the data transfer.
- Write the finished rectangle in generator memory, in a location called "M" for this example, so that it cannot be written over when a new rectangle starts. Note that when the "talk" subroutine is called, the third corner of the rectangle is fixed when the pen is pressed the second time. The other two corners of the rectangle can then be calculated by the tablet based on the first and third corners (first and second presses of the stylus).
- Increment the location point "M" by 5.

The Drawn Cursor Instruction, EE5

DEFINITION: The drawn cursor instruction, EE5, draws a cross-hair cursor of a specified size on the screen.

USES: The instruction is used to draw a cursor to the size that is best suited to your application.

SYNTAX: EE5,S

where S is the length of one line (horizontal or vertical) of the cursor.

DEFAULT: S is 1020 (full screen).

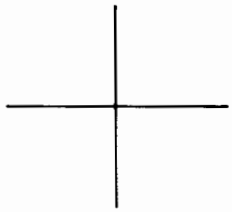
OUTPUT: The tablet sends the following instruction string to the graphics generator:

```
FL1,:PE0,:PA(X1),(Y1);:PE1,:PA(X2),(Y2);:
  PE0,:PA(X3),(Y3);:PE1,:PA(X4),(Y4);LF
```

where the X- and Y-values cause the cursor to be drawn at the current stylus position. Refer to the Graphics Generator Operating and Programming Manual for an explanation of the FL (find location), PA (plot absolute), and PE (pen enable) instructions. Note that LF is a line feed.

EXPLANATION: This instruction draws a cursor consisting of vertical and horizontal lines that cross in the middle, as in the example below:

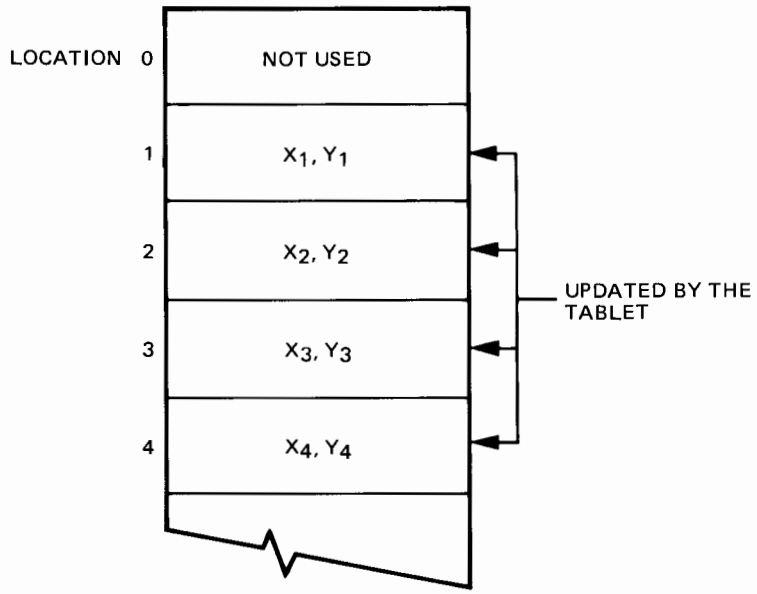
Cross-hair Cursor



The S parameter of the EE5 instruction specifies the length of one line. A cursor any size from a single dot (S = 1) to the full screen (S = 1020) can be specified. If S is not specified, a full-screen cursor is drawn.

The tablet updates generator memory locations 1 through 4 with the EE5 instruction. Refer to the following diagram for the generator memory configuration.

Generator Memory



The aesthetics of the symbol cursor drawn with the EE0 instruction are generally better. However, the EE5 instruction allows the operator to access any portion of the display screen within the bounds of the generator. Remember that the symbol cursor (EE0) is kept 12 units from the border to prevent character distortion and that the size of the EE0 cursor is controlled by the generator's character size instruction, CS.

Recommended Programming Sequence

Following is a recommended programming sequence for generating a drawn cursor:

- Send the EE5 instruction to the tablet and specify the size of the cursor with a parameter from 1 to 1020.
- Go to a "talk" subroutine that assigns the generator as listener and the tablet as talker and starts the data transfer from the tablet to the generator.
- Data transfer stops when the stylus is pressed. You can set up an interrupt routine to check for the press of the stylus and ask for the location of the digitized point. Chapter 4, HP-IB Interfacing, contains more information about interrupt routines.

Appendix A
An HP-IB Overview

The Hewlett-Packard Interface Bus (HP-IB) provides an interconnecting channel for data transfer between devices on the HP-IB.

The following list defines the terms and concepts used to describe HP-IB (bus) system operations.

HP-IB System Terms

1. **Addressing** — The characters sent by a controlling device specifying which device sends information on the bus and which device(s) receives the information.
2. **Byte** — A unit of information consisting of 8 binary digits (bits).
3. **Device** — Any unit that is compatible with the IEEE 488-1978 Standard.
4. **Device Dependent** — A response to information sent on the HP-IB that is characteristic of an individual device's design and can vary from device to device.
5. **Operator** — The person who operates either the system or any device in the system.
6. **Polling** — The process typically used by a controller to locate a device that needs to interact with the controller. There are two types of polling:
   - **Serial Poll** — This method obtains one byte of operational information about an individual device in the system. The process must be repeated for each device from which information is desired.
   - **Parallel Poll** — This method obtains information about a group of devices simultaneously (one bit per device).

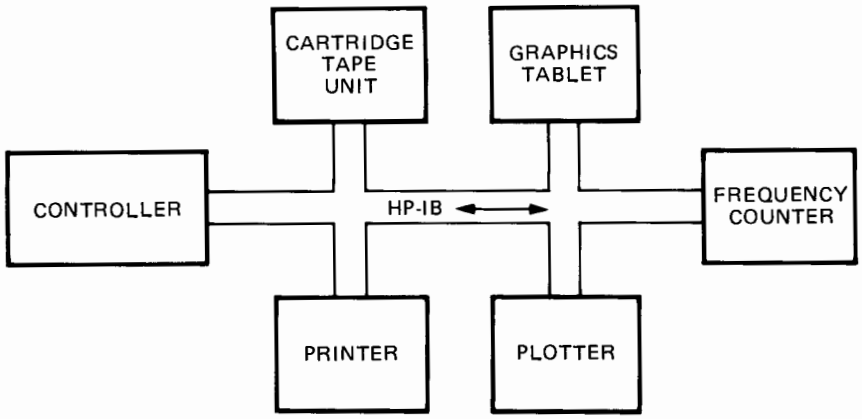
Interface Bus Concepts

Devices which communicate along the interface bus can be classified into three basic categories.

1. **Talkers** — Devices which send information on the bus when they have been addressed.
2. **Listeners** — Devices which receive information sent on the bus when they have been addressed.

3. **Controllers** — Devices that can specify the talker and listeners for an information transfer. Controllers can be categorized as one of two types:
   - **Active Controller** — The current controlling device on the bus. Only one device can be the active controller at any time.
   - **System Controller** — The only controller that can take priority control of the bus if it is not the current active controller. Although each bus system can have only one system controller, the system can have any number of devices capable of being the active controller.

HP-IB System Example



Message Concepts

Devices which communicate along the interface bus are transferring quantities of information. The transfer of information can be from one device to another device or from one device to several devices. The quantities of information can easily be thought of as "messages." Typically, each message consists of two basic parts: the address specified by the controller, and the information that comprises the message.

In turn, the messages can be classified into twelve types. The list below gives the twelve message types for the HP-IB.

1. **The Data Message** — This is the actual information which is sent from one talker to one or more listeners along the interface bus. Data can be in numeric form, or can be a string of characters (typically ASCII).
2. **The Trigger Message** — This message causes the listening device(s) to perform a device-dependent action, synchronized with the occurrence of this message.
3. **The Clear Message** — This message causes either the listening device(s) or all of the devices on the bus to return to their predefined device-dependent states.
4. **The Remote Message** — This message causes the device to switch from local front-panel control to remote program control when addressed to listen.
5. **The Local Message** — This message clears the Remote Message from the listening device(s) and returns the device(s) to local front-panel control.

6. **The Local Lockout Message** — This message prevents a device operator from manually inhibiting remote program control.

7. **The Clear Lockout and Set Local Message** — This message causes all devices on the bus to be removed from Local-Lockout control and return to Local control. This message also clears the Remote Message for all devices on the bus.

8. **The Require Service Message** — A device can send this message at any time to signify that the device needs some type of interaction with the controller. This message is cleared when the device sends its Status Byte Message or if the device no longer requires service.

9. **The Status Byte Message** — A byte that represents the status of a single device on the bus. Bit 6 indicates whether the device sent a Require Service Message, and the remaining bits indicate operational conditions defined by the device. This byte is sent from a talking device in response to a serial poll operation performed by a controller.

10. **The Status Bit Message** — A byte that represents the operational conditions of a group of devices on the bus. Each device responds on a particular bit of the byte, thus identifying a device-dependent condition. This bit is typically sent by devices in response to a parallel poll operation.

    The Status Bit Message can also be used by a controller to specify the particular bit and logic level that a device will respond with when a parallel poll operation is performed. Thus more than one device can respond on the same bit.

11. **The Pass Control Message** — This transfers the bus management responsibilities from the active controller to another controller.

12. **The Abort Message** — The system controller sends this message to unconditionally assume control of the bus from the active controller. This message terminates all bus communications (but does not implement a Clear Message).

These messages represent the full implementation of all HP-IB system capabilities. Each device in a system may be designed to use only the messages that are applicable to its purpose in the system. It is important for you to be aware of the HP-IB functions implemented on each device in your HP-IB system to ensure the operational compatibility of the system. The HP-IB functions that are implemented on the graphics tablet are discussed in the HP-IB Interfacing chapter of this manual.

The HP Interface Bus

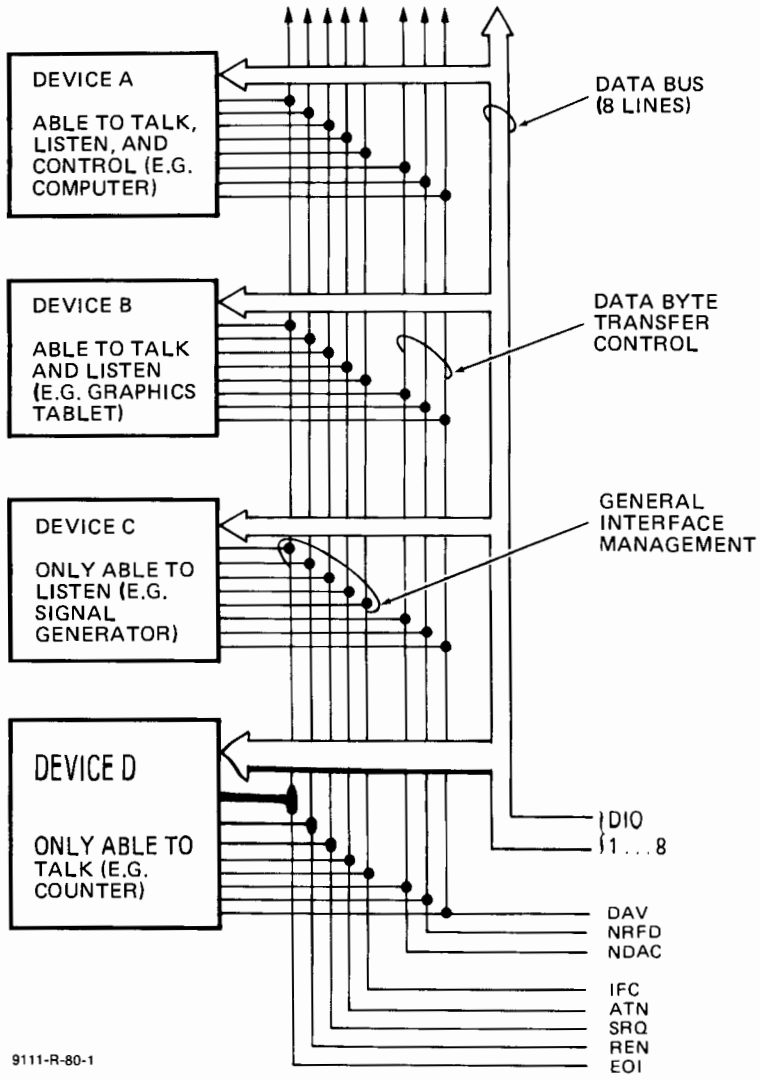
HP-IB Lines and Operations

The HP Interface Bus transfers data and commands between the components of an instrumentation system on 16 signal lines. The interface functions for each system component are performed within the component so only passive cabling is needed to connect the systems. The cables connect all instruments, controllers, and other components of the system in parallel to the signal lines.

The eight Data I/O lines (D101 through D108) are reserved for the transfer of data and other messages in a byte-serial, bit-parallel manner.

Data and message transfer is asynchronous, coordinated by the three handshake lines: Data Valid (DAV), Not Ready For Data (NRFD), and Not Data Accepted (NDAC). The other five lines are for management of bus activity. See the following figure.

HP-IB Signal Lines



The HP-IB signal lines use a low-true logic convention where a low voltage indicates a true response. This provides active true-state assertion; i.e., asserting a signal line means setting it true or low. The low state of an HP-IB signal line is the relatively less positive signal level used to assert a specific message content associated with one of two binary logic states (1 = true, 0 = false). A bit, when set, can cause the corresponding HP-IB signal line to be set true. A set bit has a value of 1; a bit that is not set has a value of 0.

Devices connected to the bus may be talkers, listeners, or controllers. The controller dictates the roll of each of the other devices by setting the ATN (attention) line true and sending talk or listen addresses on the data lines. Addresses are set into each device at the time of system configuration either by switches built into the device or by jumpers on a PC board. While the ATN line is true, all devices must listen to the data lines. When the ATN line is false, only devices that have been addressed will actively send or receive data. All others ignore the data lines.

Several listeners can be active simultaneously, but only one talker can be active at a time. Whenever a talk address is put on the data lines (while ATN is true), all other talkers will be automatically unaddressed.

Information is transmitted on the data lines under sequential control of the three handshake lines (DAV, NRFD, and NDAC). No step in the sequence can be initiated until the previous step has been completed. Information transfer can proceed as fast as devices can respond, but no faster than allowed by the slowest device presently addressed as active. This permits several devices to receive the same message byte concurrently.

The ATN line is one of the five bus management lines. When ATN is true, addresses and universal commands are transmitted on only seven of the data lines, using the ASCII code. When ATN is false, any code of eight bits or less understood by both talker and listener(s) may be used.

The IFC (interface clear) line places the interface system in a known quiescent state.

The REN (remote enable) line is used with the Remote, Local, and Clear Lockout/Set Local Messages to select either local or remote control of each device.

Any active device can set the SRQ (service request) line true via the Require Service Message. This indicates to the controller that some device on the bus wants attention, such as a counter that has just completed a time-interval measurement and wants to transmit the reading to a printer.

The EOI (end or identify) line is used by a device to indicate the end of a multiple-byte transfer sequence. When a controller sets both the ATN and EOI lines true, each device capable of a parallel poll indicates its current status on the DIO line assigned to it.

In the interest of cost-effectiveness, it is not necessary for every device to be capable of responding to all of the lines. Each can be designed to respond only to those lines that are pertinent to its function on the bus.

The operation of the interface is generally controlled by one device equipped to act as controller. The interface transmits a group of commands to direct the other instruments on the bus in carrying out their functions of talking and listening.

The controller has two ways of sending interface messages. Multiline messages, which cannot exist concurrently with other multiline messages, are sent over the eight data lines and the three handshake lines. Uniline messages are transferred over the five individual lines of the management bus.

The commands serve several different purposes:

- Addresses or talk and listen commands select the instruments that will transmit and accept data. They are all multiline messages.
- Universal commands cause every instrument equipped to do so to perform a specific interface operation. They include multiline messages and three uniline commands: interface clear (IFC), remote enable (REN), and attention (ATN).

- Addressed commands (also referred to as primary commands) are similar to universal commands, except that they affect only those devices that are addressed and are all multiline commands. An instrument responds to an addressed command, but only after an address has already told it to be talker or listener.
- Secondary commands are multiline messages that are always used in series with an address, universal command, or addressed command to form a longer version of each. Thus they extend the code space when necessary.

To address an instrument, the controller uses seven of the eight data-bus lines. This allows instruments using the ASCII seven-bit code to act as controllers. As shown in the following table, five bits are available for addresses, and a total of 31 allowable addresses are available in one byte. If all secondary commands are used to extend this into a two-byte addressing capability, 961 addresses become available (31 allowable addresses in the second byte for each of the 31 allowable in the first byte).

Command and Address Codes

Code Form								Meaning
X	0	0	A_5	A_4	A_3	A_2	A_1	Universal Commands
X	0	1	A_5	A_4	A_3	A_2	A_1	Listen Addresses
				except				
X	0	1	1	1	1	1	1	Unlisten Command
X	1	0	A_5	A_4	A_3	A_2	A_1	Talk Address
				except				
X	1	0	1	1	1	1	1	Untalk Command
X	1	1	A_5	A_4	A_3	A_2	A_1	Secondary Commands
				except				
X	1	1	1	1	1	1	1	Ignored

Code used when attention (ATN) is true (low).
X = don't care.

Interface Functions

Interface functions provide the physical capability to communicate via HP-IB. These functions are defined in the IEEE 488-1978 Standard. This standard, which is the designer's guide to the bus, defines each interface function in terms of state diagrams that express all possible interactions.

Bus capability is grouped under 10 interface functions, for example: Talker, Listener, Controller, Remote/Local. The following table lists the functions, including two special cases of Controller.

HP-IB Interface Functions

Mnemonic	Interface Function Name
SH	Source Handshake
AH	Acceptor Handshake
T	Talker (or TE = Extended Talker)*
L	Listener (or LE = Extended Listener)*
SR	Service Request
RL	Remote/Local
PP	Parallel Poll
DC	Device Clear
DT	Device Trigger
C	Any Controller
C_N	A Specific Controller (for example: C_A, C_B . . .)
C_S	The System Controller

*Extended Talkers and Listeners use a two-byte address. Otherwise, they are the same as Talker and Listener.

Bus Messages

Since interface functions are the physical agency through which bus messages are implemented, each device must implement one or more functions to enable it to send or receive a given bus message.

The following table lists the functions required to implement each bus message. Each device's operating manual lists the functions implemented by that device. The HP-IB Interfacing chapter of this manual lists the functions implemented by the graphics tablet. Some devices, such as the 98034A Interface, list the functions implemented directly on the device.

Functions Used by Each Bus Message

Bus Message	Functions Required sender function → receiver function(s) (support functions)
Data	T → L* (SH, AH)
Trigger	C → DT* (L, SH, AH)
Clear	C → DC* (L, SH, AH)
Remote	C_S → RL* (SH, AH)
Local	C → RL* (L, SH, AH)
Local Lockout	C → RL* (SH, AH)
Clear Lockout/Set Local	C_S → RL*
Require Service	SR* → C
Status Byte	T → L* (SH, AH)
Status Bit	PP* → C
Pass Control	C_A → C_B (T, SH, AH)
Abort	C_S → T, L*C

*Since more than one device can receive (or send) this message simultaneously, each device must have the function indicated by an *.

Appendix B
HP-GL Instruction Summary

This appendix summarizes the 27 HP-GL instructions used with all 9111 tablets. Each instruction is shown with its syntax, purpose, any parameters, and any output response. The instructions are explained in detail in Chapter 3, Language Reference.

The graphics tablet recognizes a line feed (LF), a semicolon (;), or asserting EOI on the last character in the instruction string as terminators for an HP-GL instruction. Note that the tablet accepts only one terminator per instruction. Refer to the Language Reference chapter for more details about terminators.

For ASCII output, which is sent in response to the HP-GL output instructions, the tablet sends CR LF with EOI set true on the line feed as terminators to the output message.

The Beep Instruction, BP

Page 3-5

BP [frequency],[duration],[amplitude] terminator

Purpose: Generates the graphics tablet's internal tone.

Parameters: frequency — integers from 0 through 255; default is 12 Hz.

duration — integers from 1 through 32 767; default is 150 ms.

amplitude — integer from 0 through 5; default is 4.

If parameters are omitted, the last specified values are used.

The Continuous-Sampling Mode Instruction, CN

Page 3-7

CN terminator

Purpose: Sets the graphics tablet's continuous-sampling mode.

The Cursor Rate Instruction, CR

Page 3-8

CR [points per second] terminator

Purpose: Specifies the rate at which the stylus position is updated.

Parameters: Integers from 1 through 60. If parameter is omitted, default value of 60 is used.

The Digitizer Clear Instruction, DC

Page 3-9

DC terminator

Purpose: Clears digitizing modes and digitized coordinates.

The Default Instruction, DF

Page 3-9

DF terminator

Purpose: Returns tablet to default conditions. Refer to table of default conditions in Appendix C.

The Digitize Point Instruction, DP

Page 3-10

DP terminator

Purpose: Prepares the tablet to recognize the next pen press as a digitized point.

The Input Mask Instruction, IM

Page 3-10

IM [E-mask value],[S-mask value],[P-mask value] terminator

Purpose: Sets masks to specify which error conditions are recognized, which conditions cause a service request, and which conditions cause a positive response to a parallel poll.

Parameters: Integers from 0 through 32767. If parameters are omitted, the default values of 7,0,0 are used.

The Initialize Instruction, IN

Page 3-14

IN terminator

Purpose: Performs the power-on self-test, sets the tablet to default conditions listed in Appendix C, and resets P1 and P2 to default values of 400,400 and 11632,8340.

The Input Points Instruction, IP

Page 3-15

IP [$P1_x$],[$P1_y$],[$P2_x$],[$P2_y$] terminator

Purpose: Sets scaling points P1 and P2.

Parameters: Integers from −999999 to 999999. If parameters are omitted, default values of 400,400 and 11632,8340 are used.

The Output Actual Stylus Position Instruction, OA

Page 3-16

OA terminator

Purpose: Same as the output cursor instruction, OC.

The Output Cursor Instruction, OC

Page 3-16

OC terminator

Purpose: Used to output the last known stylus location; the pen status; a selected softkey number, if any; the status word value; and an error number, if any.

Response: X,Y,P,M,S,E CR LF — ASCII integers.

X — From −900 to 12900 in digitizing units.

Y — From −900 to 10300 in digitizing units.

P — 0, pen not pressed; 1, pen pressed.

M — From 0 through 16.

S — Sum of set bits in 11-bit status word.

E — From 0 through 110.

The Output Digitized Point Instruction, OD

Page 3-18

OD terminator

Purpose: Used to output the location and pen status of the last digitized point.

Response: X,Y,P CR LF — ASCII integers.

X — From −120 to 12152 in digitizing units.

Y — From −120 to 9115 in digitizing units.

P — −1, digitizing mode not set; 0, last digitized point in continuous-sample mode; 1, all digitized points except last point in continuous-sampling mode.

The Output Error Instruction, OE

Page 3-19

OE terminator

Purpose: Used to output the last recognized error.

Response: error number CR LF — ASCII integers from 0 through 110.

The Output Factor Instruction, OF

Page 3-22

OF terminator

Purpose: Used to output the number of digitizing units per millimetre in both the X- and Y-axes.

Response: 40,40 CR LF — ASCII integers.

The Output Identification Instruction, OI

Page 3-22

OI terminator

Purpose: Used to output the tablet's identification.

Response: 9111T CR LF — ASCII character string.

The Output Key Instruction, OK

Page 3-22

OK terminator

Purpose: Used to output a selected softkey value, if any.

Response: softkey value CR LF — ASCII integers from 1 through 32 768.

The Output Points Instruction, OP

Page 3-23

OP terminator

Purpose: Used to output the current coordinates of the scaling points, P1 and P2, in digitizing units.

Response: $P1_x$, $P1_y$, $P2_x$, $P2_y$ CR LF — ASCII integers from −999 999 through 999 999. Default values of P1 and P2 are 400,400 and 11 632,8340.

The Output Resolution Instruction, OR

Page 3-24

OR terminator

Purpose: Used to output the tablet's apparent resolution in millimetres.

Response: 0.025,0.025 CR LF — ASCII decimal.

The Output Status Instruction, OS

Page 3-24

OS terminator

Purpose: Used to output the sum of the bit values that are set in the 11-bit status word.

Response: status CR LF — ASCII integers. Power-on status is 24.

The Read Cursor Instruction, RC

Page 3-26

RC terminator

Purpose: Same as the output cursor instruction, OC.

The Read Softkey Instruction, RS

Page 3-26

RS [menu enable] terminator

Purpose: Used to output the number of the selected softkey, then clear softkey value.

Parameters: menu enable — 0, softkeys disabled; 1, softkeys enabled. If parameter is omitted, softkeys are enabled.

Response: softkey value CR LF — ASCII integers from 0 through 16.

The Switch Follow Instruction, SF

Page 3-27

SF terminator

Purpose: Specifies that the tablet is to take points at regular time intervals when the stylus remains pressed in the continuous-sampling mode.

The Single-Sample Mode Instruction, SG

Page 3-28

SG terminator

Purpose: Sets the graphics tablet's single-sample mode.

The Set Key Instruction, SK

Page 3-28

SK terminator

Purpose: Clears the tablet of any selected softkey values.

The Switch Normal Instruction, SN

Page 3-29

SN terminator

Purpose: Sets the tablet to begin taking points at regular time intervals when the stylus is pressed, and then stop when the stylus is pressed again in the continuous-sampling mode.

The Test Digitizer Instruction, TD

Page 3-29

TD terminator

Purpose: Initiates the user-interaction self-test.

The Take Point Instruction, TP

Page 3-29

TP terminator

Purpose: Simulates the press of the digitize switch in the stylus. A point is digitized at the stylus location whether or not the stylus is actually pressed.

Appendix C
Reference Material

Binary Coding and Conversions

Binary is a base two number system using only ones and zeros. By giving the ones and zeros positional value, any decimal number can be represented. For example, this diagram shows how decimal 41 = binary 101001:

Decimal

$$4 \times 10^1 + 1 \times 10^0$$
$$4 \times 10 + 1 \times 1$$
$$4 \qquad 1_{10}$$

Binary

$$1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$
$$1 \times 32 + 0 \times 16 + 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1$$
$$1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1_2$$

Binary-Decimal Conversions

To convert from binary to decimal, the positional values of the ones are added up. From the above example, this would be:

$$2^5 + 2^3 + 2^0 = 32 + 8 + 1 = 41$$

To convert from decimal to binary, the decimal number is divided by 2. The remainder is the binary equivalent. For example:

	Remainder (read up)
$2 \overline{)41} \rightarrow$	1
$2 \overline{)20} \rightarrow$	0
$2 \overline{)10} \rightarrow$	0
$2 \overline{)5} \rightarrow$	1
$2 \overline{)2} \rightarrow$	0
$2 \overline{)1} \rightarrow$	1

= Binary 101001

Two's Complement Representation

Two's complement representation is a method of storing either positive or negative numbers in a word. It works as shown below:

Positive numbers: If bit 15 is 0, then the word is a positive number represented in normal binary form.

Negative numbers: If bit 15 is 1, then the word is a negative number represented in two's complement form. To find the absolute value of a negative number, invert all the bits and add 1.

Problem: What is the value of 11111111 11110000?

Bit 15 Bit 0

Solution: Bit 15 tells that this is a negative number.

Inverting all the bits gives: 00000000 00001111

Adding 1 results in: 00000000 00010000

So the value of the given bit pattern is: −16

Binary Default Response Format

The binary default response consists of three sixteen-bit words sent as six eight-bit bytes with EOI true on the sixth byte. Each word is a two's complement number with the most significant byte sent first. The first word represents the stylus' X-position; the second word represents the stylus' Y-position; and the third word represents the tablet's current status, which is the sum of the values of the set bits in the eleven-bit status word.

ASCII Character Codes

ASCII stands for "American Standard Code for Information Interchange." It is a commonly used code for representing letters, numerals, punctuation, and special characters. ASCII is an eight-bit binary code, containing seven data bits and one parity bit.

A complete list of the characters in the ASCII set, their binary code values, their octal and decimal representations, and any corresponding HP-IB commands is included here.

ASCII Char.	EQUIVALENT FORMS				HP-IB
	Binary	Oct	Hex	Dec	
NULL	00000000	000	00	0	
SOH	00000001	001	01	1	GTL
STX	00000010	002	02	2	
ETX	00000011	003	03	3	
EOT	00000100	004	04	4	SDC
ENQ	00000101	005	05	5	PPC
ACK	00000110	006	06	6	
BELL	00000111	007	07	7	
BS	00001000	010	08	8	GET
HT	00001001	011	09	9	TCT
LF	00001010	012	0A	10	
VT	00001011	013	0B	11	
FF	00001100	014	0C	12	
CR	00001101	015	0D	13	
SO	00001110	016	0E	14	
SI	00001111	017	0F	15	
DLE	00010000	020	10	16	
DC1	00010001	021	11	17	LLO
DC2	00010010	022	12	18	
DC3	00010011	023	13	19	
DC4	00010100	024	14	20	DCL
NAK	00010101	025	15	21	PPU
SYNC	00010110	026	16	22	
ETB	00010111	027	17	23	
CAN	00011000	030	18	24	SPE
EM	00011001	031	19	25	SPD
SUB	00011010	032	1A	26	
ESC	00011011	033	1B	27	
FS	00011100	034	1C	28	
GS	00011101	035	1D	29	
RS	00011110	036	1E	30	
US	00011111	037	1F	31	

ASCII Char.	EQUIVALENT FORMS				HP-IB
	Binary	Oct	Hex	Dec	
space	00100000	040	20	32	LA0
!	00100001	041	21	33	LA1
"	00100010	042	22	34	LA2
#	00100011	043	23	35	LA3
$	00100100	044	24	36	LA4
%	00100101	045	25	37	LA5
&	00100110	046	26	38	LA6
'	00100111	047	27	39	LA7
(	00101000	050	28	40	LA8
)	00101001	051	29	41	LA9
*	00101010	052	2A	42	LA10
+	00101011	053	2B	43	LA11
,	00101100	054	2C	44	LA12
-	00101101	055	2D	45	LA13
.	00101110	056	2E	46	LA14
/	00101111	057	2F	47	LA15
0	00110000	060	30	48	LA16
1	00110001	061	31	49	LA17
2	00110010	062	32	50	LA18
3	00110011	063	33	51	LA19
4	00110100	064	34	52	LA20
5	00110101	065	35	53	LA21
6	00110110	066	36	54	LA22
7	00110111	067	37	55	LA23
8	00111000	070	38	56	LA24
9	00111001	071	39	57	LA25
:	00111010	072	3A	58	LA26
;	00111011	073	3B	59	LA27
<	00111100	074	3C	60	LA28
=	00111101	075	3D	61	LA29
>	00111110	076	3E	62	LA30
?	00111111	077	3F	63	UNL

ASCII Char.	EQUIVALENT FORMS				HP-IB
	Binary	Oct	Hex	Dec	
@	01000000	100	40	64	TA0
A	01000001	101	41	65	TA1
B	01000010	102	42	66	TA2
C	01000011	103	43	67	TA3
D	01000100	104	44	68	TA4
E	01000101	105	45	69	TA5
F	01000110	106	46	70	TA6
G	01000111	107	47	71	TA7
H	01001000	110	48	72	TA8
I	01001001	111	49	73	TA9
J	01001010	112	4A	74	TA10
K	01001011	113	4B	75	TA11
L	01001100	114	4C	76	TA12
M	01001101	115	4D	77	TA13
N	01001110	116	4E	78	TA14
O	01001111	117	4F	79	TA15
P	01010000	120	50	80	TA16
Q	01010001	121	51	81	TA17
R	01010010	122	52	82	TA18
S	01010011	123	53	83	TA19
T	01010100	124	54	84	TA20
U	01010101	125	55	85	TA21
V	01010110	126	56	86	TA22
W	01010111	127	57	87	TA23
X	01011000	130	58	88	TA24
Y	01011001	131	59	89	TA25
Z	01011010	132	5A	90	TA26
[	01011011	133	5B	91	TA27
\	01011100	134	5C	92	TA28
]	01011101	135	5D	93	TA29
^	01011110	136	5E	94	TA30
_	01011111	137	5F	95	UNT

ASCII Char.	EQUIVALENT FORMS				HP-IB
	Binary	Oct	Hex	Dec	
`	01100000	140	60	96	SC0
a	01100001	141	61	97	SC1
b	01100010	142	62	98	SC2
c	01100011	143	63	99	SC3
d	01100100	144	64	100	SC4
e	01100101	145	65	101	SC5
f	01100110	146	66	102	SC6
g	01100111	147	67	103	SC7
h	01101000	150	68	104	SC8
i	01101001	151	69	105	SC9
j	01101010	152	6A	106	SC10
k	01101011	153	6B	107	SC11
l	01101100	154	6C	108	SC12
m	01101101	155	6D	109	SC13
n	01101110	156	6E	110	SC14
o	01101111	157	6F	111	SC15
p	01110000	160	70	112	SC16
q	01110001	161	71	113	SC17
r	01110010	162	72	114	SC18
s	01110011	163	73	115	SC19
t	01110100	164	74	116	SC20
u	01110101	165	75	117	SC21
v	01110110	166	76	118	SC22
w	01110111	167	77	119	SC23
x	01111000	170	78	120	SC24
y	01111001	171	79	121	SC25
z	01111010	172	7A	122	SC26
{	01111011	173	7B	123	SC27
|	01111100	174	7C	124	SC28
}	01111101	175	7D	125	SC29
~	01111110	176	7E	126	SC30
DEL	01111111	177	7F	127	SC31

Default Conditions

Function	Condition
Data rate	60 points per second
Input masks	
E-mask	7
S-mask	0
P-mask	0
Status word	24 (power-on and IN instruction) 16 (DF instruction)
Softkey area	On
Softkey selection	0
Sampling mode (CN or SG)	None
Switch mode (SN or SF)	SN (switch normal)
Beeper	
Frequency	12 Hz
Duration	150 ms
Amplitude	4
*P1	400, 400 DUs
*P2	11 632, 8340 DUs

*P1 and P2 are set to default values with the initialize instruction, IN. They are not affected by the default instruction, DF.

Error Messages

Normal Operation Errors

Error Number	Cause
0	No error.
1	Invalid HP-GL instruction.
2	Invalid number of parameters in HP-GL instruction.
3	Invalid parameter in HP-GL instruction.
7	Inconsistent stylus location data.
100	Run-time system error.

User-Interaction Self-Test Errors

Error Number	Cause
50	Illegal proximity signal at time of self-test.
51	Illegal pen press at time of self-test.
52	Pen press detected before proximity to platen detected.
53	Position error indicated.

Hardware Errors

Error Number	Cause	Indicator Light Status		
		DIGITIZE	MENU	ERROR
101	Processor Register or Flag Error	Off	Off	On
102	ROM Checksum Error	Off	On	Off
103	RAM Test Error	Off	On	On
104	I/O Port Error	On	Off	Off
105	Interface Chip Error	On	Off	On
106	Phase Counter Error	On	On	Off
107	Interrupt Mask	On	On	On
108	153 Hz Clock Interrupt	On	On	On
109	RAM Timer Interrupt	On	On	On
110	HP-IB Chip Interrupt	On	On	On

No Operation Instructions

In order to maintain software compatibility with other HP graphics devices, such as plotters and digitizers, the graphics tablet accepts the instructions listed below without error and without taking any action.

The no operation (NOP) instructions on the 9111 graphics tablet are:

AN (Air Normal Mode)

AT (Air Toggle Mode)

AV (Activate Vacuum)

CC (Compatibility Command, Character Chord Angle)

DD (Display Driver)

DR (Relative Direction)

IW (Input Window)

NOTE: OW (Output Window) is not accepted.

LB (Label)

LT (Line Type)

PA (Plot Absolute)

PC (Position Cursor)

PD (Pen Down)

PG (Advance Page)

PU (Pen Up)

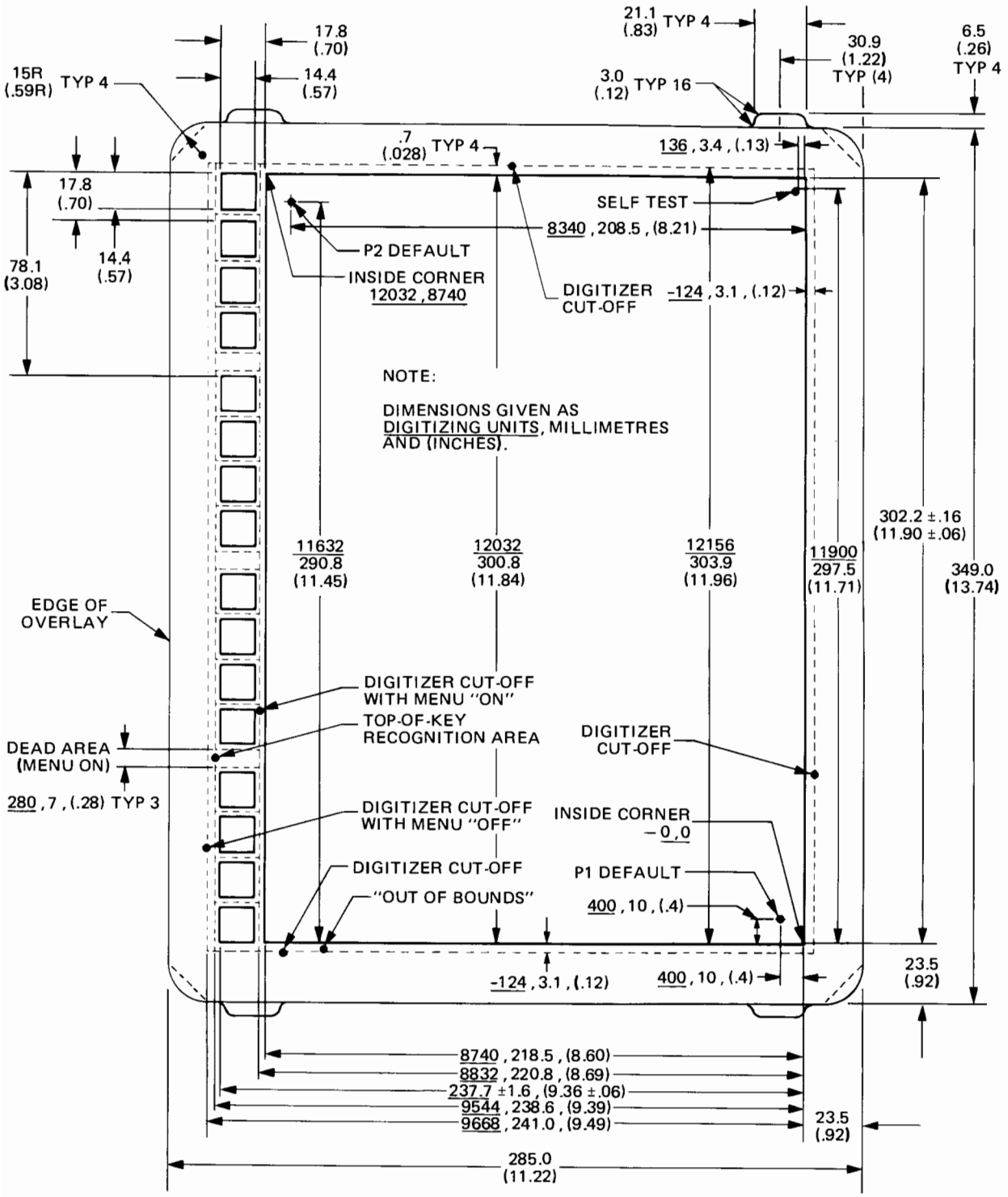
RV (Remove Vacuum)

SL (Character Slant)

SP (Pen Select)

SR (Relative Character Size)

Platen Digitizing Units and Dimensions



Appendix D
Glossary

Addressing — The characters sent by a controlling device specifying which device sends information on the bus and which device(s) receives the information.

ASCII — American Standard Code for Information Interchange. A standard code capable of representing letters, numbers, punctuation marks, and control codes in a form acceptable to machines. It consists of seven information bits and one parity bit for error checking.

Binary — A number system with a base of 2, using the numerals 0 and 1.

Binary default response — The format in which the tablet sends data unless otherwise requested. The binary default response consists of the stylus location and status word sent as six eight-bit bytes with EOI asserted on the sixth byte.

Bit — The smallest part of a byte which contains intelligible information. A binary digit.

Bus — A signal line or a set of signal lines used by an interface system to which a number of devices are connected and over which messages are carried.

Byte — A character sent over data lines, normally consisting of eight bits.

CRT — Cathode Ray Tube. Provides the display on a video terminal.

Carriage return (CR) — A special character which causes movement back to the beginning of an existing line.

Cartesian coordinate — Either of two coordinates that locate a point on a plane and measure its distance from either of two intersecting straight-line axes along a line parallel to the other axis.

Continuous digitizing — A digitizing mode in which X,Y coordinates are automatically picked up at regular time intervals as you move the stylus on the tablet's surface.

Cursor — An indicator which guides data entry or locates existing data on the display.

Cursor tracking — The use of the cursor on the display to track the movement of the stylus on the tablet's surface.

Data rate — The rate at which the tablet checks the stylus position in points per second.

Default — A value that is automatically assumed if no other value is specified.

Digitizing — The process of pressing the stylus on the tablet's surface and the tablet sending the X,Y coordinates of that location to the computer.

Digitizing limits — The physical active digitizing area on the tablet's surface. Nothing can be digitized outside these limits.

Digitizing unit (DU) A digitizing unit is 0.025 mm (approximately 0.001 in.) in length and is the unit of measure used to express physical locations in the tablet's Cartesian coordinate system.

EOI (End or Identify) A signal line on the HP-IB that is used by a device to indicate the end of a data message. When used with the ATN line, each device capable of a parallel poll indicates its current status.

Error mask (E-mask) Specifies the error conditions that the tablet is to recognize.

Factor A quantity by which a given quantity is multiplied or divided in order to indicate a difference in measurement.

Graphics tablet A data entry device that converts a physical location on its surface into digital values usable by the computer.

Hardware The electric, electronic, and mechanical equipment used for processing data.

HP-GL Hewlett-Packard Graphics Language. The instruction set sent to and understood by the graphics tablet.

HP-IB Hewlett-Packard Interface Bus. Hewlett-Packard's implementation of the IEEE 488-1978 Instrumentation Bus used to connect multiple devices together with a well-defined hardware protocol.

Initialize To set certain conditions to the starting values.

Input A process of transferring information into a device.

Input mask A programming tool allowing the user to select certain conditions by "masking" those conditions not wanted.

Interactive graphics A two-way communication between the user and the computer to produce graphics. The graphics tablet facilitates this process.

Interface status check The process of checking a bit in an interface register to see if a service request has been sent.

Interrupt A signal that suspends ordinary operation of a computer so that some immediate need can be met.

Line feed (LF) A character that causes the printing or display position to be moved to the next printing or display line. It can be used a terminator for the tablet's HP-GL instructions.

Listener A device that receives information sent on the bus when it is addressed.

Menu A listing of available choices.

Mnemonic A code that assists your memory. For example, CR suggests cursor rate. Each HP-GL instruction begins with a two-letter mnemonic.

Offset factor The factor used to compensate for starting at an origin point other than 0,0 when converting digitizing units to user units.

Output A process of transferring information out of a device.

P1 and P2 Two points on diagonally opposite corners that define a rectangular digitizing area.

Parallel poll The method of obtaining information about a group of devices simultaneously (one bit per device).

Parallel poll mask (P-mask) Specifies which conditions cause a positive response to a parallel poll on the HP-IB.

Parameter A variable that is given a fixed value for a specific use.

Pen status A code from the tablet that tells whether the stylus is pressed, whether a digitized point is the last in a series while in the continuous digitizing mode, or whether a digitizing mode is set.

Pick Pressing the stylus in a certain location on the tablet to select a menu item such as a symbol or command.

Platen The white ceramic surface of the graphics tablet.

Polling The process typically used by a computer to locate a device that needs to interact with the computer.

Program An organized set of instructions that tells the computer and tablet to perform certain tasks.

Protocol A set of conventions for transference of information between devices.

Proximity Within approximately ¼ inch of the tablet's surface.

Register A place reserved in memory for holding a piece of information to be processed or transferred.

Resolution The smallest distance the tablet can distinguish between two digitized points.

ROM Read-Only Memory. A memory device in which the memory locations are set to fixed patterns when the device is manufactured. Used for invariant programs and data.

Rubber-band line A line on the screen with a fixed point at one end while the other end moves with the tablet's stylus.

Run A single, complete execution of a computer program.

Scaling Establishing a proportion between two sets of dimensions; for example, converting digitizing units into user units.

Separator Punctuation separating numerical parameters in an HP-GL instruction, usually a comma or semicolon.

Serial poll The method of obtaining one byte of operational information about an individual device in the system.

Service request The message a device sends to signify that it needs interaction with the computer.

Single-point digitizing A digitizing mode in which one point is digitized at a time.

Softkeys The sixteen boxes at the top of the tablet's platen which the user can personalize for specific job functions and which, when digitized, send an integer corresponding to the box selected.

Software The various programs and programming aids that are used in the operation of data processing equipment.

Status byte The lower eight bits (0 to 7) of the current status word, which the tablet sends in response to a serial poll.

Status mask (S-mask) Specifies when HP-IB service requests are to be sent.

Status word — Eleven bits of information that communicate significant conditions within the tablet.

String — A linear sequence of alphanumeric characters.

Stylus — The digitizing or "pointing" device on the tablet that looks much like a ball-point pen.

Switch-follow mode — The continous-digitizing mode in which the tablet takes points at regular time intervals when the stylus remains pressed.

Switch-normal mode — The continuous-digitizing mode in which the tablet begins taking points at regular time intervals when the stylus is pressed, and then stops when the stylus is pressed again.

Syntax — The rules governing the structure of a language.

Talk-only mode — The mode in which the tablet does not use the HP-IB addressing protocol but, instead, responds to all initiated data transfers by sending the binary default response.

Talker — A device that sends information on the bus when it is addressed.

Terminator — A special character or signal that is required to end an HP-GL instruction. The terminators that are valid on the graphics tablet are the semicolon (;), a line feed (LF), or asserting EOI.

User unit — The unit of measure of the digitizing area defined to suit the user's application.

Variable — The name of something that can take one value or a succession of values in a program.

X_{min},X_{max} — The minimum and maximum coordinate points on the X-axis.

Y_{min},Y_{max} — The minimum and maximum coordinate points on the Y-axis.

Subject Index

Subject Index (Continued)

f

g

h

i

l

n

Subject Index (Continued)

o

p

r

s

Subject Index (Continued)

S (Continued)

t

u

v

PART NO. 09111-90004
MICROFICHE NO. 09111-90054

PRINTED IN U.S.A.