HEWLETT PACKARD

HP 7090A
Interfacing & Programming Manual

Interfacing and Programming Manual
Measurement Plotting System
HP 7090A

Using HP-RL and HP-GL Instructions

HEWLETT PACKARD

HP Part Number: 07090-90003
Microfiche Part Number: 07090-90203

Printed in U.S.A.
Print Date: January 1992

8600 Soper Hill Road, Everett, WA 98205-1298 U.S.A.

Herstellerbescheinigung

Hiermit wird bescheinigt, daß das Gerät/System

HP 7090A MEASUREMENT PLOTTING SYSTEM

in Übereinstimmung mit den Bestimmungen von Postverfügung 1046/84 funkentstört ist.

Der Deutschen Bundespost wurde das Inverkehrbringen dieses Gerätes/Systems angezeigt und die Berechtigung zur Überprüfung der Serie auf Einhaltung der Bestimmungen eingeräumt.

Zusatzinformation für Meß- und Testgeräte

Werden Meß- und Testgeräte mit ungeschirmten Kabeln und/oder in offenen Meßaufbauten verwendet, so ist vom Betreiber sicherzustellen, daß die Funk-Entstörbestimmungen unter Betriebsbedingungen an seiner Grundstücksgrenze eingehalten werden.

Manufacturer's declaration

This is to certify that the equipment

HP 7090A MEASUREMENT PLOTTING SYSTEM

is in accordance with the Radio Interference Requirements of Directive FTZ 1046/1984. The German Bundespost was notified that this equipment was put into circulation, the right to check the series for compliance with the requirements was granted.

Additional Information for Test- and Measurement Equipment

If Test- and Measurement is operated with unscreened cables and/or used for measurements on open set-ups, the user has to assure that under operating conditions the Radio Interference Limits are still at the border of his premises.

Table of Contents

Table of Contents
(Continued)

Chapter 2

Recorder Fundamental Concepts (Continued)

Chapter 3

Chapter 4

Table of Contents
(Continued)

Table of Contents
(Continued)

Table of Contents (Continued)

How to Use HP 7090A Documentation

The HP 7090 can be used as a stand-alone instrument, or as a programmable input/output-component of a computer-based system. Therefore, two separate manuals are provided — an Operator's Manual (Part No. 07090-90002), which details front-panel (stand-alone) operations, and an Interfacing and Programming Manual (Part No. 07090-90001), which details programming operations. A Pocket Guide (Part No. 07090-90004) is also included for use as a quick programming reference.

Using the documentation systematically will help you to achieve optimum results in the shortest possible time, and save hours of frustrating experimentation. Contents of the manuals are summarized below, followed by suggestions on using the documentation based on your application.

The Operator's Manual

The operator's manual contains all of the information necessary to perform measurement and plotting operations using front-panel controls. Chapters 1 and 2 contain important information on instrument installation and maintenance, how to load pens and paper, and how to use the controls. The remaining chapters detail how to set up and perform measurement and plotting operations.

The Interfacing and Programming Manual

The interfacing and programming manual contains all of the information you will need to perform remote-controlled operation of the HP 7090. A complete description of all programming instructions usable for recorder and plotter functions is included, as well as directions on using the HP-IB interface bus to interface the HP 7090 with specific computers.

The Pocket Guide

The pocket guide is intended for those who are already familiar with the information contained in the interfacing and programming manual, and who need a convenient reference while programming the HP 7090. Each programming instruction, along with its parameters and function, is listed in alphabetical order.

For All HP 7090 Users

If you have just received your HP 7090, or are just learning to use it, read the following portions of the operator's manual before attempting any front-panel or programming operations:

Chapter 1. This chapter will help you setup the HP 7090. It contains important information on the accessories provided, instrument installation, maintenance, and power requirements.

Chapter 2. This chapter will acquaint you with the HP 7090's major features, controls, and operating modes. In addition, it explains how to load pens and paper in preparation for measurement and plotter applications.

Chapter 3. The section in this chapter entitled "Connecting Analog Input Signals" contains information on connecting analog input signals intended for measurement, as well as instructions on use of the guard switch. To prevent damage to the HP 7090, be sure to read this section before performing any measurement operation.

After reading the introductory material listed above, refer to the manual that is suitable for your application. If you intend to use the HP 7090 as a stand-alone instrument, read the remaining portions of the operator's manual. On the other hand, if you intend to interface the HP 7090 with a host computer and perform programming operations, read the interfacing and programming manual, using the operator's manual as an additional source of information on specific topics. After you are comfortable with using the HP 7090 programming instructions, the pocket guide can be used as a handy reference.

Understanding Manual Conventions

Before reading this manual, you should understand the meaning of type styles, symbols, and number representation used in text. Words typed in **SMALL BOLDFACE TYPE** are either buttons, switches, or words actually found on the HP 7090 or computer. Headings in large, colored type identify specific parts of the discussion of an instruction. **BOLD CONDENSED TYPE** denotes a single ASCII character which should be sent to the HP 7090. Numbers are typed using SI (International System of Units) standards: i.e., numbers with more than four digits are placed in groups of three, separated by a space instead of commas. The groups are counted both to the left and right of the decimal point (54 321.123 45).

Chapter 1
Introduction to Programming

What You'll Learn in This Chapter

This chapter briefly describes the HP 7090's measurement and plotting capabilities and its programming instruction sets and their syntax. Additional information describes the computers and programming languages that can be used, the connection and addressing requirements, and how to communicate with the HP 7090 using several Hewlett-Packard computers.

Terms You Should Understand

HP-IB (Hewlett-Packard Interface Bus — HP's implementation of the IEEE standard 488-1978 digital interface for programmable instrumentation.

HP-RL (Hewlett-Packard Recorder Language) — the I/O instructions understood by the HP 7090 that control the recorder and data acquisition functions.

HP-GL (Hewlett-Packard Graphics Language) — the I/O instructions understood by the HP 7090 that control the plotter functions.

A Brief Look at the HP 7090

The HP 7090 is a three-channel analog-to-digital data acquisition system and a digital plotter packaged into a single instrument. The HP 7090 can operate as a conventional X-Y recorder, a waveform recorder, a data acquisition system component, or a six-pen vector plotter. Recordings and plots can be produced on four sizes of paper or transparency film.

ISO A4 (210 × 297 mm)	ANSI A (8½ × 11 in.)
ISO A3 (297 × 420 mm)	ANSI B (11 × 17 in.)

The three floating and guarded analog input channels will accept a maximum input voltage of 200 V, dc or peak, and provide X-Y or Y-T direct recording for signals up to approximately 10 Hz and X-Y-Y or Y-Y-Y-T buffered recording for signals up to approximately 3 kHz. The range sensitivity of each channel may be set independently from 5 mV to 100 V full scale.

The direct recording mode is functionally equivalent to an X-Y recorder and is used to produce real-time recordings of low-frequency analog signals (amplitude dependent to 10 Hz). The sampling rate for all direct recordings is 250 samples/s. Slewing speed in this mode is 125 cm/s; acceleration is approximately 1960 cm/s^2. Any one of the three input channels may be plotted against time. Sweep time for Y-T (versus-time) measurements is determined by the total time setting, and times of 1.0 second to 24.0 hours may be used. In addition to versus-time measurements, X-Y (versus-chan3) measurements can be made in which channel 1 or channel 2 is plotted against channel 3. The duration of versus-chan3 measurements is determined by the specified total time setting, unless manual triggering is used and the measurement is initiated from the front panel. In this case, versus-chan3 measurements are "free-running."

The buffered recording mode is functionally equivalent to a waveform recorder and can be used to perform both versus-time and versus-chan3 measurements. In this mode, 1000 samples from each channel are digitized and stored in separate buffers at a rate determined by the total time setting. Each buffer is simultaneously filled in the specified time and measurement durations from 0.03 seconds to 24.0 hours are allowed. The stored sample values can subsequently be plotted or sent to a computer.

The HP 7090 includes manual, external, and internal channel 1 trigger capabilities. The internal trigger mode allows ± slope triggering at a set voltage level, triggering when the signal level is outside the voltage window, or when the signal level is inside the voltage window. Storing a digitized signal beginning before or some time after the trigger point is also possible using the pre-trigger and post-trigger capabilities. Pre-trigger recording allows information to be recorded before the trigger point, which aids the user in determining the events which led up to the trigger. Post-trigger recording inserts a precise delay between the trigger point and the beginning of recording. This feature is important if the portion of the incoming waveform containing useful information occurs a specific length of time after trigger conditions are met.

The data acquisition mode permits the transfer of digitized data to a computer. When taken directly from the A/D converters, the data sample sets can be continuously streamed or a finite number of data sample sets between 1 and 32767 can be sent. Data sample sets between 1 and 1000 can also be sent from the buffers. Data is transferred directly at a maximum rate of 500 sample sets per second, but buffered data sets can be sent as fast as the computer can receive them. In each case, the sample sets can be sent in either ASCII or binary formats.

The plotter mode allows the HP 7090 to be used as a conventional digital plotter to produce multicolor graphics. Graphs can be annotated using any of five character sets, including three European sets. Text can be written in any direction, with or without character slant, and in varying sizes. Trace identification capabilities include symbol mode plotting and seven programmable dashed-line fonts.

The HP 7090 powers up in the plotter mode and specific instructions must be executed to enter the recorder or data acquisition modes.

Stand-Alone Versus Programmed Operation

The HP 7090 can be programmed to do everything you can do from the front panel, except initiate a "free-running" versus-chan3 measurement and access the **DATA DISPLAY** and **SAVE/RESTORE SETUP** features. In addition, buffered or real-time data can be output to a computer in the data acquisition mode. This data, or some modified form of it, can subsequently be sent to the HP 7090 for plotting and annotation. The front panel can also be deactivated to prevent inadvertent interruption of long time duration measurements.

The data acquisition mode and most plotter functions are the main features that cannot be accessed from the front panel. This means that you don't have to learn to program to perform direct and buffered recording measurements. For full details of how to use the front panel, refer to the HP 7090 Operator's Manual.

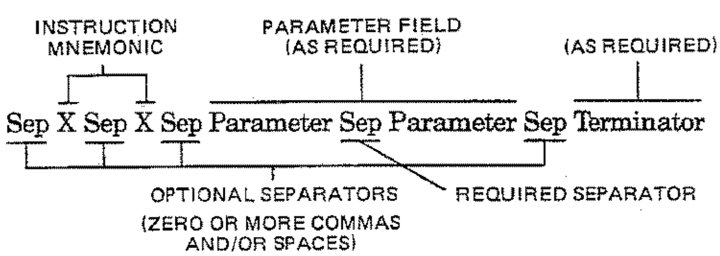
The HP 7090 Instruction Set

Two types of instructions are used to program the HP 7090: HP-RL and HP-GL. HP-RL instructions are used to control the recording and data acquisition functions. HP-GL instructions are used to control the plotter functions. Both types of instructions consist of a two-letter code, its parameter field, if any, and a terminator. The parameters are interpreted in accordance with the functions defined by the two-letter code. In most cases, the code letters are mnemonics that should help you remember the instruction. For instance, SP is the code for selecting a pen. This instruction uses only one parameter to define the number (1 through 6) of the pen to be selected (i.e., SP 4 selects pen number 4).

HP-RL and HP-GL Syntax

The two-letter mnemonic of each instruction may be upper- or lowercase. If parameters follow the mnemonic, they must be separated from each other by at least one comma or space. Optional commas and/or spaces may be used as separators before, after, and between the mnemonic and before the terminator. All instructions will execute immediately after the mnemonic or last possible parameter is received. However, an instruction is normally terminated by a semicolon, line feed, or the first letter of the next mnemonic. The label instruction, LB, is a special case; it must be terminated with the label terminator character. This character defaults to the ASCII end-of-text character, ETX (decimal equivalent 3). The label terminator may be changed from its default value using the DT instruction.

The following diagram illustrates the general instruction syntax and where optional separators may be inserted.



Parameter Formats

The parameter fields must be specified in the range and format defined by each respective instruction. The format can be of three types:

1. Integer Format — any number subset between −32 768 and +32 767. Decimal fractions, if included, are truncated. If no sign is specified, the parameter is assumed to be positive.

2. Real Format — any number subset between −32 768.000 0 and +32 767.999 9. The decimal point may be omitted when no decimal fraction is specified. If no sign is specified, the parameter is assumed to be positive.

3. Label Fields — any combination of ASCII characters. Refer to the LB instruction, Chapter 8, for the complete description.

NOTE: All ASCII control characters (decimal equivalents 0 through 31 and 127) are ignored, except as label characters. ■

Notations Used for Expressing Syntax

The syntax shown under the description for each instruction uses the following notations.

MNemonic — For readability, the mnemonic is shown uppercase and separated from the parameters and/or terminator.

( ) — All parameters in parentheses are optional.

(, . . .) — Any number of optional X,Y coordinate pairs.

(c . . . c) — Any number of labeling characters.

Terminator — A semicolon, line feed, or the next mnemonic are valid terminators.

Syntax Compatibility

The syntax implemented on the HP 7090 is extremely flexible and differs from that used on other plotters such as the HP 9872. Therefore, any software written for the HP 7090 plotter function which takes advantage of its less rigorous syntax will not be able to drive earlier devices. If software is to be written to drive other HP-GL plotters, the more rigorous syntax of the HP 9872 should be used. The following diagram illustrates the general syntax of the HP 9872 plotter.

XX Parameter (,Parameter) Terminator

MNEMONIC (XX) — OPTIONAL PARAMETER ((,Parameter))

The HP 9872 syntax totally ignores spaces. It does not allow commas between the characters of the mnemonic, only one comma must separate parameters, and only a semicolon or line feed may be used as the terminator. In addition, parameters requiring integer format may not contain a decimal point or decimal fraction.

HP-RL and HP-GL Instruction Lists

All HP-RL and HP-GL instructions are listed alphabetically in the following tables. The complete explanation for each instruction is contained in the indicated chapter. Refer to Appendix C for a list of other instructions which are recognized, but cause no operation. These instructions are included for software compatibility with other HP plotters.

Mnemonic	Function	Chapter
DG	Draw grid	3
DO	Set data output	4
GL	Set grid divisions	3
IR	Set full scale range	3
IT	Set real-time clock	3
IZ	Set zero and full scale	3
MS	Measurement start	4
MT	Measurement terminate	4
PL	Plot buffer	4
QA	Query trigger level and width	3
QB	Query total time	3
QC	Query channel status	12
QD	Query direct A/D data samples	4
QG	Query post- and pre-trigger time	3
QI	Query buffered A/D data samples	4
QL	Query grid divisions	3
QM	Query recording mode	3
QR	Query full scale range	3
QS	Query recorder status	12
QT	Query trigger mode	3
QU	Query trigger time	3
QV	Query dc offset	3
QW	Query real-time clock	3
QZ	Query zero and full scale points	3
RE	Set recording mode	3
RL	Remote/local	11
SD	Set sample delay	4
SV	Set dc offset	3
TA	Set trigger level and width	3
TB	Set total time	3
TD	Label time and date	3
TG	Set post- and pre-trigger time	3
TM	Set trigger mode	3
XS	Set plotter or recorder status byte for polling	12

Mnemonic	Function	Chapter
CA	Designate alternate character set	8
CP	Character plot	8
CS	Designate standard character set	8
DF	Set default values	2
DI	Set absolute label direction	8
DR	Set relative label direction	8
DT	Define label terminator	8
IM	Input error and polling masks	12
IN	Initialize	2
IP	Input P1 and P2	6
IW	Input window (soft-clip limits)	6
LB	Label ASCII string	8
LO	Set label origin	8
LT	Set line type	7
OA	Output actual position and pen status	10
OC	Output commanded position and pen status	10
OE	Output error	12
OF	Output factors	10
OH	Output hard-clip limits	2
OI	Output identification	10
OO	Output options	10
OP	Output P1 and P2	6
OS	Output plotter status	12
OW	Output window (soft-clip limits)	6
OY	Locate syntax error	12
OZ	Output command string	10
PA	Plot absolute	5
PD	Pen down	5
PR	Plot relative	5
PS	Paper size	2
PU	Pen up	5
RO	Rotate coordinate system	13
SA	Select alternate set	8
SC	Scale	6
SI	Set absolute character size	8
SL	Set character slant	8
SM	Set symbol mode	7
SP	Select pen	5
SR	Set relative character size	8
SS	Select standard set	8
TL	Set tick length	7
VS	Velocity select	5
XT	X-axis tick	7
YT	Y-axis tick	7

What Computer and Programming Language Do I Use?

Any ASCII[1] based computer with an HP-IB interface and a programming language that is capable of sending literal strings of HP-RL and/or HP-GL instructions can be used to remotely control the HP 7090. The literal strings of HP-RL and HP-GL instructions are the only data that the HP 7090 understands. The programming language statements (BASIC, FORTRAN, PASCAL, and etc.) are only used to tell the computer what to do. Refer to the section Sending Instructions to the HP 7090, in this chapter, for examples using various HP computers.

If you have an HP computer, graphics statements for the plotter are available as an extension to BASIC. Sometimes this extension is called AGL (A Graphics Language). AGL statements resemble English so you can understand them, but they translate their functions into HP-GL so that the HP 7090 can understand them. Depending on your computer, AGL statements might be part of the supplied BASIC language set, or they might be available separately in a device called a plotter ROM (read-only memory). You can program the HP 7090 using AGL, HP-GL, and HP-RL in the same program. The HP 17090A Measurement Graphics Software package is available to perform many of your measurement graphics functions if you have an HP 9000, Series 200 computer (models 216, 226, 236; previously models 9816, 9826, 9836).

Initial HP-IB Setup and Addressing

To communicate effectively with the HP 7090, it is important that you completely understand the addressing protocol of your computer. Therefore, you may wish to review this aspect of your computer before proceeding.

Computers with No High Level I/O Statements

On low level computers, addressing devices on the HP-IB bus 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.

[1] American Standard Code for Information Interchange

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.

Computers with High Level I/O Statements

In more powerful computers, higher level input/output (I/O) statements are used to specify device addresses on the HP-IB bus. 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 the user.

Connecting the HP 7090

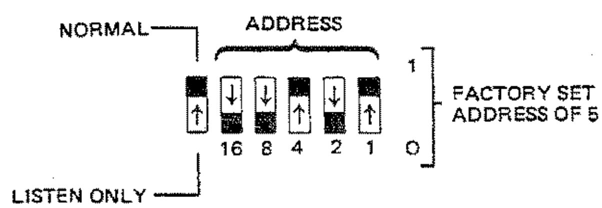
A standard HP-IB cable connected to the HP-IB connector on the rear panel is all that is required to connect the HP 7090 to your computer. Analog input signals are connected to the three CHANNEL input connectors on the front panel. Refer to the HP 7090 operator's manual for complete instructions on the use of the GUARD connectors.

Addressing the HP 7090

The HP 7090 can function in either addressable mode or listen-only mode. In addressable mode, the HP 7090 can function as a talker or as a listener depending on the instructions it receives from the computer. In listen-only mode, it can only listen and it hears all activity on the bus.

Each HP-IB interface can have as many as 15 devices connected to it, set to different specific address codes. When the rear panel LISTEN ONLY/NORMAL switch is set to NORMAL, the HP 7090 can be set to any one of 31 HP-IB addresses, ranging from 0 through 30. Each address can be selected by setting the ADDRESS switches to the appropriate bit positions for the particular address value desired. The address selected establishes the HP 7090's device address. The address is set to 05 at the factory. This corresponds to a listen character of % and a talk character of E. Check the following figure for the factory set ADDRESS switch positions.

NOTE: The positions of the ADDRESS switches are read only at power-up. ■



When using the HP 7090 with an HP desktop computer, do not use address 21 because it is reserved for the desktop computer's address. Address 31 should not be used with any computer because it is the universal unlisten command. When not using an HP desktop computer, be sure the computer and the HP 7090 do not have the same address. The following table lists the switch positions for each address value.

Address Switch Positions

Address Characters		Address Switch Settings					Address Codes		
Listen	Talk	16	8	4	2	1	Decimal	Octal	
SP	@	0	0	0	0	0	0	0	
!	A	0	0	0	0	1	1	1	
“	B	0	0	0	1	0	2	2	
#	C	0	0	0	1	1	3	3	
$	D	0	0	1	0	0	4	4	
%	E	0	0	1	0	1	5	5	← PRESET
&	F	0	0	1	1	0	6	6	
’	G	0	0	1	1	1	7	7	
(	H	0	1	0	0	0	8	10	
)	I	0	1	0	0	1	9	11	
*	J	0	1	0	1	0	10	12	
+	K	0	1	0	1	1	11	13	
,	L	0	1	1	0	0	12	14	
—	M	0	1	1	0	1	13	15	
.	N	0	1	1	1	0	14	16	
/	O	0	1	1	1	1	15	17	
0	P	1	0	0	0	0	16	20	
1	Q	1	0	0	0	1	17	21	
2	R	1	0	0	1	0	18	22	
3	S	1	0	0	1	1	19	23	
4	T	1	0	1	0	0	20	24	
5	U	1	0	1	0	1	21	25	← RESERVED FOR HP DESKTOP COMPUTER ADDRESS
6	V	1	0	1	1	0	22	26	
7	W	1	0	1	1	1	23	27	
8	X	1	1	0	0	0	24	30	
9	Y	1	1	0	0	1	25	31	
:	Z	1	1	0	1	0	26	32	
;	[	1	1	0	1	1	27	33	
<	\	1	1	1	0	0	28	34	
=	]	1	1	1	0	1	29	35	
>	^	1	1	1	1	0	30	36	
?	_	1	1	1	1	1	31	37	← RESERVED FOR UNIVERSAL UNLISTEN COMMAND

How to Use the Examples in This Manual

The examples in this manual are designed primarily to show the use of the instructions with which they appear. If you are new to programming, key in each example. Then you may wish to change some parameters on an instruction and note the results when you run the example. In this way, you will quickly learn how to use the instructions. The examples are presented in two ways. Sometimes they are presented as complete programs, and sometimes they are given as only the pertinent HP-RL or HP-GL strings. These are described in the following paragraphs.

Examples Presented as Complete Programs

Some examples are presented as complete programs, written in the BASIC used by the HP Series 200 computers. These programs assume that you have already connected the HP 7090 and computer, and that paper and pens are loaded.

You may need to make changes to some of the BASIC statements if you are programming in another language, or if the BASIC on your computer is different. Here are the statements that you are most likely to need to change:

ASSIGN @Hp7090 TO 705	This statement identifies the HP 7090 as the recipient of the instructions which are sent as literal strings in OUTPUT @Hp7090 statements. The 7 is the HP-IB select code, and 05 is the HP 7090's HP-IB address. You will need to change this statement according to your version of BASIC, and change the select code/address numbers according to your interface configuration.
OUTPUT @Hp7090	This statement sends the HP-RL and HP-GL instructions to the HP 7090 as literal strings. Your computer may use a statement such as WRITE or PRINT instead of OUTPUT.
OUTPUT @Hp7090 USING	This statement is the same as the OUTPUT @Hp7090 statement, except that it makes use of image specifiers, which control the output format of the HP-RL and HP-GL instructions. "K" is the most common specifier; it suppresses any blank spaces in the output field. Your computer may use a statement such as WRITE, FORMAT, or IMAGE.
ENTER @Hp7090	This statement causes information from the HP 7090 to be received by the computer. Your computer may use a statement such as READ or INPUT.
ENTER @Hp7090 USING	This statement differs from the ENTER @Hp7090 statement in the same way that OUTPUT @Hp7090 USING differs from the OUTPUT @Hp7090 statement.

FOR . . . NEXT X = 3.14	FOR . . . NEXT are BASIC loop statements. X = 3.14 is a variable assignment. Change these statements to whatever is comparable for your computer.

NOTE: You may need an I/O ROM for your computer to use the BASIC statements OUTPUT and ENTER to send or receive information. ■

Examples Presented as HP-RL or HP-GL Strings

When an example is very short, only the strings of HP-RL or HP-GL instructions are listed. They are enclosed in quotation marks because many computer languages define literal strings by placing them inside quotation marks. In these cases, add the statements required by your computer to send the string of instructions. For example, suppose the following string is printed in the manual:

```
"SP1;"
```

First send whatever statements are required to establish the HP 7090 as the recipient of the string (such as ASSIGN @Hp7090 TO 705 on HP Series 200 computers). Then, send the string within a statement similar to those shown here (only a few computers are listed to give you an idea of the variety of statements available):

HP Series 200, BASIC	OUTPUT @Hp7090;"SP1;"
IBM Personal Computer, BASIC 1.10	PRINT #1, "SP1;"
HP 3000, FORTRAN	WRITE(6,10) 10 FORMAT(X,4HSP1;)

The following paragraphs give simple complete programs to send and receive information between the HP 7090 and several specific computers.

Sending Instructions to the HP 7090

Transmitting data from a computer to the HP 7090 is typically accomplished using I/O programming statements such as OUTPUT, WRITE, PRINT, or PRINT#. The following examples of sending program data to the HP 7090 from various computers are only intended to illustrate the necessity for understanding the I/O programming statement protocol implemented by your computer. Each of these examples will cause the HP 7090 to label the identity of the computer sending data, beginning at the X,Y coordinates 1000, 2000. The examples involve sending both character string and numeric data as variables, and constants or literals.

HP Series 80 Example

```
10 PRINTER IS 705
20 A$="SENDING DATA"
30 B=80
40 Y=2000
50 PRINT "SP1;PA1000,",Y
60 PRINT "LBHP SERIES";B;A$;"¤"
70 END
```

Result: HP SERIES 80 SENDING DATA

HP Series 200 HPL Example

```
0: fxd 0;dim A$[13]
1: " SENDING DATA"→A$
2: 2000→Y
3: 200→B
4: wrt 705,"SP1;PA1000,",Y
5: wtb 705,"LBHP SERIES",str(B),A$,3
6: end
```

Result: HP SERIES 200 SENDING DATA

HP Series 200 BASIC Example

```
10    ASSIGN @Hp7090 TO 705
20    A$=" SENDING DATA"
30    B=200
40    Y=2000
50    OUTPUT @Hp7090;"SP1;PA1000,";Y
60    OUTPUT @Hp7090;"LBHP SERIES";B;A$;"␃"
70    END
```

Result: HP SERIES 200 SENDING DATA

HP 1000 Example

This example uses FORTRAN IV and assumes that the system has been configured so that the logic unit number 13 sends data to an HP-IB address of 1. Therefore, the HP 7090 **ADDRESS** switch on the back panel has been set to 1. The program name, VBP1, may be changed to any other legal program name and line numbers as given are only necessary for the format statements. The field width specification of 4 is adequate to plot to the locatior 1000,2000; you may wish to use a field width of 5 or 6 to allow for pen positions up to the hard-clip limits of the HP 7090. A field larger than six characters would only be necessary for scaled data greater than 999 999 or less than −99 999.

```
FTN4,L
      PROGRAM VBP1
   10 IY=2000
   20 IB=1000
   30 WRITE [13,101] IY
   40 WRITE [13,102] IB
  101 FORMAT ["PA1000,",I4]
  102 FORMAT ["LB HP ",I4," SENDING DATA"]
      END
```

Result: HP 1000 SENDING DATA

Receiving Data from the HP 7090

Outputting data from the HP 7090 to the computer is typically accomplished using I/O programming statements such as ENTER, READ, or INPUT. Sometimes these statements are only available in I/O ROMs; check your computer's documentation or ask your HP Sales and Support Office. The following examples of obtaining output data from the HP 709 using various computers are only intended to illustrate the necessity for understanding the I/O programming statement protocol implemented on your computer. Each of these examples commands the pen to move to the coordinates X = 1000, Y = 1000 and then output the current pen position and the HP 7090 identifier string to the computer.

HP Series 80 Example

```
10 PRINTER IS 705
20 PRINT "PA1000,1000;OC;"
30 ENTER 705 ; A,B,C
40 PRINT "OI;"
50 ENTER 705 ; A$
60 DISP A,B,C,A$
70 END
```

Displayed current pen position and identification.

1000	1000
0	7090A

HP Series 200 HPL Example

```
0: fxd 0;dim A$[5]
1: wrt 705,"PA1000,1000;OC;"
2: red 705,A,B,C
3: wrt 705,OI;"
4: red 705,A$
5: dsp A,B,C,A$
6: end
```

Displayed current pen position and identification.

1000 1000 0 7090A

HP Series 200 BASIC Example

```
10    ASSIGN @Hp7090 TO 705
20    OUTPUT @Hp7090;"PA1000,1000;OC;"
30    ENTER @Hp7090;A,B,C
40    OUTPUT @Hp7090;"OI;"
50    ENTER @Hp7090;A$
60    DISP A,B,C,A$
70    END
```

Displayed current pen position and identification.

1000 1000 0 7090A

HP 1000 Example

This example uses FORTRAN IV and assumes that the system has been configured so that the logic unit number 13 sends data to an HP-IB address of 1. Therefore, the HP 7090 ADDRESS switch on the back panel has been set to 1. The program name, VBP2, may be changed to any other legal program name and the line numbers as given are only necessary for the format statements. A field width specification of 5 would be necessary to output pen positions where either the X- or Y-position was greater than 9999 or less than −999. Data is read in from the HP 7090 using free field format so that the commas sent by the HP 7090 between output parameters are interpreted as separators.

```
FTN4,L
      PROGRAM VBP2
      DIMENSION A[3]
10    WRITE [13,101]
20    READ [13,*] IX,IY,IP
30    WRITE [13,103]
40    READ [13,104] A
45    WRITE [1,102] IX,IY,IP
50    WRITE [1,104] A
101   FORMAT ["PA1000,1000;OC"]
102   FORMAT [3I5]
103   FORMAT ["OI"]
104   FORMAT [3A2]
      END
```

Displayed current pen position and identification.

```
 1000 1000    0
7090A
```

Notes

Chapter 2
Recorder Fundamental Concepts

What You'll Learn in This Chapter

This chapter presents the fundamentals that you must understand to successfully program the HP 7090. You will learn about the writing area, the two independent areas that are used for recording and plotting, and how to define a point in these areas. This chapter also explains measurement setup conditions and the two instructions that establish default conditions.

Instructions Covered

DF The Default Instruction
IN The Initialize Instruction
OH The Output Hard-Clip Limits Instruction
PS The Paper Size Instruction

Terms You Should Understand

Plotting area — the area where labeling and plotter pen motion can occur. The HP-GL instruction, IW, defines the location and size of the plotting area. The scaling points P1 and P2 are used with the SC instruction to scale this area.

Recording — the act of filling the buffers or drawing a real-time plot.

Recording Area — the area where real-time and buffered data are plotted. The zero (Z) and full scale (FS) points explicitly define quadrant I and implicitly define quadrants II, III, and IV.

The Writing Area

The writing area is that area of each size of paper in which the pen can draw. The writing area is defined by the hard-clip limits that are established when each size of paper is selected. Except for a margin of approximately ½ inch at the top of the paper, the hard-clip limits allow the pen to draw anywhere on the paper. The margin is required to prevent the paper from moving out from under the pinch wheels when the paper is moved fully forward.

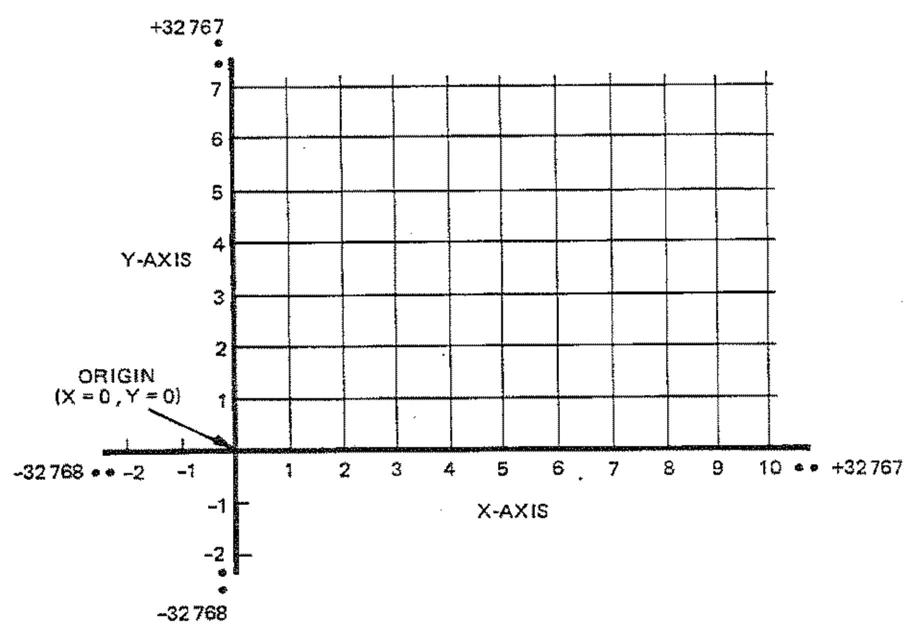
Operation of the HP 7090 is based on two independent scaling areas; the recording area and the plotting area. Both of these areas can be defined to occupy all, or any part, of the writing area. The recording area is defined by the Zero (Z) and Full Scale (FS) scaling points. All analog signals that are measured in the direct and buffered recording modes are drawn only within the recording area. Signal amplitudes that are greater than the defined full scale range are clipped at the boundaries of the recording area.

The plotting area is defined by the soft-clip limits that are established by the HP-GL input window instruction, IW. All labeling and HP-GL programmed pen motion is restricted to the area defined by the soft-clip limits. The soft-clip limits have no effect on where direct and buffered recording signals are drawn and do not restrict pen motion that is initiated from the front panel. The scaling points P1 and P2 are used in conjunction with the HP-GL scaling instruction, SC, to scale the plotting area.

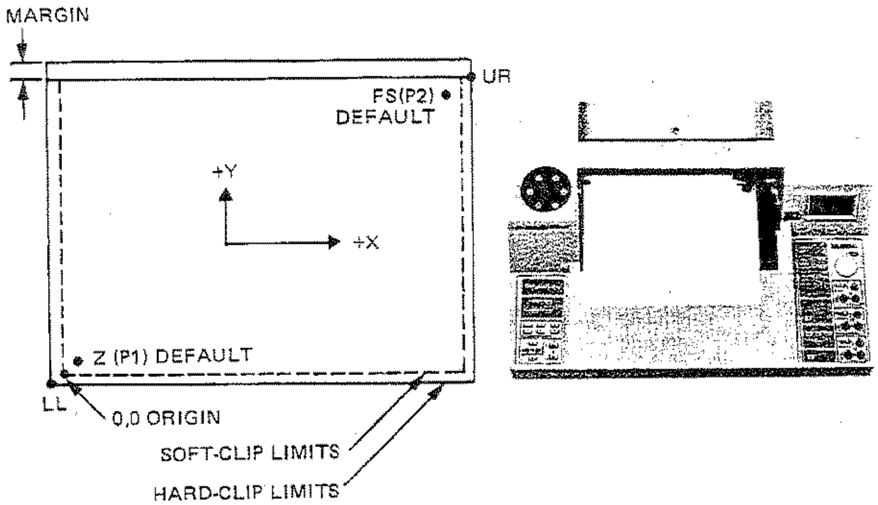
The Reference Coordinate System

The writing area, and all points within it, are referenced to the origin of the two-dimensional Cartesian coordinate system shown in the following illustration. The reference coordinate system is divided (scaled) into plotter units which are always the same size; one plotter unit equals 0.025 mm (0.000 98 in.). There are 40.2 plotter units per millimetre, or approximately 1021 plotter units per inch. Integer Cartesian coordinates are used to define the location of the hard-clip and soft-clip limits. They are also used as parameters in some HP-RL and HP-GL instructions to define the location of the scaling points or to move the pen to a given point within the hard-clip limits.

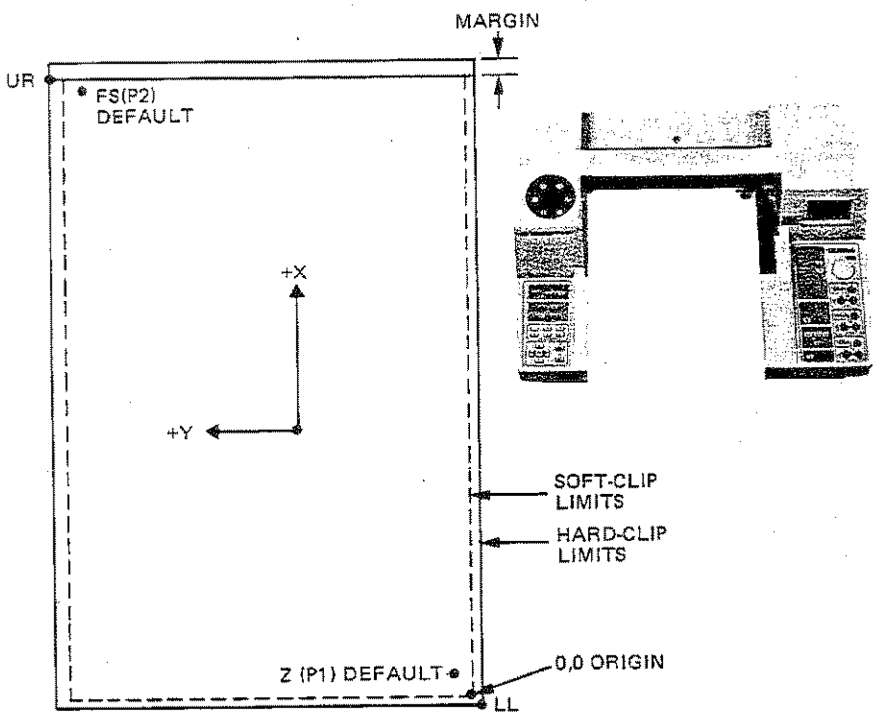
Reference Coordinate System



The location of the reference coordinate system origin and the orientation of the independent (X) and dependent (Y) axes with respect to A and A4 or B and A3 paper sizes are shown in the following diagrams. Hard-clip limits and the approximate default locations of the plotting area soft-clip limits and both sets of scaling points are also shown. Although their default locations are coincident, the zero and full scale points are totally independent of the P1 and P2 points. The following tables show the default plotter unit coordinate values which define the sizes and/or locations of each area and each scaling point on all four paper sizes.



Default Orientation of Reference Coordinate System
(A/A4 Paper)



Default Orientation of Reference Coordinate System
(B/A3 Paper)

Hard-Clip Limits and Default Soft-Clip Limits

Paper Size	Hard-Clip Limits		Soft-Clip Limits	
	X_{LL}, Y_{LL}	X_{UR}, Y_{UR}	X_{LL}, Y_{LL}	X_{UR}, Y_{UR}
A 8.5 × 11 in.	−333, −100	10 703, 7987	0, 0	10 370, 7987
B 11 × 17 in.	−475, −333	16 260, 10 703	0, 0	16 260, 10 370
A4 210 × 297 mm	−322, −100	11 400, 7785	0, 0	11 078, 7785
A3 297 × 420 mm	−525, −322	15 762, 11 400	0, 0	15 762, 11 078

Default Scaling Point Locations

Paper Size	Zero and P1	Full Scale and P2
	Z_X, Z_Y ($P1_X, P1_Y$)	FS_X, FS_Y ($P2_X, P2_Y$)
A 8.5 × 11 in.	160, 447	10 210, 7682
B 11 × 17 in.	865, 160	16 140, 10 210
A4 210 × 297 mm	514, 348	10 564, 7583
A3 297 × 420 mm	325, 514	15 600, 10 564

The Output Hard-Clip Limits Instruction, OH

USES: The OH instruction is used to output the lower-left (LL) and upper-right (UR) coordinates of the current hard-clip limits. This instruction can be used with the IP, IW, and IZ instructions to make use of the entire writing area.

SYNTAX: *OH* terminator

PARAMETERS: None

EXPLANATION: After an OH instruction is received, the LL and UR plotter unit coordinates are output as four numbers in ASCII, separated by commas.

$X_{lower\ left}, Y_{lower\ left}, X_{upper\ right}, Y_{upper\ right}$ CRLF

Changing the soft-clip limits with an IW instruction does not affect the hard-clip limits. Changing the paper size, however, will change the size of the hard-clip limits. The 90-degree rotation function will interchange the hard-clip limit coordinate values. Thus, if the UR Y-axis value is larger than the UR X-axis value, you know that the coordinate axes are rotated from their default orientation.

The Paper Size Instruction, PS

USES: The PS instruction is used to programmatically toggle between A and B, or A3 and A4 paper sizes.

SYNTAX: *PS* paper size terminator

Parameter	Format	Range	Default
paper size	integer	0 to 3 = B or A3 size 4 to 127 = A or A4 size	none

EXPLANATION: This instruction performs the functions of a front-panel paper size change. The new paper size is determined by the parameter and the setting of the rear-panel paper switch. A parameter in the range of 0 to 3 selects either B or A3 size paper, and a parameter in the range of 4 to 127 selects either A or A4 size paper. The PS instruction, however, cannot switch from ISO to ANSI size paper or vice versa. To change from ISO to ANSI size paper, reset the rear-panel paper switch, change the right pinch wheel position, and cycle the HP 7090's power.

Specifying out-of-range parameters sets error 3, and the instruction is ignored.

Setting Up the Recording Area

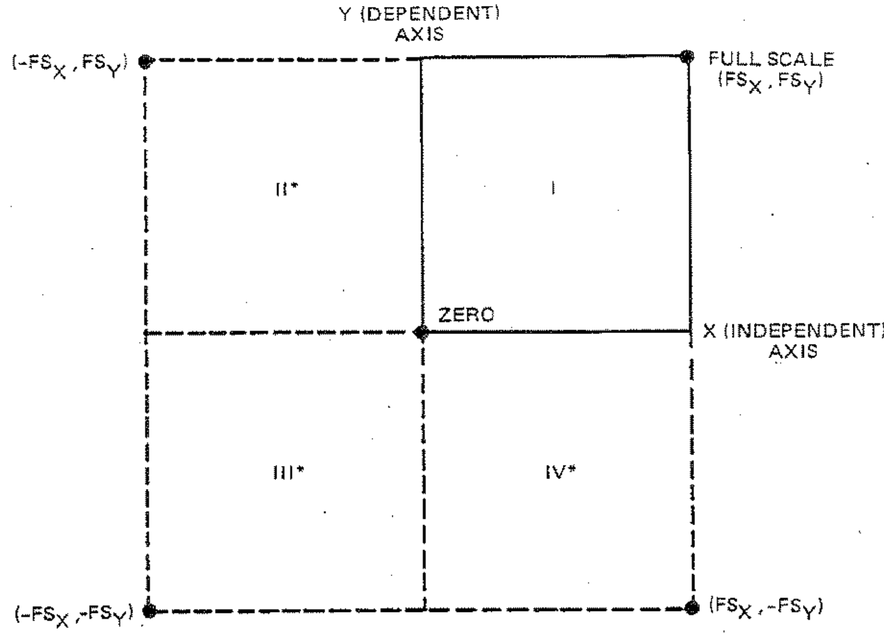
Both direct and buffered recording measurements are displayed in the recording area in accordance with the set up conditions that are established when the measurement is initiated. Therefore, to achieve optimum display results, the measurement set up conditions must be defined based on some prior knowledge of the input signal characteristics. The set up conditions that determine how the measurement is displayed are described in the following paragraphs. The HP-RL instructions that are used to programmatically define each set up condition are explained in Chapter 3.

Defining the Recording Area

The recording area is a Cartesian coordinate system that is mapped onto the reference coordinate system via the zero and full scale points. The locations of the zero and full scale points specifically define quadrant I of the recording area. Quadrant I is automatically extrapolated to set up quadrants II, III, and IV, as shown in the following illustration. The origin of the coordinate system is the zero point and the limits of the four quadrants are expressed in terms of the full scale point.

NOTE: If the zero and full scale points are set such that any portion of quadrants II, III, or IV fall outside of the hard-clip limits, then that quadrant boundary is defined by the hard-clip limits. ■

The Recording Area Boundaries



*Implicity defined when quadrant I is defined by ZERO and FULL SCALE

Channel Selection (Axes Definitions)

The independent (X) and dependent (Y) axes of the recording area must be defined to set up either a versus-chan3 or a versus-time measurement. A versus-chan3 measurement is set up when the independent (X) axis is defined as channel 3 and the dependent (Y) axis is defined as channel(s) 1 and/or 2. A versus-time measurement is set up when the independent (X) axis is defined as time and the dependent (Y) axis is defined as any combination of channels 1, 2, and 3. If more than one channel is specified for the dependent (Y) axis, a plot buffer command will plot each selected channel in order, but a direct recording command will only plot the channel with the lowest selected number.

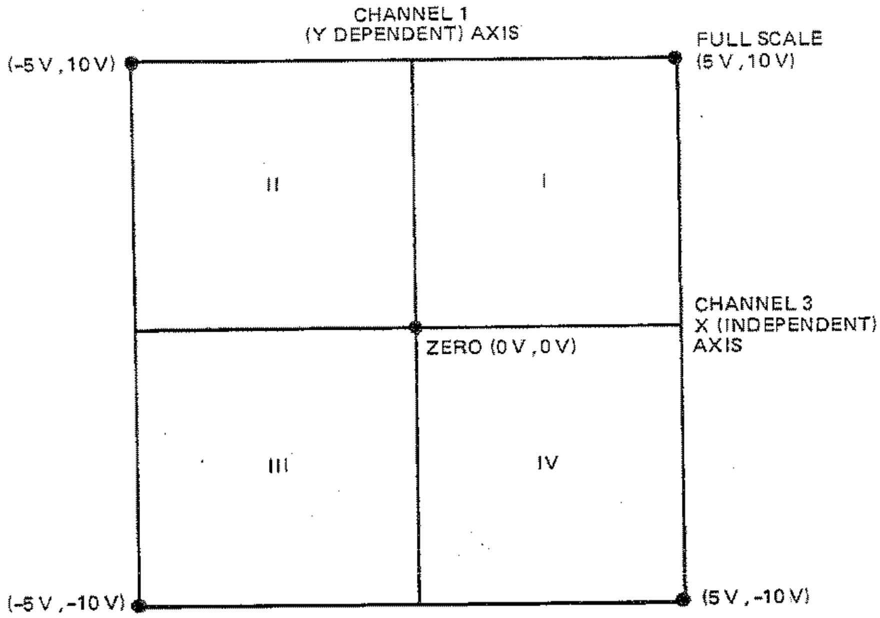
Scaling the Recording Area

The input signal voltage that can be measured and mapped onto each axis of the recording area is determined by the range and dc offset values specified for each channel. The range and dc offset values interact algebraically and scale the axis of the recording area to which the channel is assigned. The range value sets the total number of volts that can be mapped onto the axis and the dc offset value defines the location of the zero voltage reference point of the axis.

A single positive range value between 0.005 and 100 volts defines the positive limit of the axis. An equal voltage with a negative sign is automatically mirrored to define the negative limit of the axis. Thus, the total range of the axis is twice the specified range value.

The following illustration shows the range of the versus-chan3 input signal voltages that can be plotted with the recording area scaled to the indicated range values. A zero input signal voltage in each channel is plotted at the zero point, unless a dc offset voltage is specified. Any input signal amplitude greater than the specified range will turn on the applicable front panel OVERFLOW light to indicate the signal has exceeded the boundaries of the recording area. If this condition occurs during a measurement, those portions of the signal that exceed the boundaries are clipped (not drawn); i.e., in the buffered recording mode the pen lifts while the signal is clipped, but in the direct recording mode the pen draws along the boundary until the signal amplitude is again in bounds. When the signal amplitude is again in bounds, the **OVERFLOW** light goes out.

Range Scaling Example

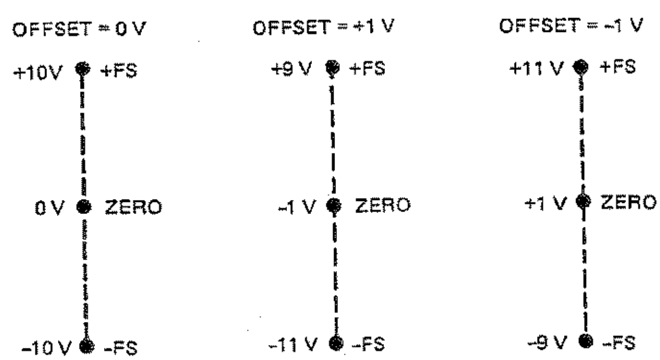


SET UP CONDITIONS:

1. All four quadrants of the recording area are inside the hard-clip limits.
2. The axes are defined as channels 1 and 3 as shown.
3. Channel 1 range = 10 volts.
4. Channel 3 range = 5 volts.

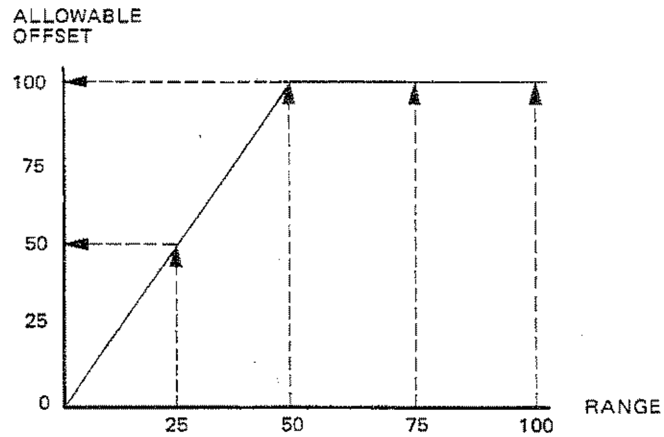
Specifying a dc offset voltage shifts the zero voltage reference point: a positive offset voltage shifts it up; a negative offset voltage shifts it down. Shifting the location of the zero voltage reference point also shifts the range scaling limits by a corresponding amount. The following illustration shows how specifying different offset voltages effectively change the scaling shown in the Range Scaling Example. In the interest of simplicity, the effects on only the vertical axis (channel 1) are shown.

Offset Effects on Range Scaling Limits



The allowable dc offset is limited to ± twice the range value or ±100 volts, whichever is less. The applicable front-panel **OFFSET** light will blink as a warning if this restriction is exceeded. The following diagram graphically illustrates the relationship between range and the allowable dc offset.

Range Versus Allowable Offset



Maximum measurement accuracy is achieved when the recorded signal occupies as much of the recording area as possible. For balanced sinusoidal signals with a median voltage of zero volts, dc offset is not required, and range can be set to the anticipated peak voltage. For signals with a large median dc amplitude, dc offset should be used to shift the range limits so that the recorded signal fills the recording area. The following equations can be used to determine the optimum range and dc offset values.

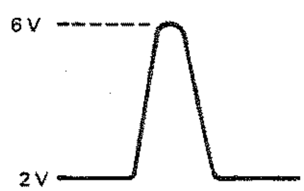
$$\text{Range} = |V_{max} - V_{min}|/2$$

$$\text{Offset} = -(V_{max} + V_{min})/2$$

where: V_{max} = the maximum anticipated voltage

V_{min} = the minimum anticipated voltage

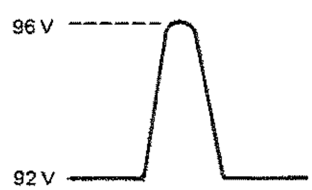
The following example illustrates using these equations to scale the recording area for maximum accuracy with either normal or extreme signal conditions.



Normal Conditions

$$\text{Range} = |V_{max} - V_{min}|/2 = (6-2)/2 = 2$$

$$\text{Offset} = -(V_{max} + V_{min})/2 = -(6+2)/2 = -4$$



Extreme Conditions

$$\text{Range} = |V_{max} - V_{min}|/2 = (96-92)/2 = 2$$

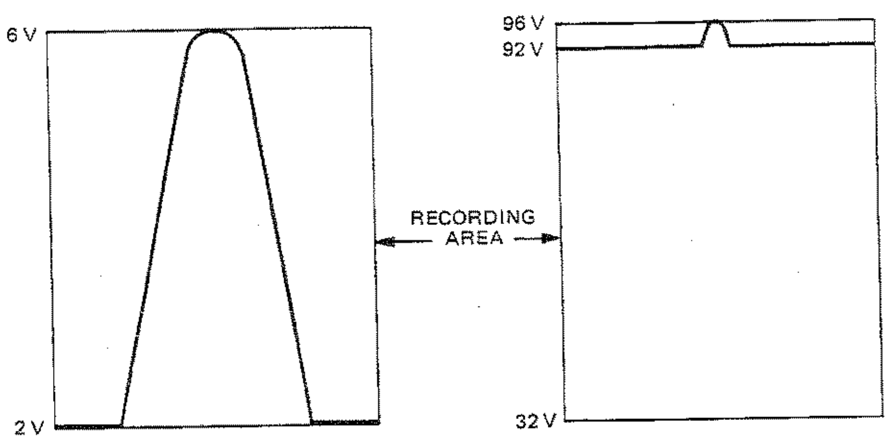
$$\text{Offset} = -(V_{max} + V_{min})/2 = -(96+92)/2 = -94$$

The values computed for the extreme condition cannot be used because the offset value is greater than twice the range; therefore,

IF $|V_{off}| > 2 * RANGE$ or > 100 volts, THEN

1. Set range to $1/3\,|V_{max}|$ or $1/3\,|V_{min}|$, whichever is greater.
2. Set dc offset to 2 * range computed in step 1. Use the sign computed in the offset equation.

The following diagrams illustrate the optimum scaling that can be achieved for the above signals. The measurement accuracy of signals recorded under these conditions is shown in Appendix C.



Range = 2 V
Offset = −4 V

Normal Conditions

Range = 32 V
Offset = −64 V

Extreme Conditions

Total Time

In the buffered recording mode, total time specifies the amount of time required to store 1000 data samples in the buffers. The buffers in all three channels are filled simultaneously, and any given total time applies equally to each buffer. The sample rate for a buffered recording equals 1000 divided by total time.

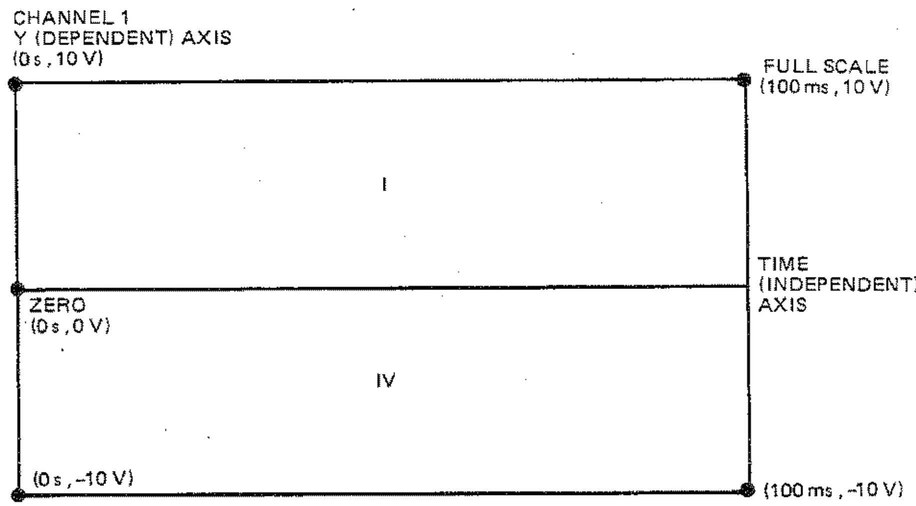
In the direct recording mode, total time specifies the real-time duration of all recordings, including when a versus-chan3 recording is initiated with a manual trigger. Only front-panel initiation allows for a "free-running" versus-chan3 recording that is not bounded by time. In this case, the recording will continue until either the HP-RL instruction MT is executed or the front-panel STOP button is pressed. The sample rate for all direct recording measurements is fixed at 250 samples per second.

The value of total time can vary from 0.03 second to 24 hours in the buffered recording mode or from 1.0 second to 24 hours in the direct recording mode. If total time is set to less than 1.0 second, and a direct recording is initiated, the measurement is performed with a total time value of 1.0 second.

NOTE: For direct recording measurements with a duration of 1.0 hour or longer, use the special HP 24-hour recording paper to avoid the possibility of excessive ink bleeding. ■

The following illustration shows how total time is mapped onto the independent (X) axis when a versus-time measurement is initiated. Note that only quadrants I and IV are applicable to these measurements.

Total Time Scaling Example



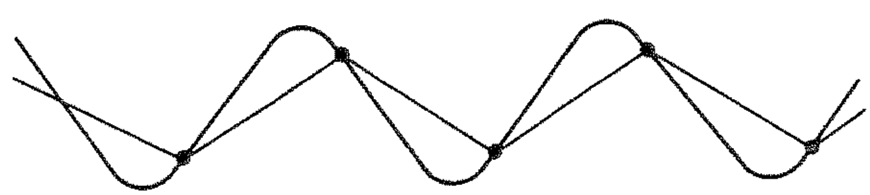
SET UP CONDITIONS:

1. Quadrants I and IV of the recording area are inside the hard-clip limits.
2. The axes are defined as channel 1 versus time, as shown.
3. Channel 1 range = 10 volts.
4. Total time = 100 milliseconds.

Signal Reconstruction and Aliasing

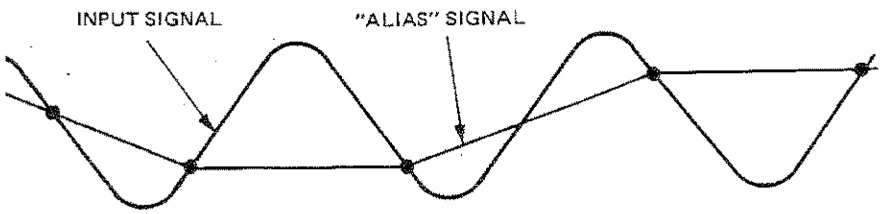
In order to accurately reconstruct a waveform, the input signal must be sampled often enough for its characteristics to be apparent from the sampled data. The Nyquist Sampling Theorem states that the minimum frequency required for sampling is twice the highest frequency in the signal.

As an illustration, consider a sinewave input with frequency f. If this sinewave is sampled with frequency 2f, the correct frequency (but not amplitude or phase) may be reconstructed from the samples.



When sampling at a higher rate, the reconstruction is improved.

However, if the signal is sampled at a rate less than 2f, a result called "aliasing" occurs when the signal is reconstructed:



In this case, the samples appear to be part of a sinewave with a frequency less than f. These "alias" frequencies occur when the sampling rate is not at least twice that of the highest frequency in the input signal. However, to clearly see the shape of the input signal, a sample rate should be selected to obtain a minimum of 10 samples per cycle of the highest frequency in the input signal.

The maximum sample rate of the HP 7090 is 33 333 samples per second. Therefore, input frequencies greater than 3.3 kHz are reconstructed with less than 10 samples per cycle.

Drawing Grids

The recording area can be filled with plotted grid lines to aid in analyzing plotted data. Each axis of quadrant I can be filled independently with 1 to 100 grid lines. The number of grid lines should normally be specified in increments which correspond with range and total time values. The grid currently defined for quadrant I can be drawn in any combination of the four quadrant areas. The grid lines drawn in adjacent quadrants will align to form a continuous grid.

A standard centimetre grid can also be drawn on each size paper. This grid is completely independent of the recording area. For A and A4 paper sizes, the standard grid is always 25 × 18 cm. For B and A3 paper sizes, the standard grid is always 38 × 25 cm.

Refer to the explanations for HP-RL instructions DG and GL, in Chapter 3, for complete information on drawing grids.

Triggering

The HP 7090 includes manual, external, and internal triggering modes. The trigger generated in these modes is the actual event which starts data recording in either the direct or buffered recording modes. The exception to this rule is when internal inside window triggering is selected and the measurement is a buffered recording. In this case, the trigger terminates the measurement. The time between initiating a measurement and locating the trigger is defined as the trigger search time. The following paragraphs explain the post- and pre-triggering cabilities and the triggering action in each trigger mode.

Post-Trigger/ Pre-Trigger

A post-trigger value (+) specifies the amount of time that must elapse, after a valid trigger occurs, before a direct recording begins or the buffers start filling. The start of data recording is subject to any specified post-trigger time delay, except it is ignored when internal inside window triggering is selected and the measurement is a buffered recording.

A pre-trigger value (−) specifies the portion of total time that is used for storing input signal data that occurs just before receipt of a *valid* trigger. The pre-trigger capability is applicable only when internal or external triggering is selected and the measurement is a buffered recording.

The pre-trigger data is stored in the buffers and continually updated until a valid trigger is received. Upon receipt of a valid trigger, the pre-trigger data currently in the buffers is frozen, and the rest of the buffer space is allowed to fill. The sample rate for storing the pre-trigger data is determined by the specified total time. Therefore, if the trigger search time is less than the specified pre-trigger time, the actual trigger search time is used instead of the pre-trigger time.

The pre-trigger value should never exceed the value specified for total time. Exceeding this restriction causes the front-panel **PRE-TRIGGER** menu light to blink, indicating an invalid condition, and the pre-trigger value is set internally to the value of total time.

Manual Trigger Mode

In this mode, the trigger is generated when the MS instruction is received or the front panel **FILL BUFFER** or **RECORD DIRECT** buttons are pressed. There is no trigger search time and any specified pre-trigger value is ignored.

External Trigger Mode

In this mode, data recording starts when the HP-IB GET command is received or when a negative going TTL level transition occurs on the **EXT TRIG** input.

Internal Trigger Mode

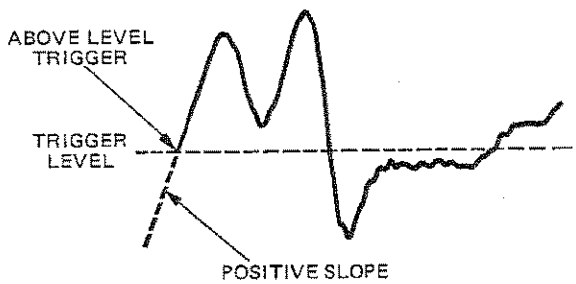
Internal triggering is based on the amplitude and slope of the input signal on channel 1. The triggering options include above and below level triggering and inside or outside window triggering. The following paragraph explain each internal triggering option.

Above Level

This mode defines rising (positive) slope triggering at a specified trigger level value. The trigger is generated when the input signal rises from below the trigger level value and becomes equal to or greater than the trigger level value. Refer to the following diagram.

The front panel TRIGGER LEVEL menu LED will blink any time the trigger level exceeds the voltage limits of the recording area. In this case, the trigger level value is set internally to the voltage limit of the recording area.

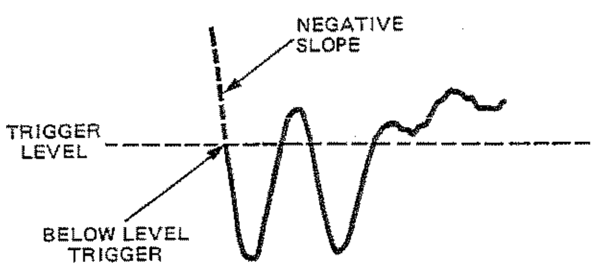
Above Level Triggering



Below Level

This mode defines falling (negative) slope triggering at a specified trigger level value. The trigger is generated when the input signal falls from above the trigger level value and becomes equal to or less than the trigger level value. Refer to the following diagram.

Below Level Triggering

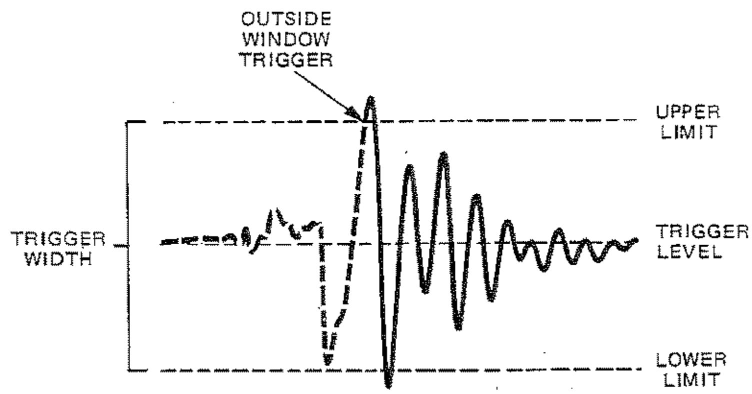


Outside Window

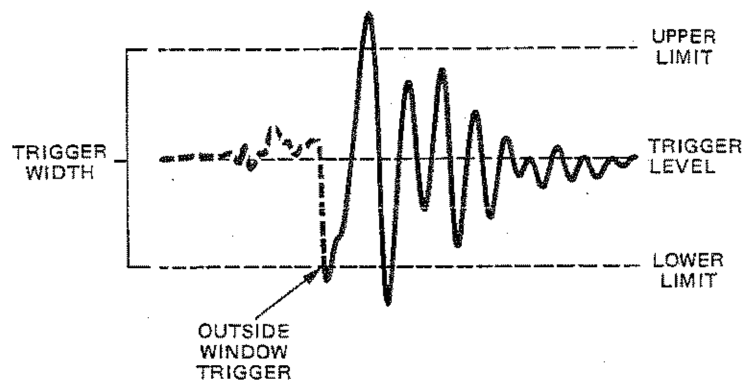
This mode uses a trigger window that is centered about the specified trigger level value. The total width of the window is specified by the trigger width value. Half of the width value is added to the trigger level value to define the upper limit and half of the width value is subtracted from the trigger level value to define the lower limit. The trigger is generated when a sample of the input signal exceeds the trigger window limits in either direction. Refer to the following diagram.

The front panel TRIGGER WIDTH menu LED will blink any time the width value exceeds twice the range value. If a trigger window limit is in violation of the voltage limits of the recording area, it is set internally to the voltage limit of the recording area.

Outside Window Triggering



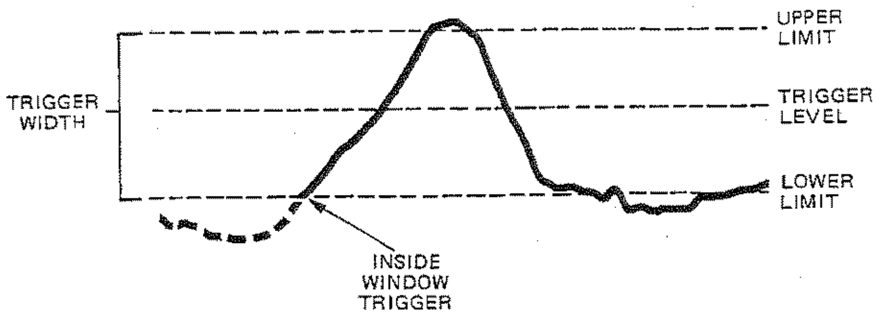
OR



Inside Window

This mode uses a trigger window that is centered about the specified trigger level value. The total width of the window is specified by the trigger width value. Half of the width value is added to the trigger level value to define the upper limit and half of the width value is subtracted from the trigger level value to define the lower limit.

For direct recording and direct data acquisition measurements, the trigger is generated when the signal is sampled and found to be inside of the window limits. Any specified post-trigger time delay is executed before data recording begins. Any specified pre-trigger time is ignored. The following diagram illustrates the inside window triggering action for direct recording and direct data acquisition measurements.



For buffered recording measurements the inside window trigger action is different. After a buffered measurement is initiated, each data sample is compared with the trigger window limits, and the buffers start filling immediately. The buffer contents are continually updated until 1000 data samples in a row are within the trigger window limits. The trigger is generated coincident with receipt of the sample that completely fills the buffers. This trigger freezes the contents of the buffers and terminates the measurement.

For buffered recordings with inside window triggering, any specified post-trigger time delay is ignored, but pre-trigger time is recognized. In this case, pre-trigger data is defined as the data samples which immediately precede the inside window samples, rather than the samples just prior to the trigger. Since the buffer capacity is always 1000, the required number of consecutive inside window samples is reduced proportionately with any specified pre-trigger time. The following equation computes the required number of consecutive inside window data samples for any combination of pre-trigger time and total time.

$$S = 1000(1 - T_P/T_T)$$

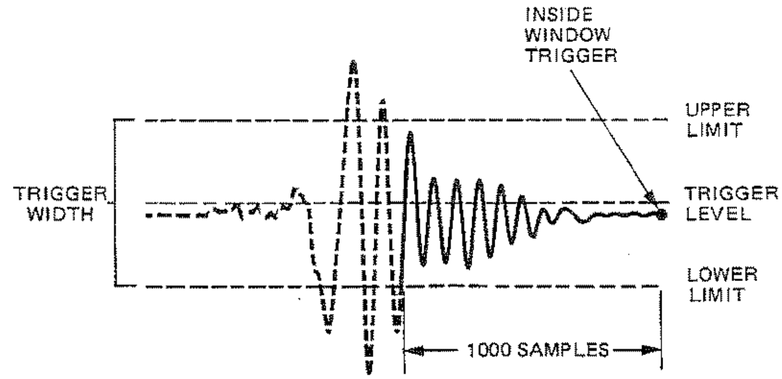
where: S = Total number of consecutive inside window samples

T_P = Pre-trigger time

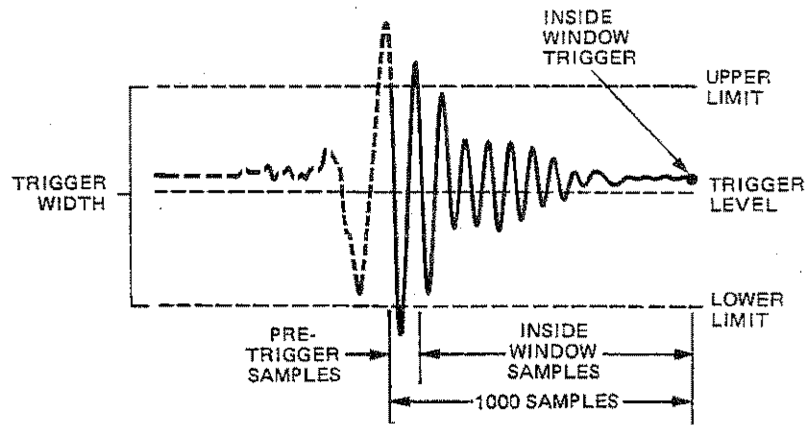
T_T = Total time

The following diagrams illustrate inside window triggering for buffered recording measurements.

Inside Window Triggering (Buffered Recording Measurements)



OR



The Default Instruction, DF

USES: The DF instruction sets certain plotter functions to a predefined state. Use it to ensure that graphics parameters such as character size and slant or plotter scaling are not inherited from a previous program.

SYNTAX: *DF* terminator

PARAMETERS: None

EXPLANATION: A DF instruction sets the conditions shown in the following table. It is equivalent to executing all of the indicated instructions. The DF instruction has no effect on recorder functions or the following plotter functions.

- Locations of P1 and P2
- Current pen and its position
- 90-degree rotation

Plotter Default Conditions

Function	Equivalent Instructions	Conditions
Plotting mode	PA;	Absolute (PA)
Relative character direction	DR 1 , 0 ;	Horizontal
Line type	LT;	Solid line
Line pattern length	LT;	4% of the diagonal distance between P1 and P2
Input window	IW;	Set according to current size paper (see Reference Coordinate System)
Relative character size	SR;	Width = 0.75% of ($P2_X - P1_X$) Height = 1.5% of ($P2_Y - P1_Y$)
Symbol mode	SM;	Off
Tick length	TL;	tp = tn = 0.5% of ($P2_X - P1_X$) for Y-tick and 0.5% of ($P2_Y - P1_Y$) for X-tick
Standard character set	CS 0;	Set 0
Alternate character set	CA 0;	Set 0
Character set selected	SS;	Standard
Character slant	SL 0;	0 degrees
Mask value	IM 233 , 0 , 0;	Recognizes all errors, except position overflow
Scaling	SC;	Plotter units
Pen velocity	VS;	75 cm/s (30 in./s)
Label terminator	DT ETX	ETX (ASCII decimal equivalent 3)
Label origin	LO;	Label starts at current pen position

The Initialize Instruction, IN

USES: The IN instruction returns all recorder and plotter conditions to their initial power-on state.

SYNTAX: *IN* terminator

PARAMETERS: None

EXPLANATION: The IN instruction sets the plotter functions to the same conditions as the DF instruction and the recorder functions to the conditions shown in the following table. It is equivalent to executing all of the indicated instructions, plus it sets these additional conditions.

- The pen is raised, but its position is not changed.
- All error conditions are cleared.
- The plotter and recorder status bytes are cleared and bit position 3 of the plotter status byte is set true (1) to indicate the HP 7090 is initialized.
- The HP-IB I/O is reset to initial conditions. The plotter status byte (OS) is used as the HP-IB status byte (STB).
- The channel buffers are flushed.
- The rotation function is set to the default orientation (0 degrees).
- The hard-clip limits and both sets of scaling points (Z/FS and P1/P2) are set to the default coordinate values for the currently selected paper size. These values are listed in the tables given under The Reference Coordinate System paragraph, in this chapter.

NOTE: The positions of the rear-panel switches are not checked when an IN instruction is executed. These positions are checked only at power-on. ■

Recorder Default Conditions

Function	Equivalent Instructions	Conditions
Range	IR;	All channels set to 10.0 volts
Offset	SV;	All channels set to 0.0 volts
Grid type and Grid divisions	DG;	User-defined grid A/A4 — X,Y = 25,18 B/A3 — X,Y = 38,25
Grid quadrants	DG;	Quadrant 1
Total time	TB;	1.0 second
Post-trigger time	TG;	0 second
Pre-trigger time	TG;	0 second
Trigger level	TA;	0.0 volts
Trigger width	TA;	0.0 volts
Trigger mode	TM;	Manual
Channel selection	RE 9;	Channel 1 versus time
Data output	DO;	Continuous, all channels, ASCII format, no trigger status
Sample delay	SD;	0.1 second

Notes

Chapter 3
The Recorder Setup Instructions

What You'll Learn in This Chapter

Now that you understand the fundamentals, you are ready to learn how to program the measurement setup conditions. This chapter explains the HP-RL instructions that specify the channels that are to be measured and how the data is to be displayed in the recording area. This includes defining the size and location of the recording area and setting range, dc offset, and total time. The instructions for drawing grids, specifying trigger conditions, setting the real-time clock, and labeling the setup conditions are also explained.

Instructions Covered

DG The Draw Grid Instruction
GL The Set Grid Divisions Instruction
IR The Set Full Scale Range Instruction
IT The Set Real-Time Clock Instruction
IZ The Set Zero and Full Scale Instruction
QA The Query Trigger Level and Width Instruction
QB The Query Total Time Instruction
QG The Query Post- and Pre-Trigger Time Instruction
QL The Query Grid Divisions Instruction
QM The Query Recording Mode Instruction
QR The Query Full Scale Range Instruction
QT The Query Trigger Mode Instruction
QU The Query Trigger Time Instruction
QV The Query DC Offset Instruction
QW The Query Real-Time Clock Instruction
QZ The Query Zero and Full Scale Instruction
RE The Set Recording Mode Instruction
SV The Set DC Offset Instruction
TA The Set Trigger Level and Width Instruction
TB The Set Total Time Instruction
TD The Label Time and Date Instruction
TG The Set Post- and Pre-Trigger Time Instruction
TM The Set Trigger Mode Instruction

The Set Recording Mode Instruction, RE

USES: The RE instruction defines the independent (X) and dependent (Y) axes of the recording area in terms of the input channels and/or time. Use it to select the channel(s) to be recorded against time or against channel 3.

SYNTAX: *RE* mode terminator
or
RE terminator

Parameter	Format	Range	Default
mode	integer	3 through 15 except 6, 10, and 14	9

EXPLANATION: The mode parameter is the sum of valid combinations of the bit values shown below. Each bit value included in the integer sets the corresponding bit to a logic 1, and thus defines the recording area axes as indicated.

NOTE: Any bit position that is not set to logic 1 has a bit value of zero. ■

Bit Value	Bit Position	Logic State	Axes Definition
0	0	0	X-axis = Channel 3
1		1	X-axis = Time
2	1	1	Y-axis = Channel 3
4	2	1	Y-axis = Channel 2
8	3	1	Y-axis = Channel 1

The values shown in the following table are the only valid combinations of the bit values that can be specified by the mode parameter. Specifying any bit value combination that leaves either axis undefined, or assigns the same channel to both axes, sets error 3 and the instruction is ignored.

Axes Definitions	Permitted Mode Parameter Values
Versus-Chan3 Measurements	
Channel 1 vs. Channel 3	8
Channel 2 vs. Channel 3	4
Channels 1 and 2 vs. Channel 3	12
Versus-Time Measurements	
Channel 1 vs. Time	9
Channel 2 vs. Time	5
Channel 3 vs. Time	3
Channels 1 and 2 vs. Time	13
Channels 1 and 3 vs. Time	11
Channels 2 and 3 vs. Time	7
Channels 1, 2, and 3 vs. Time	15

Executing an RE instruction without a parameter (RE;) sets the default definition: channel 1 versus time (RE 9;).

If more than one channel is specified for the Y-axis, a plot buffer (PL) instruction will plot each selected channel in order, but a direct recording (MS 0) instruction will only plot the channel with the lowest specified number.

The Query Recording Mode Instruction, QM

USES: The QM instruction outputs the current definitions of the independent (X) and dependent (Y) axes of the recording area.

SYNTAX: *QM* terminator

PARAMETERS: None

EXPLANATION: When a QM instruction is received, the decimal equivalent value for the current definitions of the X- and Y-axes of the recording area are output in ASCII as follows:

mode CRLF

All possible values and their meanings are shown in the following table. Refer to the RE instruction for additional information.

Mode Value	Axes Definitions
3	Channel 3 vs. Time
4	Channel 2 vs. Channel 3
5	Channel 2 vs. Time
7	Channels 2 and 3 vs. Time
8	Channel 1 vs. Channel 3
9	Channel 1 vs. Time
11	Channels 1 and 3 vs. Time
12	Channels 1 and 2 vs. Channel 3
13	Channels 1 and 2 vs. Time
15	Channels 1, 2, and 3 vs. Time

The Set Zero and Full Scale Instruction, IZ

USES: The IZ instruction provides program control over the locations of the zero (Z) and full scale (FS) points. Use this instruction to define the size and location of the recording area. Refer to Defining the Recording Area in Chapter 2.

SYNTAX: *IZ* Z_X, Z_Y (, FS_X, FS_Y) terminator
or
IZ terminator

Parameter	Format	Range	Default
X,Y coordinates	integer	−32 768 to 32 767	dependent on current paper size

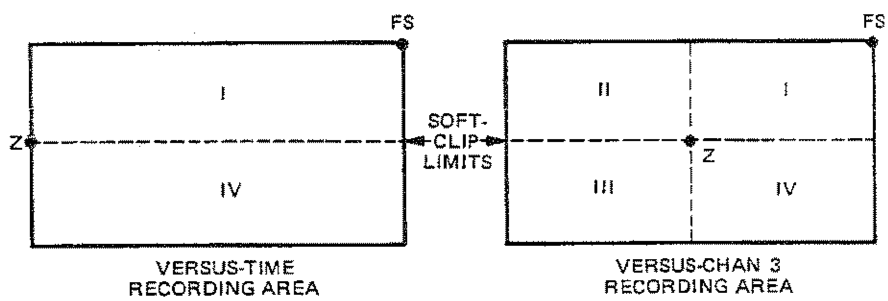
EXPLANATION: The coordinates of zero and full scale are specified in the order shown and are interpreted as plotter units. Neither point can be located outside of the current hard-clip limits. If the location of either point is specified outside of the hard-clip limits, its coordinates are changed auto matically to the hard-clip limits and no error is set. Parameters less than −32 768 or greater than 32 767 will cause error 3 and the instruction is ignored.

Specifying the full scale coordinates is optional. However, if the full scale coordinates are omitted, then full scale tracks zero and its coordinates change so that the X- and Y-distances between zero and full scale do not change. If zero continues to move after full scale encounters the hard-clip limits, the X- and Y-distances between zero and full scale will change.

If either coordinate of full scale is set equal to the corresponding coordinate of zero, then that coordinate of full scale is incremented by one plotter unit. This ensures a two-dimensional recording area.

An IZ instruction without parameters (IZ;) sets zero and full scale to the default coordinate values for the currently selected size of paper. Refer to The Reference Coordinate System, Chapter 2, for the default coordinate values of zero and full scale.

The default zero and full scale points define most of the writing area as quadrant I. Other useful recording areas are when either quadrants I and IV or all four quadrants are approximately centered on the page. A subroutine similar to the following program can be used to set up the recording area for either a versus-time (quadrants I and IV) or a versus-chan3 (all quadrants) measurement. Run the program and press the front-panel **ZERO** and **FULL SCALE** buttons to verify the resultant recording areas shown next.




```
10    !Program to set up recording area for versus-time
20    !or versus-chan3 measurements.  Requires response
30    !to select measurement.
40    ASSIGN @Hp7090 TO 705
50    OUTPUT @Hp7090;"IN;QZ;"
60    ENTER @Hp7090;A,B,C,D
70    Zx=INT((C-A)/2+A)
80    Zy=INT((D-B)/2+B)
90    INPUT J !0 for versus-time or 1 for versus-chan3.
100   !Press continue.
110   IF J=1 THEN 140
120   OUTPUT @Hp7090;"IZ";A;Zy;C;D
130   IF J=0 THEN 150
140   OUTPUT @Hp7090;"IZ";Zx;Zy;C;D
150   END
```

Lines 50 and 60 obtain the default coordinate values of the zero and full scale points.

Lines 70 and 80 compute the coordinate values of the mid-points on each axis.

Line 90 allows user to select two quadrants for versus-time setup or four quadrants for versus-chan3 setup.

Line 120 establishes a two-quadrant recording area inside the default zero and full scale area.

Line 140 establishes a four-quadrant recording area inside the default zero and full scale area.

The Query Zero and Full Scale Instruction, QZ

USES: The QZ instruction outputs the current X- and Y-coordinates of the zero and full scale points.

SYNTAX: *QZ* terminator

PARAMETERS: . None

EXPLANATION: When a QZ instruction is received, the current coordinates of zero and full scale are output in plotter units as four numbers in ASCII, separated by commas.

Z_X, Z_Y, FS_X, FS_Y CRLF

The Set Full Scale Range Instruction, IR

USES: The IR instruction sets the preamplifier range sensitivity of each analog input channel. Use the instruction to define the maximum input signal amplitude that can be plotted within the recording area.

SYNTAX: *IR* FS_1 (,FS_2 (,FS_3)) terminator
or
IR terminator

Parameter	Format	Range	Default
FS_1, FS_2, FS_3	real	0.0050 V to 100.0 V	10.0 V

EXPLANATION: The parameters are in volts and specify the full scale range of channels 1, 2, and 3, in order. The parameters for channels 2 and 3 may be omitted without changing their current range values. Omitting only the third parameter changes channel 3 to its default value (10.0 V).

Each parameter specified is limited to four digits within the indicated range. Any additional digits are truncated, but no error is set. Out-of-range parameters set error 3, and the instruction is ignored.

An IR instruction without parameters (IR;) sets the range of all three channels to 10.0 volts (default).

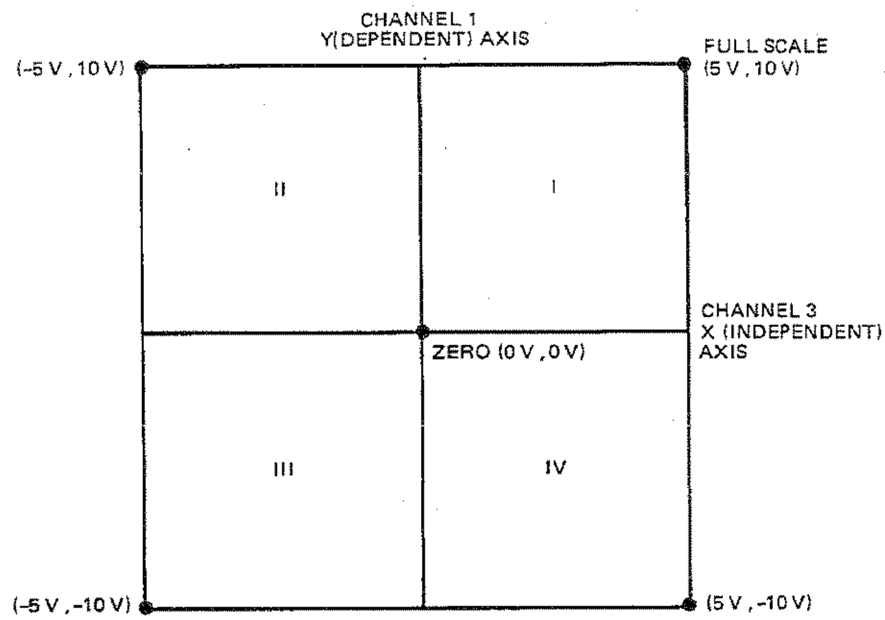
The following program and diagram illustrates one use of the IR instruction.

```
10    ASSIGN @Hp7090 TO 705
20    OUTPUT @Hp7090;"IN;QZ;"
30    ENTER @Hp7090;A,B,C,D
40    Zx=INT((C-A)/2+A)
50    Zy=INT((D-B)/2+B)
60    OUTPUT @Hp7090;"IZ";Zx;Zy;C;D
70    OUTPUT @Hp7090;"RE8;"
80    OUTPUT @Hp7090;"IR10,10,5;"
90    END
```

Lines 10 through 60 establish a four-quadrant recording area inside the default zero and full scale area.

Line 70 defines the independent (X) axis as channel 3 and the dependent (Y) axis as channel 1.

Line 80 sets the full scale range of channels 1 and 2 to 10 volts and channel 3 to 5 volts. Channel 2 is not relevant for this set up, but its parameter cannot be omitted because the range for channel 3 must be set. Any value ≤ 100 could be used for channel 2.



The Query Full Scale Range Instruction, QR

USES: The QR instruction outputs the current range settings for all three input channels.

SYNTAX: *QR* terminator

PARAMETERS: None

EXPLANATION: When a QR instruction is received, the current range setting of all three input channels is output in volts as three numbers in ASCII, separated by commas.

range 1, range 2, range 3 CRLF

Each range value is a decimal number between 0.0050 and 100.0 volts.

The Set DC Offset Instruction, SV

USES: The SV instruction provides program control over the zero voltage position of each channel's input signal. Use it in conjunction with the IR instruction to scale the recording area to match the input signal amplitudes.

SYNTAX: *SV* offset 1 (, offset 2 (, offset 3)) terminator
or
SV terminator

Parameter	Format	Range	Default
offset 1, 2, 3	real	±100 volts	0.0 volts

EXPLANATION: The parameters are in volts and specify the zero voltage offset of channels 1, 2, and 3, in order. The parameters for channels 2 and 3 may be omitted without changing their current offset values. Omitting only the third parameter changes channel 3 to its default value (0.0 V). Any requested value within the ±100 volt range is displayed when the front panel **OFFSET** menu function is selected. However, the **OFFSET** menu light will be blinking if the requested offset exceeds twice the corresponding range value. The response from a QV instruction will also indicate the requested value and whether or not it is valid.

Each parameter specified is limited to three digits within the indicated range. Any additional digits are truncated, but no error is set. Out-of-range parameters set error 3, and the instruction is ignored.

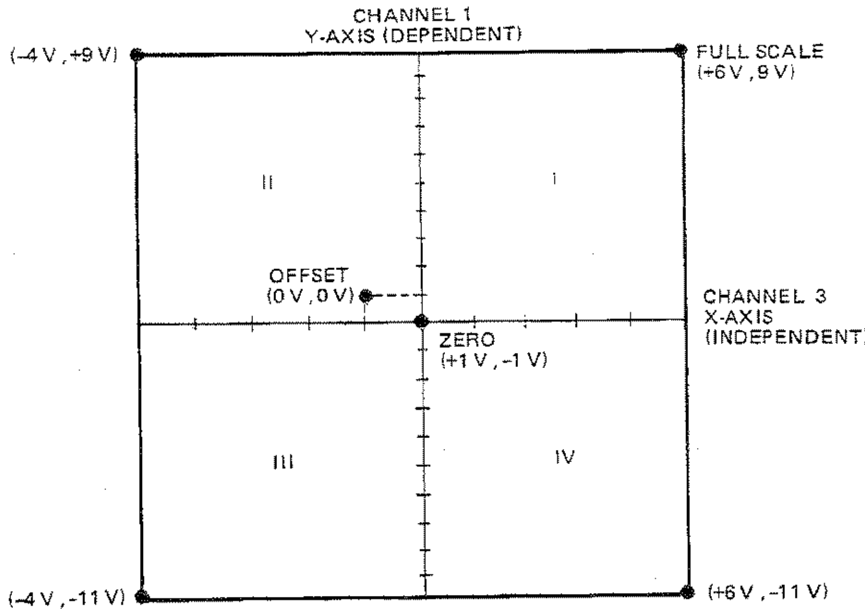
An SV instruction without parameters (SV;) sets the offset of all three channels to 0.0 volts (default).

The following program and diagram illustrate the effect on range scaling when the SV instruction is used to offset zero volts from the zero scaling point.

```
10    ASSIGN @Hp7090 TO 705
20    OUTPUT @Hp7090;"IN;QZ;"
30    ENTER @Hp7090;A,B,C,D
40    Zx=INT((C-A)/2+A)
50    Zy=INT((D-B)/2+B)
60    OUTPUT @Hp7090;"IZ";Zx;Zy;C;D
70    OUTPUT @Hp7090;"RE8;"
80    OUTPUT @Hp7090;"IR10,10,5;"
90    OUTPUT @Hp7090;"SV1,0,-1;"
100   END
```

Lines 10 through 80 are identical to the example shown for the set full scale range instruction, IR.

Line 90 sets the offset of channel 1 to +1 volt and channel 3 to −1 volt. Channel 2 is not relevant for this set up, but its parameter cannot be omitted because the offset for channel 3 must be set.



The Query DC Offset Instruction, QV

USES: The QV instruction outputs the currently requested dc offset for each input channel and whether or not the values are valid.

SYNTAX: *QV* terminator

PARAMETERS: None

EXPLANATION: When a QV instruction is received, four numbers are output in ASCII, separated by commas as shown below. The first three numbers are the currently requested dc offset voltage values. The fourth number indicates whether or not one or more of the dc offset values are invalid: 0 = valid, 1 = invalid.

offset 1, offset 2, offset 3, validity CRLF

NOTE: Any dc offset value that exceeds ±100 volts or twice the corresponding range value is an invalid condition. ■

The Set Total Time Instruction, TB

USES: The TB instruction specifies the duration of a recording. Use it to define total time in the buffered recording mode or the sweep time in the direct recording mode.

SYNTAX: *TB* time (, units) terminator
or
TB terminator

Parameter	Format	Range	Default
time	real	seconds: 0.030 to 100 minutes: 1.00 to 100 hours: 1.00 to 24.0	1
units	integer	0 = seconds 1 = minutes 2 = hours	0

EXPLANATION: The time parameter is limited to three significant digits within the indicated ranges. Any additional digits are truncated. The units parameter defines the units of time. If the unit parameter is omitted, the units of time default to seconds.

A TB instruction without parameters (TB;) sets total time to 1.0 second (default).

Total time may be set to any value from 30 milliseconds to 24 hours in the buffered recording mode or from 1.0 second to 24 hours in the direct recording mode. If the time value is set to less than 1.0 second, and a direct recording is initiated, the measurement is performed with a total time value of 1.0 second.

Out-of-range parameters set error 3, and the instruction is ignored.

The following program shows one use of the TB instruction. The diagram illustrates the conditions that are set up by the program and how total time is mapped onto the independent (X) axis when a versus-time recording is initiated. Note that only quadrants I and IV are applicable to a versus-time recording.

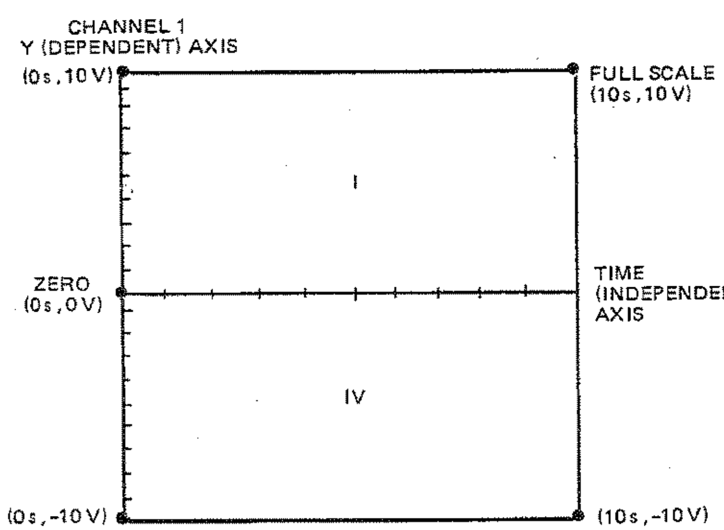
```
10    ASSIGN @Hp7090 TO 705
20    OUTPUT @Hp7090;"IN;QZ;"
30    ENTER @Hp7090;A,B,C,D
40    Zy=INT((D-B)/2+B)
50    OUTPUT @Hp7090;"IZ";A;Zy;C;D
60    OUTPUT @Hp7090;"RE9;"
70    OUTPUT @Hp7090;"IR10;"
80    OUTPUT @Hp7090;"TB10,0;"
90    END
```

Lines 10 through 50 establish a two-quadrant recording area inside the default zero and full scale area.

Line 60 defines the independent (X) axis as time and the dependent (Y) axis as channel 1.

Line 70 sets the full scale range of channel 1 to 10 volts.

Line 80 sets total time to 10 seconds.



The Query Total Time Instruction, QB

USES: The QB instruction outputs the currently specified duration of a recording.

SYNTAX: *QB* terminator

PARAMETERS: None

EXPLANATION: When a QB instruction is received, the current total time value is output as two numbers in ASCII, separated by a comma.

time , units CRLF

where: time = 0.030 to 100 (seconds)
1.00 to 100 (minutes)
1.00 to 100 (hours)

units = 0 (seconds)
1 (minutes)
2 (hours)

The Draw Grid Instruction, DG

USES: The DG instruction draws a user-defined grid or a fixed centimetre grid to aid in analysis of plotted data.

SYNTAX: *DG* grid type terminator
or
DG terminator

Parameter	Format	Range	Default
grid type	integer	0 = user-defined grid 1 = centimetre grid	0

EXPLANATION: A DG instruction with no parameter (DG ;), or a parameter of zero (DG 0 ;), draws a user-defined grid within the currently defined recording area. The user-defined grid is drawn in accordance with the conditions specified by the most recent GL instruction.

A DG instruction with a parameter of one (DG 1 ;) draws a standard centimetre grid that is dependent only on the currently selected paper size. For A and A4 paper sizes, the standard grid is always 25 × 18 cm. For B and A3 paper sizes, the standard grid is always 38 × 25 cm.

NOTE: The standard centimetre grid **is not** subject to the conditions specified by the GL instruction. ■

Grids of both types are drawn using the currently selected pen. Refer to the GL instruction for an example of drawing grids.

The Set Grid Divisions Instruction, GL

USES: The GL instruction specifies the number of divisions in the user-defined grid and the quadrants where it is to be drawn by the DG instruction.

SYNTAX: *GL* X-axis divisions, Y-axis divisions (, quadrants) terminator
or
GL terminator

Parameter	Format	Range	Default	
			A/A4	B/A3
X-axis divisions	integer	1 to 100	25	38
Y-axis divisions	integer	1 to 100	18	25
quadrants	integer	1 to 15	1	1

EXPLANATION: The X-axis divisions parameter specifies the number of grid divisions that are to be drawn vertically along the X-axis in any given quadrant.

The Y-axis divisions parameter specifies the number of grid divisions that are to be drawn horizontally along the Y-axis in any given quadrant.

The quadrants parameter defines the quadrants where the user-defined grid is to be drawn. Any quadrant combination can be selected by specifying an integer which is the sum of any combination of the bit values shown below. Each bit value included in the integer sets the corresponding bit to a logic 1; thus designating the indicated quadrant.

NOTE: Any bit position that is not set to a logic 1 has a value of zero. ■

Bit Value	Bit Position	Logic State	Definition
1	0	1	Quadrant 1
2	1	1	Quadrant 2
4	2	1	Quadrant 3
8	3	1	Quadrant 4

The parameter values for drawing the user-defined grid in any given quadrant combination are shown in the following table. If the quadrants parameter is omitted, the grid location defaults to quadrant I.

Value	Quadrants Specified
0	Error 3 (instruction ignored)
1	1
2	2
3	1 and 2
4	3
5	1 and 3
6	2 and 3
7	1, 2, and 3
8	4
9	1 and 4
10	2 and 4
11	1, 2, and 4
12	3 and 4
13	1, 3, and 4
14	2, 3, and 4
15	1, 2, 3, and 4

A GL instruction without parameters (GL;) establishes default grid divisions based on the currently selected paper size and specifies the grid is to be drawn only in quadrant I. For A and A4 paper sizes, the default grid divisions are X = 25 and Y = 18. For B and A3 paper sizes, the default grid divisions are X = 38 and Y = 25.

Parameters which include a decimal fraction are truncated to an integer, and the instruction is executed without setting an error. Out-of-range parameters set error 3, and the instruction is ignored.

NOTE: The soft-clip limits established by the IW instruction have no effect on the user-defined grid or the standard centimetre grid. ■

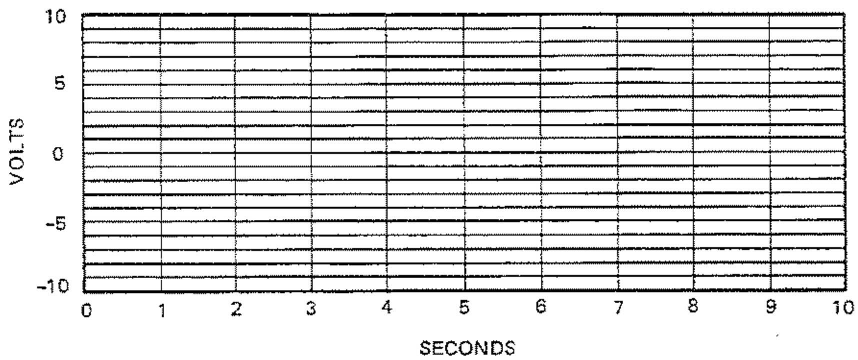
The following example illustrates the use of the DG and GL instructions. The program is a modified version of the program used to illustrate the TB instruction. The grid divisions set by the GL instruction correspond with the specified range and total time values. Note that quadrants I and IV are specified to correspond with the recording area that is defined for a versus-time measurement.

```
10      ASSIGN @Hp7090 TO 705
20      OUTPUT @Hp7090;"IN;QZ;"
30      ENTER @Hp7090;A,B,C,D
40      Zy=INT((D-B)/2+B)
50      OUTPUT @Hp7090;"IZ";A;Zy;C;D
60      OUTPUT @Hp7090;"RE9;"
70      OUTPUT @Hp7090;"IR10;"
80      OUTPUT @Hp7090;"TB10,0;"
90      OUTPUT @Hp7090;"GL10,10,9;"
100     OUTPUT @Hp7090;"SP1;DG0;"
110     END
```

Lines 10 through 80 are identical to the program used to illustrate the TB instruction.

Line 90 specifies a 10 × 10 grid in both quadrants I and IV.

Line 100 selects pen 1 and draws the user-defined grid specified in line 9



The Query Grid Divisions Instruction, QL

USES: The QL instruction outputs the currently specified number of divisions in the user-defined grid and the quadrants where it is to be drawn.

SYNTAX: *QL* terminator

PARAMETERS: None

EXPLANATION: When a QL instruction is received, the number of grid divisions in each axis and the quadrants where the grid is to be drawn are output as three numbers in ASCII, separated by commas.

X-axis divisions, Y-axis divisions, quadrants CRLF

The first two numbers are the actual number of divisions that will be drawn in each quadrant. The third number is the sum of the bit values which define the quadrants where the grid is to be drawn. All possible bit values and their meaning are shown in the following table.

Value	Quadrants Specified
1	1
2	2
3	1 and 2
4	3
5	1 and 3
6	2 and 3
7	1, 2, and 3
8	4
9	1 and 4
10	2 and 4
11	1, 2, and 4
12	3 and 4
13	1, 3, and 4
14	2, 3, and 4
15	1, 2, 3, and 4

The Set Trigger Mode Instruction, TM

USES: The TM instruction selects the triggering mode that is to be used to start data recording. Refer to Triggering, Chapter 2.

SYNTAX: *TM* mode(,options) terminator
or
TM terminator

Parameter	Format	Range	Default
mode	integer	0 = manual 1 = external 2 = internal	0
options	integer	0 = above level 1 = below level 2 = inside window 3 = outside window	0

EXPLANATION: The mode parameter selects either the manual, external, or internal triggering mode. The options parameter is ignored when either manual or external triggering is selected.

The options parameter selects the slope triggering options that are available only when internal (channel 1) triggering is specified. The options parameter defaults to 0 (above level) when no value is specified.

A TM instruction without parameters (TM;) selects manual triggering mode (default).

Out-of-range parameters set error 3, and the instruction is ignored.

The Query Trigger Mode Instruction, QT

USES: The QT instruction outputs the currently selected trigger mode.

SYNTAX: *QT* terminator

PARAMETERS: None

EXPLANATION: When a QT instruction is received, the trigger mode status is output as two numbers in ASCII, separated by a comma.

mode, option CRLF

where: mode = 0 (manual)
1 (external)
2 (internal)

options = 0 (above level)
1 (below level)
2 (inside window)
3 (outside window)

NOTE: The options parameter is always 0 for mode parameters of 0 or 1. ■

The Set Trigger Level and Width Instruction, TA

USES: The TA instruction specifies the trigger level for above and below level triggering and the trigger level and width for inside and outside window triggering. Refer to Triggering, Chapter 2.

SYNTAX: *TA* level (, width) terminator
or
TA terminator

Parameter	Format	Range	Default
level	real	±200 volts	0 volts
width	real	0 to 200 volts	0 volts

EXPLANATION: The level parameter specifies the voltage level where above or below level triggering occurs. When inside or outside window triggering is selected, the level parameter specifies the center voltage point of the trigger window.

The width parameter is in volts and specifies the total width of the trigger window that is centered on the trigger level voltage. The width parameter is applicable only when inside or outside window triggering is selected. The default value (0 volts) is set if the width parameter is omitted.

Each parameter specified is limited to three significant digits within the indicated ranges. Any additional digits are truncated, but no error is set. Out-of-range parameters set error 3, and are ignored. Subsequent parameters are also ignored, but the instruction is executed with any prior valid parameter.

Any requested value within the indicated ranges is displayed when the front-panel menu selects the **TRIGGER LEVEL** or **TRIGGER WIDTH** positions. However, the **TRIGGER LEVEL** or **TRIGGER WIDTH** menu LED will be blinking if the requested value exceeds the current voltage limits of the recording area. The response from a QA instruction will also indicate the requested value and whether or not the values are valid.

A TA instruction without parameters (TA;) sets both the trigger level and width to zero volts (default).

The Query Trigger Level and Width Instruction, QA

USES: The QA instruction outputs the currently requested trigger level and width voltages and whether or not the values are valid.

SYNTAX: *QA* terminator

PARAMETERS: None

EXPLANATION: When a QA instruction is received, three numbers are output in ASCII, separated by commas as shown below. The first two numbers are the currently requested trigger level and width voltage values. The third number indicates whether or not one or both of the values are invalid: 0 = valid, 1 = invalid.

level, width, validity CRLF

The Query Trigger Time Instruction, QU

USES: The QU instruction outputs the time the last trigger occurred.

SYNTAX: *QU* terminator

PARAMETERS: None

EXPLANATION: When a QU instruction is received, the time the last trigger occurred is output as three numbers in ASCII, separated by commas.

hours , minutes , seconds CRLF

where: hours = 0 to 23
minutes = 0 to 59
seconds = 0 to 59

NOTE: The QU instruction outputs the time the HP 7090 is powered-up until a measurement is performed. The time returned is dependent on the setting of the real-time clock. ■

The Set Post- and Pre-Trigger Time Instruction, TG

USES: The TG instruction specifies either a post-trigger time delay or the amount of total time that is used to record pre-trigger data. Refer to Triggering, Chapter 2.

SYNTAX: *TG* time (, units) terminator
or
TG terminator

Parameter	Format	Range	Default
time	real	±0.00 to 100 (seconds) ±1.00 to 100 (minutes) ±1.00 to 24.0 (hours)	0
units	integer	0 = seconds 1 = minutes 2 = hours	0

EXPLANATION: The sign of the time parameter selects the function: positive values specify post-trigger time delay and negative values specify pre-trigger time. Any time within the indicated ranges may be specified for the post-trigger capability, but the pre-trigger time is restricted to 100 percent of the currently specified total time. The time parameter used is limited to three significant digits and any additional digits are truncated.

The units parameter defines the units of time. If the units parameter is omitted, the units of time default to seconds.

A TG instruction without parameters (TG;) sets the post- and pre-trigger time to zero seconds (default).

Out-of-range parameters set error 3, and the instruction is ignored.

Any requested value within the indicated ranges is displayed when the front panel menu selects the POST TRIGGER (+)/PRE TRIGGER (-) position. However, the menu LED will be blinking if a pre-trigger time is requested that exceeds the current total time value. The response from a QG instruction will also indicate the requested time and whether or not the value is valid.

NOTE: The post-trigger capability cannot be used when internal inside window triggering is selected and the measurement is a buffered recording. The pre-trigger capability is applicable only when internal or external triggering is selected and the measurement is a buffered recording. ■

The Query Post- and Pre-Trigger Time Instruction, QG

USES: The QG instruction outputs the currently requested post- or pre-trigger time and whether or not the time value is valid.

SYNTAX: *QG* terminator

PARAMETERS: None

EXPLANATION: When a QG instruction is received, three numbers are output in ASCII, separated by commas as shown below. The first two numbers indicate the requested post- or pre-trigger time. The second number indicates the units of the first number. The sign of the first number indicates the function: positive for post-trigger time delay and negative for pre-trigger time.

The third number indicates whether or not the time value is valid: 0 = valid, 1 = invalid. An invalid time is indicated only if a pre-trigger time in excess of total time is requested.

time , units , validity CRLF

where: time = ±0.00 to 100 (seconds)
±1.00 to 100 (minutes)
±1.00 to 24.0 (hours)

units = 0 (seconds)
1 (minutes)
2 (hours)

The Set Real-Time Clock Instruction, IT

USES: The IT instruction sets the time and date of the internal real-time clock.

SYNTAX: *IT* year , month , day , hours , minutes terminator

Parameter	Format	Range	Default
year	integer	0 to 99	none
month	integer or character	1 to 12 or JAN to DEC	
day	integer	1 to 31	
hours	integer	0 to 23	
minutes	integer	0 to 59	

EXPLANATION: Values in the indicated ranges must be specified for all parameters. The month parameter may be specified as either a number from 1 to 12 or as the three letter abbreviation for the month. The day parameter range is 1 to 31; however, the day is set to 1 if more days than the month can have are specified (i.e., Feb 29 would set the day to Feb 1). The hours parameter is specified in "24-hour clock" terms rather than 12-hour AM/PM terms.

Executing an IT instruction with a missing or out-of-range parameter sets error 3, and the instruction is ignored.

NOTE: Seconds on the real-time clock are automatically set to zero when an IT instruction is executed. ■

The Query Real-Time Clock Instruction, QW

USES: The QW instruction outputs the current time and date of the real-time clock.

SYNTAX: *QW* terminator

PARAMETERS: None

EXPLANATION: When a QW instruction is received, the current time and date are output as six numbers in ASCII, separated by commas.

year , month , day , hours , minutes , seconds CRLF

where:
- year = 00 to 99
- month = 01 to 12
- day = 01 to 31
- hours = 00 to 23
- minutes = 00 to 59
- seconds = 00 to 59

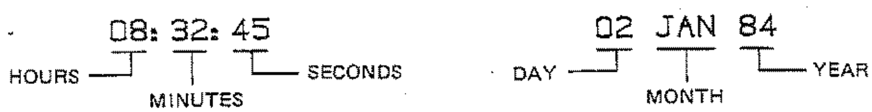
The Label Time and Date Instruction, TD

USES: The TD instruction provides the means to label the set up conditions for a recording, the current date and time, and the time that the trigger occurred.

SYNTAX: *TD* label type terminator
or
TD terminator

Parameter	Format	Range	Default
label type	integer	0 = time and date 1 = time 2 = date 3 = trigger time 4 = set up conditions	0

EXPLANATION: All labeled values for time, date, and trigger time are determined by the real-time clock. A TD instruction with no parameter (TD ;) or a parameter of zero (TD 0 ;) prints the time and date label. The time (TD 1 ;), date (TD 2 ;), and trigger time (TD 3 ;) can all be labeled separately. However, until a measurement is performed, the trigger time defaults to the time that the HP 7090 was last turned on. The format for time and date labels is as follows:



The label format for the default set up conditions (TD 4 ;) is shown below. The label changes automatically to reflect the current set up conditions.

```
RANGES:    10.00V    10.00V    10.00V
OFFSETS:   0.0V      0.0V      0.0V
TOTAL TIME:  1.00S
POST-TRIG:  0.0S
TRIGGER:  MAN
```

The TD instruction labels the specified data using the currently selected pen. The direction, size, and slant of the characters in the label are determined by the values of the DI, DR, SI, SR, and SL instructions in effect when the TD instruction is executed.

The specified label begins at the current pen position, or the label can be shifted from the current pen position using the CP or LO instructions.

NOTE: When labeling the set up conditions (TD 4 ;), the CP instruction has an effect on only the first line of the label. Subsequent lines in the label will be started at the carriage return point. Refer to the LB instruction for an explanation of the conditions which determine the location of the carriage return point. ■

An out-of-range parameter sets error 3, and the instruction is ignored.

Chapter 4
The Recorder Measurement Instructions

What You'll Learn in This Chapter

In this chapter you will learn how to start and stop measurements, plot the buffer contents, and stream data from the HP 7090 to a computer. Finally, a practical example is given to illustrate using many of the HP-RL and HP-GL instructions.

Instructions Covered

DO The Set Data Output Instruction
MS The Measurement Start Instruction
MT The Measurement Terminate Instruction
PL The Plot Buffer Instruction
QD The Query Direct A/D Data Sample Instruction
QI The Query Buffered A/D Data Sample Instruction
SD The Set Sample Delay Instruction

The Measurement Start Instruction, MS

USES: The MS instruction is used to programmatically initiate either direct or buffered recording measurements.

SYNTAX: *MS* measurement terminator
or
MS terminator

Parameter	Format	Range	Default
measurement	integer	0 = direct recording 1 = buffered recording	0

EXPLANATION: Executing an MS instruction is equivalent to pressing one of the front panel MEASUREMENT CONTROL buttons. An MS instruction with no parameter (MS;), or a parameter of 0 (MS 0;) initiates the direct recording mode. An MS instruction with a parameter of 1 (MS 1;) initiates the buffered recording mode. The specified measurement will be performed according to the current setup conditions.

An out-of-range parameter sets error 3, and the instruction is ignored.

All HP-RL and HP-GL instructions, except QC and QS, will terminate a buffered recording (MS 1) during trigger search or a direct recording (MS 0) at any time. After the trigger is found in a buffered recording, a PL or QI instruction can be executed without terminating the measurement. The PL instruction will plot the buffer contents as the buffers continue filling. The QI instruction will copy data points from the buffers and send them to a computer as the buffers continue filling. In either case, any specified post-trigger time delay must elapse before the specified operation is performed.

The Measurement Terminate Instruction, MT

USES: The MT instruction is used to programmatically terminate either direct and buffered recording measurements or the data acquisition mode.

SYNTAX: *MT* terminator

PARAMETERS: None

EXPLANATION: Executing an MT instruction is equivalent to pressing the front-panel STOP button.

The Plot Buffer Instruction, PL

USES: The PL instruction is used to initiate plotting of the current buffer contents.

SYNTAX: *PL* terminator

PARAMETERS: None

EXPLANATION: Executing a PL instruction is equivalent to pressing the front-panel PLOT BUFFER button. The buffer contents for each selected channel are plotted in order, using the pen with the same number as the channel(s).

The Set Data Output Instruction, DO

USES: The DO instruction defines how data samples are to be output when the data acquisition mode is subsequently initiated by the QD (direct) or QI (buffered) instructions.

SYNTAX: *DO* channel(s), # of samples, format, trigger status terminator

or

DO terminator

Parameter	Format	Range	Default
channel(s)	integer	0 = channels 1, 2, and 3 1 = channel 1 2 = channel 2 3 = channel 3	0
# of samples	integer	0 to 32 767 (0 = continuous sampling)	0
format	integer	0 = ASCII 1 = binary	0
trigger status	integer	0 = no trigger status 1 = send trigger status	0

EXPLANATION: Executing a DO instruction with parameters in the indicated ranges defines the contents and the number of samples, or sample sets, that will be output when the data acquisition mode is subsequently invoked by a QD or QI instruction. The first and second parameters specify the channels that are to be sampled and the number of samples, or sample sets, that are to be output. The third parameter specifies whether the samples, or sample sets, are to be output in ASCII or binary format. The fourth parameter specifies whether or not the trigger status is to be output with each sample, or sample set.

A DO instruction without parameters (DO ;) sets the default conditions for data sample output. The default conditions specify continuous sampling from all channels in ASCII format without trigger status (DO 0,0,0,0;). Unless a specific DO instruction is executed prior to invoking the data acquisition mode, data samples will be output in accordance with the default conditions.

The samples, or sample sets, are output in the following form:

chX, (trig,) . . . chX, (trig) CRLF (ASCII format)

chX (trig) . . . chX (trig) (binary format)

or

ch1, ch2, ch3, (trig,) . . . ch1, ch2, ch3, (trig) CRLF (ASCII format)

ch1 ch2 ch3 (trig) . . . ch1 ch2 ch3 (trig) (binary format)

In ASCII format, each channel sample is output as a four digit decimal number with a minus sign if the value is negative. The resolution of the sample voltage is determined by the range setting of the corresponding channel. The trigger status, if included, is either an ASCII zero or one: zero indicates the trigger has not occurred, one indicates the trigger has occurred. If the channel sample value exceeds the limits of the range/offset values, either 999 or −999 is output for the sample value.

In binary format, each channel sample of the 12-bit A/D converters is output as two bytes. The most significant four data bits are the low four bits of the first byte and the lower eight data bits are the second byte. The trigger status, if included, is a separate byte of either binary zero or one: zero indicates the trigger has not occurred, one indicates the trigger has occurred. If the channel sample value exceeds the limits of the range/offset values, the bits of the channel sample bytes are set as follows:

Overload Condition	First Byte	Second Byte
Positive direction	11111111	11111111
Negative direction	11110000	00000000

Binary output is in terms of A/D converter counts. The A/D conversion has a total span of 3948 A/D counts that are centered around 2048 A/D counts (2048 ± 1974 A/D counts). This means that + full scale volts is mapped to 4022 A/D counts, zero volts is mapped to 2048 A/D counts, and − full scale volts is mapped to 74 A/D counts. To convert binary to volts use the following formula:

$$\text{Volts(i)} = \left(\frac{\text{Binary(i)} - 2048}{1974}\right)\text{Full Scale Range(i)} - \text{DC Offset(i)}$$

where: (i) = channel 1, 2, or 3

If a finite number of samples are specified, the END message (the EOI line) is set true concurrently with output of the last byte. In ASCII format, the END message is set when the line feed character is output. In binary format, the END message is set when the last byte of the last sample set is output.

The Set Sample Delay Instruction, SD

USES: The SD instruction sets the time interval between samples, or sample sets, for subsequent data transfers initiated by the QD instruction.

SYNTAX: *SD* time terminator
or
SD terminator

Parameter	Format	Range	Default
time	real	0.002 to 1000 seconds	0.1 second

EXPLANATION: The time parameter determines the time interval between each A/D converter sample. If the computer has not completed reading a sample when it is time for the next sample, the new sample is discarded and bit 3 of the recorder status byte is set to indicate data has been lost. The number of samples requested by DO will always be sent even if some interim samples are discarded. Refer to the QS instruction, Chapter 12.

An SD instruction with no parameter (SD;) sets the time interval to 0.1 second (default).

The minimum time interval that can be specified without data loss is dependent on the content and format of the data and the speed of the computer. The following table gives typical minimum time intervals for transfering one complete sample set of data using the HP series 200 computers.

ASCII Data Transfers
(Binary Data Transfers)

Number of Channels in Sample Set	String Variable		Array Variable	
	No Trigger Status	With Trigger Status	No Trigger Status	With Trigger Status
1	0.006 s (0.002 s)	0.007 s (0.003 s)	0.006 s (0.004 s)	0.010 s (0.007 s)
3	0.017 s (0.006 s)	0.017 s (0.006 s)	0.022 s (0.011 s)	0.026 s (0.013 s)

The Query Direct A/D Data Sample Instruction, QD

USES: The QD instruction initiates the data acquisition mode. Use it to stream data samples from the A/D converters to a computer.

SYNTAX: *QD* terminator

PARAMETERS: None

EXPLANATION: When the QD instruction is executed, and the HP 7090 is requested to talk, data samples are output from the A/D converters to the computer. The number of samples to be sent are specified by the DO instruction and the rate at which the samples are taken is specified by the SD instruction. The default values for the DO and SD instructions are used if specific DO and SD instructions are not executed prior to the QD instruction. These default values specify continuous sampling of all channels in ASCII format at a 0.1 second rate.

NOTE: If sample n + 1 is ready for output before the computer has read sample n, then sample n + 1 is discarded and bit 3 of the recorder status byte is set to indicate data has been lost. Refer to the QS instruction, Chapter 12. ■

The data acquisition mode is automatically exited after the number of samples requested by the DO instruction are output. This mode will be exited prematurely if any instruction, including QC and QS, is received. After the received instruction is processed, another QD instruction is required to invoke the data acquisition mode.

The Query Buffered A/D Data Sample Instruction, QI

USES: The QI instruction initiates the transfer of the channel buffe contents to a computer.

SYNTAX: *QI* terminator

PARAMETERS: None

EXPLANATION: When the QI instruction is executed, the currently buffered data samples are output in accordance with the conditions specified by the DO instruction. Any number of samples, or sample sets, up to the maximum capacity (1000) of the buffers may be requested by the DO instruction. The sample rate specified by the SD instruction is ignored, and the data samples are sent as fast as the computer can receive them. The following table shows typical minimum time intervals for transferring one complete sample set of data using the HP series 200 computers.

ASCII Data Transfers
(Binary Data Transfers)

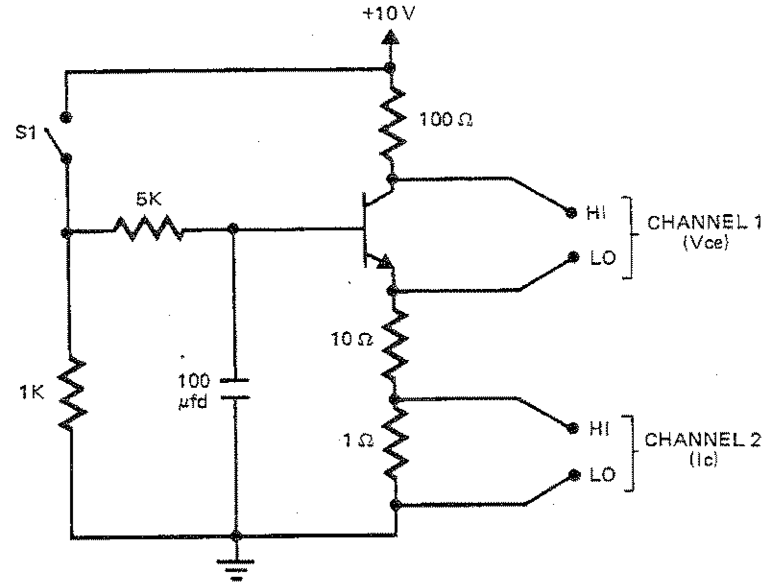
Number of Channels in Sample Set	String Variable		Array Variable	
	No Trigger Status	With Trigger Status	No Trigger Status	With Trigger Status
1	0.0057 s (0.0016 s)	0.0064 s (0.0018 s)	0.0058 s (0.004 s)	0.008 s (0.0052 s)
3	0.0168 s (0.004 s)	0.0175 s (0.0046 s)	0.017 s (0.0088 s)	0.018 s (0.010 s)

The data acquisition mode is automatically exited after the number of samples requested by the DO instruction are output. This mode will be exited prematurely if any instruction, including QC and QS, is received. After the received instruction is processed, another QI instruction is required to invoke the data acquisition mode.

A Practical Measurement Example

The following program and plot illustrate how buffered data can be transferred to a computer, manipulated, and then plotted to obtain data that cannot be measured directly.

In this example, channel 1 and channel 2 are connected to a simple transistor circuit as shown. This allows simultaneous versus-time measurement of the emitter/collector voltage (V_{ce}) and collector current (I_c) when switch S1 is closed. The V_{ce} and I_c data samples are transferred to the computer and multiplied together to compute the power (P_{dis}) that must be dissipated by the transistor at any given time. The measured and computed data values are also used as the Y-coordinate parameters of the PA (plot absolute) instructions which plot the actual data curves. The Y-data values are plotted versus time along the X-axis by using FOR-NEXT loops to generate the X-coordinate values in the PA instructions. Refer to Chapters 5, 6, and 8 for an explanation of plotting, scaling, and labeling instructions that are used in the program.

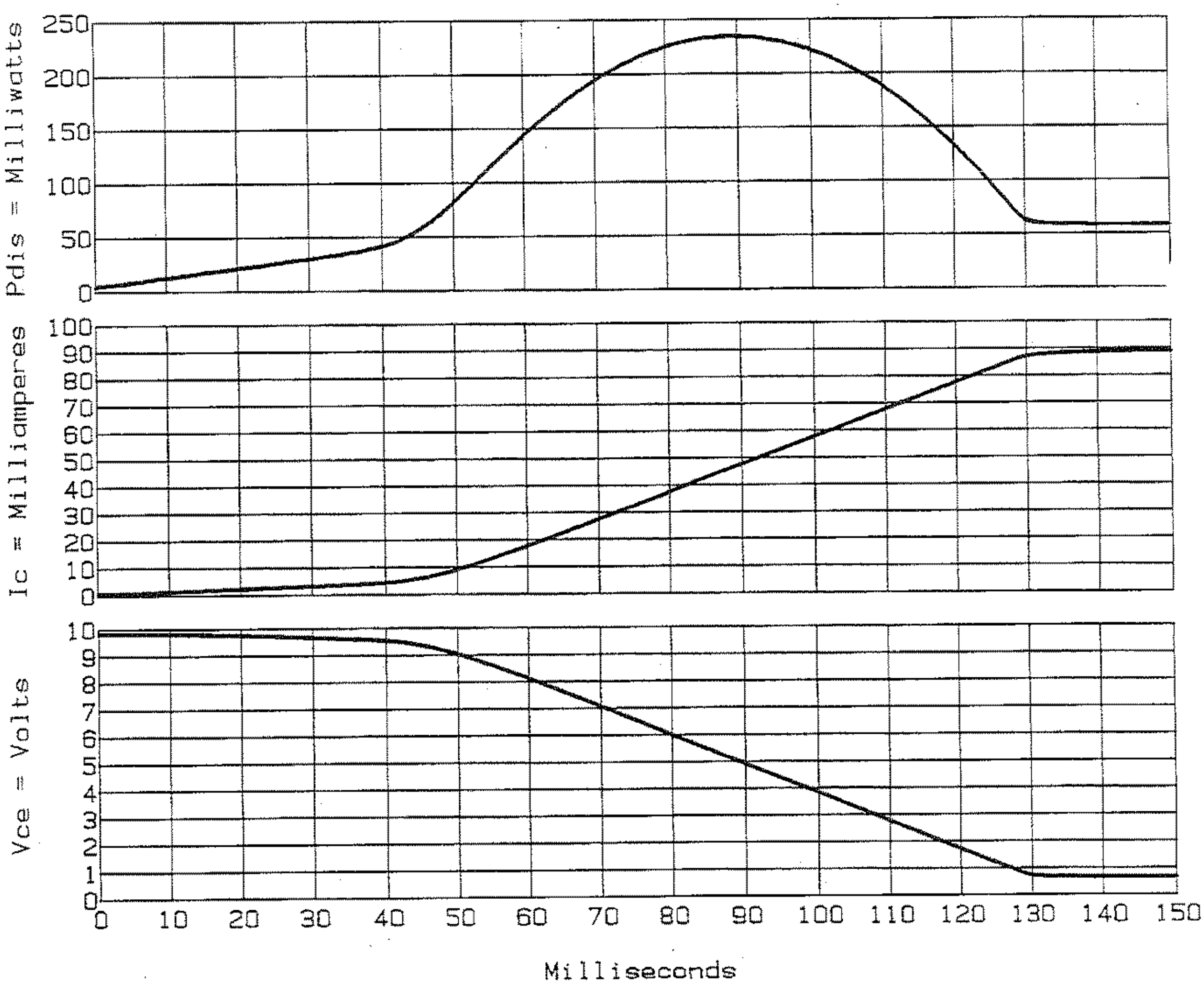



```
10   !This program takes a measurement of transistor voltage (Vce) versus time
20   !in Channel 1; and simultaneously, transistor current (Ic) versus time
30   !in Channel 2.  These two data channels are then read into array variables.
40   !The array variables are then multiplied together to give the power (Pdis)
50   !versus time that is dissipated by the transistor.  This resultant power
60   !(Pdis) is then plotted on the HP 7090, as well as the Vce and Ic.
70   !
80   !
90   !
100  !
110   ASSIGN @Hp7090 TO 705
120   OPTION BASE 1
130   DIM Chan1(1000),Chan2(1000),Result(1000)
140   OUTPUT @Hp7090;"IR10,.1;"   !Sets RANGE for Channels 1 and 2.
150   OUTPUT @Hp7090;"TB.15,0;"   !Sets TOTAL TIME to 0.15 second.
160   OUTPUT @Hp7090;"TA9,0;"     !Sets TRIGGER LEVEL to 9 volts.
170   OUTPUT @Hp7090;"TG-.05,0;"  !Sets PRETRIGGER TIME TO 0.05 second.
180   OUTPUT @Hp7090;"TM2,1;"     !Sets  TRIGGER MODE to INTERNAL,BELOW LEVEL.
190   OUTPUT @Hp7090;"MS1;"       !Invokes buffered recording mode.
200   DISP "CLOSE S1, THEN PRESS CONTINUE."
210   PAUSE
220   WAIT .5
230   GOSUB Data_out
240   GOSUB Result
250   GOSUB Plot_area1
260   GOSUB Grid_1and2
270   GOSUB Plot_1
280   GOSUB Plot_area2
290   GOSUB Grid_1and2
300   GOSUB Plot_2
310   GOSUB Plot_area3
320   GOSUB Grid_3
330   GOSUB Plot_res
340   GOSUB Annotate1
350   GOSUB Annotate2
360   GOSUB Annotate3
370   STOP
380   !
390   !
```

```
400    !
410    !
420    !
430    !
440 Data_out:     !This  subroutine takes data from the buffers
450               !and stores this data in array variables Chan1
460               !and Chan2.
470               !
480               OUTPUT @Hp7090;"DO1,1000,0,0;"
490               OUTPUT @Hp7090;"QI;"
500               FOR N=1 TO 1000
510               DISP "TRANSFERRING DATA"
520               ENTER @Hp7090 USING "#,K";Chan1(N)
530               NEXT N
540               OUTPUT @Hp7090;"DO2,1000,0,0;"
550               OUTPUT @Hp7090;"QI;"
560               FOR N=1 TO 1000
570               ENTER @Hp7090 USING "#,K";Chan2(N)
580               NEXT N
590               DISP "DATA TRANSFER COMPLETE"
600               RETURN
610               !
620 Result:       !This subroutine multiplies the data points
630               !in the Chan1 and Chan2 arrays together and
640               !places the results in the Result array.
650               !
660               FOR N=1 TO 1000
670               Result(N)=Chan1(N)*Chan2(N)
680               NEXT N
690               RETURN
700               !
710 Plot_area1:   !This subroutine sets the first plotting
720               !area at the lower one third of the page.
730               !
740               OUTPUT @Hp7090;"IP1000,1000,9000,3000;"
750               OUTPUT @Hp7090;"IZ1000,1000,9000,3000;"
760               RETURN
770               !
780 Grid_1and2:   !This subroutine grids the plotting area #1/#2.
790               !
800               OUTPUT @Hp7090;"SP2;"
810               OUTPUT @Hp7090;"GL15,10;"
820               OUTPUT @Hp7090;"DG;"
830               RETURN
840               !
850 Plot_1:       !This subroutine plots the data in Chan1 array.
860               !
870               OUTPUT @Hp7090;"IW1000,1000,9000,3000;"
880               OUTPUT @Hp7090;"SC0,1000,0,1000;"
890               OUTPUT @Hp7090;"SP1;"
900               FOR N=1 TO 1000
910               IF N=1 THEN OUTPUT @Hp7090;"PUPA";N;Chan1(N)*100
920               OUTPUT @Hp7090;"PDPA";N;Chan1(N)*100
930               NEXT N
940               OUTPUT @Hp7090;"SC;IW;"
950               RETURN
960               !
970 Plot_area2:   !This subroutine sets the second plotting
980               !area at the middle one third of the page.
990               !
1000              OUTPUT @Hp7090;"IP1000,3250,9000,5250;"
1010              OUTPUT @Hp7090;"IZ1000,3250,9000,5250;"
1020              RETURN
1030              !
```

```
1040 Plot_2:        !This subroutine plots the data in Chan2 array.
1050                !
1060                OUTPUT @Hp7090;"IW1000,3250,9000,5250;"
1070                OUTPUT @Hp7090;"SC0,1000,0,1000;"
1080                OUTPUT @Hp7090;"SP1;"
1090                FOR N=1 TO 1000
1100                IF N=1 THEN OUTPUT @Hp7090;"PUPA";N;Chan2(N)*10000
1110                OUTPUT @Hp7090;"PDPA";N;Chan2(N)*10000
1120                NEXT N
1130                OUTPUT @Hp7090;"SC;IW;"
1140                RETURN
1150                !
1160 Plot_area3:    !This subroutine sets the third plotting
1170                !area at the upper one third of the page.
1180                !
1190                OUTPUT @Hp7090;"IP1000,5500,9000,7500;"
1200                OUTPUT @Hp7090;"IZ1000,5500,9000,7500;"
1210                RETURN
1220                !
1230 Grid_3:        !This subroutine grids plotting area #3
1240                !
1250                OUTPUT @Hp7090;"SP2;"
1260                OUTPUT @Hp7090;"GL15,5;"
1270                OUTPUT @Hp7090;"DG;"
1280                RETURN
1290                !
1300 Plot_res:      !This subroutine plots the data in Result array.
1310                !
1320                OUTPUT @Hp7090;"IW1000,5500,9000,7500;"
1330                OUTPUT @Hp7090;"SC0,1000,0,25000;"
1340                OUTPUT @Hp7090;"SP1;"
1350                FOR N=1 TO 1000
1360                IF N=1 THEN OUTPUT @Hp7090;"PUPA";N;Result(N)*100000
1370                OUTPUT @Hp7090;"PDPA";N;Result(N)*100000
1380                NEXT N
1390                OUTPUT @Hp7090;"SC;IW;"
1400                RETURN
1410                !
1420 Annotate1:     !This subroutine Annotates area #1.
1430                !
1440                OUTPUT @Hp7090;"IP1000,1000,9000,3000;"
1450                OUTPUT @Hp7090;"SC0,150,0,10;"
1460                OUTPUT @Hp7090;"SP2;PUPA0,0;"
1470                OUTPUT @Hp7090;"LO4;SI.2,.3;"
1480                FOR X=0 TO 150 STEP 10
1490                OUTPUT @Hp7090 USING "K";"PA";X;",-1.5;LB";X;CHR$(3)
1500                NEXT X
1510                OUTPUT @Hp7090;"PA75,-3.5;LBMilliseconds";CHR$(3)
1520                OUTPUT @Hp7090;"PA0,0;LO18;"
1530                FOR Y=0 TO 10
1540                OUTPUT @Hp7090 USING "K";"PA-.5,";Y;";LB";Y;CHR$(3)
1550                NEXT Y
1560                OUTPUT @Hp7090;"LO4;PA-10,5;DI0,1;LBVce = Volts";CHR$(3)
1570                OUTPUT @Hp7090;"DI1,0;SC;LO;"
1580                RETURN
1590                !
1600 Annotate2:     !This subroutine annotates area #2.
1610                !
1620                OUTPUT @Hp7090;"IP1000,3250,9000,5250;"
1630                OUTPUT @Hp7090;"SC0,150,0,100;"
1640                OUTPUT @Hp7090;"PA0,0;LO18;"
1650                FOR Y=0 TO 100 STEP 10
1660                OUTPUT @Hp7090 USING "K";"PA-.5,";Y;";LB";Y;CHR$(3)
```

```
1670                    NEXT Y
1680                    OUTPUT @Hp7090;"LO4;PA-10,50;DI0,1;LBIc = Milliamperes";CHR$(3)
1690                    OUTPUT @Hp7090;"DI1,0;SC;LO;"
1700                    RETURN
1710                    !
1720 Annotate3:         !This subroutine annotates area #3.
1730                    !
1740                    OUTPUT @Hp7080;"IP1000,5500,9000,7500;"
1750                    OUTPUT @Hp7090;"SC0,150,0,250;"
1760                    OUTPUT @Hp7090;"PA0,0;LO18;"
1770                    FOR Y=0 TO 250 STEP 50
1780                    OUTPUT @Hp7090 USING "K";"PA-.5,";Y;";LB";Y;CHR$(3)
1790                    NEXT Y
1800                    OUTPUT @Hp7090;"LO4;PA-10,125;DI0,1;LBPdis = Milliwatts";CHR$(3)
1810                    OUTPUT @Hp7090;"DI1,0;SC;LO;SP0;"
1820                    RETURN
1830                    !
1840                    END
```



Notes

Chapter 5
Pen Control and Plotting

What You'll Learn in This Chapter

In this chapter, you will learn how to select or change pens, how to set and change pen velocity, how to raise and lower the pen, and how to plot. You will learn how to plot to absolute X,Y coordinates or to plot relative to the last pen position. You will also learn how to send variables as parameters of plot instructions; this will enable you to write general purpose graphics programs.

Instructions Covered

PA	The Plot Absolute Instruction
PR	The Plot Relative Instruction
PU/PD	The Pen Up/Down Instructions
SP	The Select Pen Instruction
VS	The Velocity Select Instruction

Terms You Should Understand

Absolute Plotting — plotting to a point whose location is specified relative to the reference coordinate system origin (0,0). When the PA instruction is used to plot to a point, the pen always moves to the same point on the plotting surface, no matter where the pen was before the move.

Relative Plotting — plotting to a point whose location is specified relative to the current pen position. The point moved to then becomes the effective origin for the next parameter of a plot relative instruction. When the PR instruction is used to plot to a point, the destination of the pen depends on where the pen was when the instruction was received.

Plotter Unit Equivalent — the X,Y coordinates of a point, given in user units, if they were expressed in plotter units.

The Select Pen Instruction, SP

USES: The SP instruction selects and/or stores a pen. The instruction is used to load a pen into the pen holder so that drawing will occur. It can be used during the plotting program to select pens of different colors or widths. It can be used with a zero parameter or no parameter to store the pen currently in the pen holder into its stall at the end of a program.

SYNTAX: *SP* pen number terminator
or
SP terminator

Parameter	Format	Range	Default
pen number	integer	0 to 6	0

EXPLANATION: The pen number parameter should be in the range 0 to 6. Decimal fractions are truncated. When the SP instruction is executed, the plotter lifts and stores its current pen, if any, before it retrieves the specified pen. After retrieving the new pen, it returns to the current graphics position.

An SP instruction with no parameter (SP;) or a parameter of zero (SP0;) stores the current pen in the stall from which it was retrieved.

Parameters less than 0 or greater than 6 are ignored. Parameters less than −32 768 or greater than 32 767 set error 3.

The Pen Instructions, PU and PD

USES: The PU and PD instructions raise and lower the pen. The instructions are used to raise and lower the pen during plotting. They may be used with parameters to plot or move to the points specified by the parameters.

SYNTAX: *PU* terminator
or
PD terminator
and
PU X, Y (,...) terminator
or
PD X, Y (,...) terminator

Parameter	Format	Range	Default
X,Y	integer	−32 768 to 32 767	none

EXPLANATION: When no parameters are included, the pen up instruction, PU, raises the pen without moving it to a new location. The pen down instruction, PD, lowers the pen without moving it to a new location, if the pen is within the soft-clip limits.

When parameters are included, the PU and PD instructions are interpreted as forms of the plot absolute instruction, PA, or the plot relative instruction, PR, depending on whether a PA or PR was the last plot instruction executed. If a PA or PR instruction has not been executed, PU and PD with parameters are interpreted as forms of the PA instruction.

If parameters are included, both coordinates of an X,Y coordinate pair must be given. An odd number of parameters will set an error condition, but all X,Y pairs that precede the unmatched parameter will be plotted. The coordinates are interpreted as plotter units if scaling is off and user units if scaling is on. Refer to the PA and PR instructions in this chapter and to the SC instruction in Chapter 6.

NOTE: The HP 7090 has an automatic pen lift feature which will lift the pen if it has remained motionless for approximately 65 seconds. ■

The Plot Absolute Instruction, PA

USES: The PA instruction moves the pen to the point(s) specified by the X- and Y-coordinate parameters. The instruction can be used together with PD to draw lines or with PU to move the pen to a specific point on the plot. It can also be executed without parameters to establish absolute plotting, as opposed to relative plotting for PU or PD instructions with parameters. In this case, the parameters of PU and PD are interpreted as absolute X,Y coordinates until any PR instruction is received.

SYNTAX: *PA* X_1 coordinate, Y_1 coordinate (, X_2 coordinate, Y_2 coordinate, ..., X_n coordinate, Y_n coordinate) terminator

or

PA terminator

Parameter	Format	Range	Default
X,Y coordinates	integer	−32 768 to 32 767	none

EXPLANATION: A PA instruction with parameters requires that both the X- and Y-coordinates are specified (coordinate pair) in integer format. The X-coordinate parameter specifies the X-axis location to which the pen will move. The Y-coordinate parameter specifies the Y-axis location to which the pen will move. If scaling is on, the coordinates are interpreted as user units; if scaling is off, the coordinates are interpreted as plotter units. Refer to The Scaling Instruction, SC, in Chapter 6 for a description of user units.

A PA instruction without parameters sets absolute plotting mode for PU and PD instructions with parameters.

The mnemonics PU and PD can be included ahead of, between, or after X,Y coordinate pairs. PU lifts the pen; PD lowers the pen.

Any number of coordinate pairs, as well as PU or PD mnemonics, can be listed after a PA instruction. (This is limited by the ability of the computer to continue instructions on multiple lines without generating a line feed character, which is an instruction terminator.) The pen will move to each point in the order given. Commas or spaces are required between numeric parameters and are optional after two-letter mnemonics. The last entry is followed by the terminator. In the following examples, commas are used to show optional and required separators. Optional commas or spaces which can be used between each letter of the mnemonics are not shown. The semicolon is used to indicate the terminator.

PA, PD, X_1, Y_1, PU, X_2, Y_2, PD, X_3, Y_3, ;

OPTIONAL

PU, X_1, Y_1, PD, X_2, Y_2, X_3, Y_3, PU;

OPTIONAL

PD, X_1, Y_1, X_2, Y_2, X_3, Y_3, ;

OPTIONAL

If no pen control parameter is given, the pen will assume the pen state (up or down) of the previous statement. The PU or PD mnemonics can also be substituted for the PA (or PR) mnemonic. This is equivalent to having PU; or PD; preceding the PA or PR instruction. Therefore, PU and PD with parameters are interpreted to be in place of PA or PR, depending upon which mnemonic, PA or PR, was last specified.

PA is specified by any of the following:

- power-up or front-panel initialization,
- execution of an IN instruction,
- execution of a DF instruction, or
- execution of a PA instruction with or without parameters.

The pen moves and draws lines only within the currently defined window. Refer to The Input Window Instruction, IW, in Chapter 6.

The plotter ignores parameters which are out of range. When scaling is off, in-range parameters are greater than or equal to −32 768 and less than or equal to 32 767. When scaling is on, both the parameters and their plotter unit equivalent must also be in that same range. To find the plotter unit equivalent, use the equations in the section Scaling Without Using the SC Instruction in Appendix C.

There are four types of vectors that can be drawn with a PA instruction from a given last point to some new point.

LAST POINT		NEW POINT
1. inside window area	to	inside window area
2. inside window area	to	outside window area
3. outside window area	to	inside window area
4. outside window area	to	outside window area

In type one, the pen moves from the last point to the new point with the pen up or down as programmed.

In type two, the pen moves from the last point toward the new point and stops where the line between the two points intersects the current window. The pen up/down condition is as programmed until the intersection is reached. Then, the pen is raised and OUT OF LMT (limit) is displayed.

In type three, the pen moves with the pen up, to the point where the straight line between the last and new point intersects the window limit. When the pen reaches this point, the **OUT OF LMT** goes out and the pen assumes its programmed (up or down) position. The pen then moves to the new point.

In type four, no pen movement occurs unless the straight line between the last and new point intersects the window. The X- and Y-coordinates of the current pen position are updated. If part of the vector is in the window area, the pen moves, pen up, to the point where the line between the last and the new point first intersects the window limit. The pen moves under programmed pen up/down control to the intersection of the vector and the other window limit. At this point, the pen stops and lifts.

Since out-of-range points are ignored, the plotter will draw a line between the two points on either side of discarded points. You can be sure all lines on your plot represent actual data if you:

1. have not changed the error mask from its default setting;

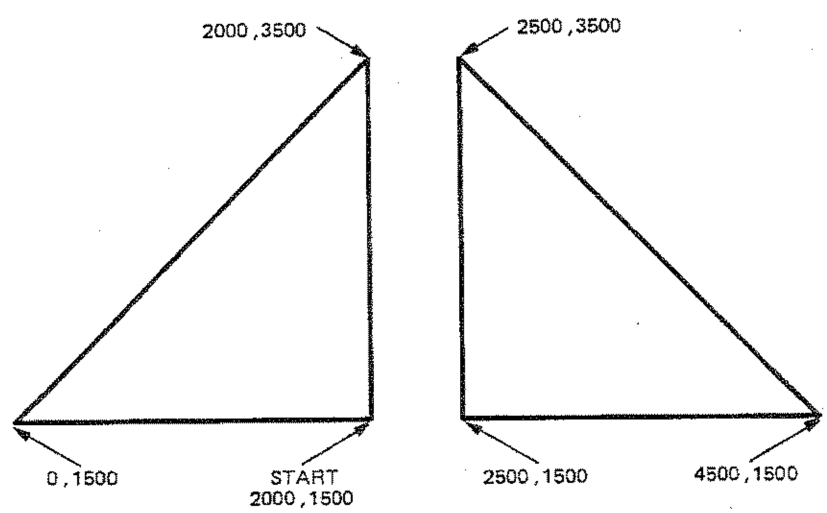
2. have not executed an output error instruction; and

3. the **ERROR** display is not on at the end of your plot.

(The fact that the **ERROR** display is on does not necessarily mean out-of-range data has been encountered; there are other situations that cause the **ERROR** display to turn on. For example, too many parameters in any instruction will turn on the **ERROR** display.)

The following strings of HP-GL instructions, if sent to the plotter using a suitable output statement such as PRINT or OUTPUT, will draw two triangles and then move to the point 10 000, 7000 with the pen up.

```
"IN;SP1;"
"PA2000,1500,PD,0,1500,2000,3500,2000,1500,
 PU,2500,1500;"
"PAPD4500,1500,2500,3500,2500,1500,PU,10000,7000;"
```



The Plot Relative Instruction, PR

USES: The PR instruction moves the pen relative to its current locatior by the number of units specified by the X- and Y-increment parameters. A PR instruction can be used like a PA instruction to draw lines and move to a point. However, with PR, pen movement is relative to the current pen position. The instruction can be executed without parameters to establish relative plotting as opposed to absolute plotting for PU or PD instructions with parameters. It is often used to draw multiple occurrences of some figure on a plot, for example, to draw several rectangles of the same size.

SYNTAX: *PR* X_1 increment , Y_1 increment (, X_2 increment ,
Y_2 increment , . . . , X_n increment , Y_n increment) terminator
or
PR terminator

Parameter	Format	Range	Default
X,Y increments	integer	−32 768 to 32 767	none

EXPLANATION: A PR instruction requires that both increments of an X,Y pair be given. An odd number of parameters will set an error condition, but all X,Y pairs which precede the unmatched parameter will be plotted.

The X-increment specifies the number of units the pen will move in the direction of the X-axis. The Y-increment specifies the number of units the pen will move in the direction of the Y-axis. The sign of the parameter determines the direction of movement; a positive value moves the pen ii the positive direction and a negative value moves the pen in the negative direction. If scaling is on, both parameters are interpreted as user units. If scaling is off, both parameters are interpreted as plotter units. Refer to The Scaling Instruction, SC, in Chapter 6 for a description of user units.

The mnemonics PU and PD can be included ahead of, between, or after X,Y increment pairs. PU lifts the pen; PD lowers the pen. Any number of increment pairs, as well as PU or PD mnemonics, (limited only by the ability of the computer to continue instructions on multiple lines without generating a line feed character) can be listed after the PR instruction. The placement of optional or required separators and the terminator is the same as for the PA instruction.

If no pen control parameter is given, the pen will assume the pen state (up or down) of the previous instruction. The PU or PD mnemonics can also be substituted for the PR mnemonic. This is equivalent to having PU ; or PD ; preceding the PR instruction. Since the power-on default is absolute plotting mode, a PR instruction must be executed before parameters of PD or PU instructions will be interpreted as X,Y increments. Relative plotting mode is cancelled by execution of a PA, IN, or DF instruction.

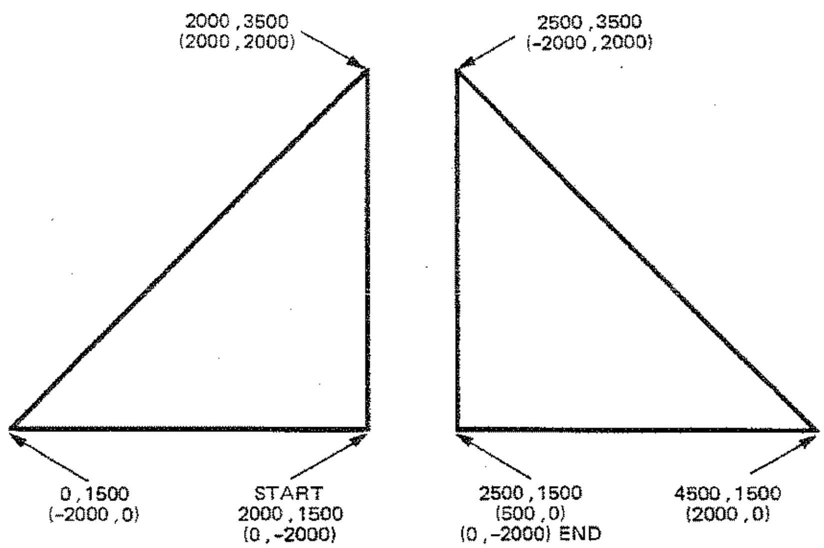
The pen moves and draws lines only within the currently defined window. Refer to The Input Window Instruction, IW, in Chapter 6. Drawing of vectors in relation to the window is as described under the PA instruction.

The plotter ignores parameters which are out of range or whose plotter unit equivalent would be out of range if the indicated move were made. Error is set (out-of-range parameter).

When scaling is off, in-range parameters are between −32 768 and 32 767. When scaling is on, in-range parameters and their plotter unit equivalent must be in the same range. To find plotter unit equivalents, refer to the section Scaling Without Using the SC Instruction in Appendix C.

The following strings of HP-GL instructions, when sent to the plotter using your computer's output statements, cause triangles to be drawn that are identical to the ones previously drawn using only the PA instruction. The numbers in parenthesis on the plot are the X,Y increments of the PR instructions. The numbers without parenthesis are the plotter unit coordinates of the vertices.

```
"IN;SP1;"
"PR2000,1500,PD,PR-2000,0,2000,2000,0,-2000,PU,500,0;"
"PD2000,0,-2000,2000,0,-2000,PU;"
```



Plotting with Variables

In many plotting applications, it is necessary to plot using variables rather than fixed numbers to define the X- and Y-coordinate values. The values of all HP-RL and HP-GL instruction parameters have the same restrictions (integer or decimals in a valid range) when sent as variables as when sent as literals (fixed numbers). The terminators and delimiters of HP-RL and HP-GL instructions must be sent to the plotter too. The method of defining output format and variable precision varies from computer to computer. Refer to your computer manual for the appropriate format statements that may be needed in your program.

The following BASIC program illustrates the use of variables in plotting a circle and shows how OUTPUT statements can be used to send variables as parameters of HP-GL instructions. You will use a similar method if you are programming in another language. Quotation marks are used by many computers and languages to delimit literal characters. Note the comma in line 70, which is part of the HP-GL instruction to be sent to the plotter; it is specified as a literal in quotes. With the HP 7090, a space may be substituted for the literal comma, shown in quotes. If your computer automatically sends spaces between variables, these spaces will delimit the coordinate parameters and a literal comma or space will not be necessary. The X and

Y variables are automatically truncated to integers by the plotter. Unless you are writing software to be compatible with other HP plotters such as the HP 9872, it is not necessary to add a formatting statement to assur variables are sent as integers by your computer.

To run this program, be sure that PI is a function recognized by your computer. If PI is not recognized, add a line before line 40 to define PI as a variable (PI = 3.1416).

```
10  OUTPUT 705;"IN;SP1;"
20  OUTPUT 705;"IP1000,1000,6000,6000;"
30  OUTPUT 705;"SC0,25000,0,25000;"
40  FOR T=0 TO 2*PI STEP PI/20
50  X=4.5*1000*COS(T)+12500
60  Y=4.5*1000*SIN(T)+12500
70  OUTPUT 705;"PA";X;",";Y;";PD;"
80  NEXT T
90  OUTPUT 705;"PU;SP0;"
100 END
```

This statement causes the plotter to move to the X, Y coordinate defined by the X- and Y- variables and lower the pen. The HP-GL mnemonics, delimiters and terminators are sent as literals in quotes on some computers. The delimiter "," is included here to delineate the variables X and Y.

The Velocity Select Instruction, VS

USES: The VS instruction specifies the pen-up and pen-down speed for plotting and labeling operations. Use it to slow velocity to 10 cm/s when plotting on transparency film. A slightly wider line can be created by slowing down the pen speed on any medium. A pen nearing the end of its life will write with a clearer, sharper, more solid line if the velocity is slowed.

SYNTAX: *VS* pen velocity terminator
or
VS terminator

Parameter	Format	Range	Default
pen velocity	real	0 to 127.9999 cm/s	75 cm/s

EXPLANATION: A VS instruction without parameters (VS ;) sets pen-up and pen-down speed to the default velocity of 75 cm/s (30 in./s). Parameters equal to or greater than 38 cm/s set both pen-up and pen-down speeds to the specified value. Parameters less than 38 cm/s set the pen-up speed to 38 cm/s and the pen-down speed to the specified value.

Any parameter in the range of 0 to 127.9999 may be specified. However, a velocity of 0 to 0.97 is set to 0.97 cm/s. Parameters greater that 0.97 cm/s are set to the nearest multiple of 1.94 cm/s; i.e., a parameter of 3 is set to 2.91 cm/s (0.97 + 1.94). Negative parameters and parameters greater than or equal to 128 set error 3, and the instruction is ignored.

A DF or IN instruction will reset pen velocity to 75 cm/s.

Notes

Chapter 6

Plotter Scaling and Windows

What You'll Learn in This Chapter

In this chapter you will learn how to scale the plotting area into user units appropriate for your data, and how to set or read the current scaling points. After reading this chapter, you will also be able to restrict plotting to only a portion of the plotting area and read the current soft-clip limits.

Instructions Covered

IP The Input P1 and P2 Instruction
IW The Input Window Instruction
OP The Output P1 and P2 Instruction
OW The Output Window Instruction
SC The Scale Instruction

Terms You Should Understand

Scaling — dividing the plotting area into units convenient for your application. Units need not be the same physical size in both axes, nor do there need to be an equal number of units in the X- and Y-axes.

Scaling Points — the points on the plotting surface moved to when the front-panel buttons P1 and P2 are pressed. These points are assigned the user-unit values specified by the parameters of the scaling instruction SC.

Window — that part of the plotting area in which plotting of points, lines and labels can occur. At power-on, the window is set inside the hard-clip limits. Only recorder plots, grids, and labels from the front-panel can be drawn outside the current window.

Clipping — restricting plotting to a portion of the plotting area by establishing a window of a certain size.

Setting the Scaling Points

On power-up, the default location of scaling point P1 is in the lower-left corner of A/A4 size paper or in the lower-right corner of B/A3 size paper. In each case, the default location of scaling point P2 is in the corner opposite from P1. The exact default coordinate locations of scaling points P1 and P2 are shown in the tables given in The Reference Coordinate System paragraph, Chapter 2. These default coordinate values define opposite corners of a rectangular area that is approximately centered on the associated size of paper. Regardless of its size, the rectangular area defined by P1 and P2 will hereafter be referred to as the "P1/P2 frame."

The locations of scaling points P1 and P2 can be changed manually from the front panel or programmatically with the IP instruction. Refer to the following paragraph for the manual procedure and to the following section for a description of the IP instruction. The default locations for P1 and P2 can be reestablished by any of the following methods:

- power-up initialization,
- execution of either the IN instruction or the IP instruction without parameters,
- pressing SHIFT+STOP buttons (front-panel initialization).

Setting P1 and P2 Manually

P2 moves when P1 is moved manually. If you want P2 to be at a specific location, set P1 first and then P2. If you want to establish an area of a certain size onto which the parameters of a scale instruction will be mapped, you may set P2 in the desired location relative to the current P1, and then move P1. P2 will move with P1 so that both the X- and Y-distances between P1 and P2 remain constant. If such a move means the new location of P2 will be beyond the hard-clip limits, either or both coordinates of P2 are set to the hard-clip limits. In this case, the size of the rectangle established by P1 and P2 will, of course, not remain the same.

To set P1 and P2 manually:

1. Press the front-panel SHIFT+P1 buttons to invoke the P1 positioning function and turn on the associated indicator. Move the pen to the desired location using the front-panel pen cursor buttons.
2. Press the P1 button a second-time. This extinguishes the indicator and sets the new location of P1.
3. Press the front-panel SHIFT+P2 buttons to invoke the P2 positioning function and turn on the associated indicator. Move the pen to the desired location using the front-panel pen cursor buttons.
4. Press the P2 button a second-time. This extinguishes the indicator and sets the new location of P2.
5. Check the new locations of the scaling points by pressing SHIFT+P1; then SHIFT+P2. The pen will move to the new P1 and P2 points.

NOTE: While either indicator is lit, moving the pen with the pen cursor buttons will change the location of the associated scaling point. Always exit the positioning functions to avoid accidental repositioning of P1 or P2. ■

The Input P1 and P2 Instruction, IP

USES: The IP instruction provides program control over the locations of the P1 and P2 scaling points. The IP instruction can be used to move P1 and P2 from their default or current locations; to give mirror images of vectors and labels; to change the size of a user unit, thus reducing or enlarging an image; to change the size or direction of labels when relative character size or direction is in effect; and to set P1 and P2 back to their default locations.

SYNTAX: *IP* $P1_X, P1_Y(, P2_X, P2_Y)$ terminator
or
IP terminator

Parameter	Format	Range	Default
P1, P2	integer	−32 768 to 32 767	dependent on current paper size

EXPLANATION: The coordinates of P1 and P2 are specified in the order shown above and are interpreted as plotter units. Neither point can be located outside of the current hard-clip limits. If the location of either point is specified outside of the hard-clip limits, its coordinates are changed automatically to the hard-clip limits and no error is set. Parameters less than −32 768 or greater than 32 767 will cause error 3 and the instruction is ignored.

Specifying the coordinates of P2 is optional. However, if the P2 coordinates are omitted, then P2 tracks P1 and its coordinates change so that the X-and Y-distances between P1 and P2 do not change, unless P2 intersects the hard-clip limits.

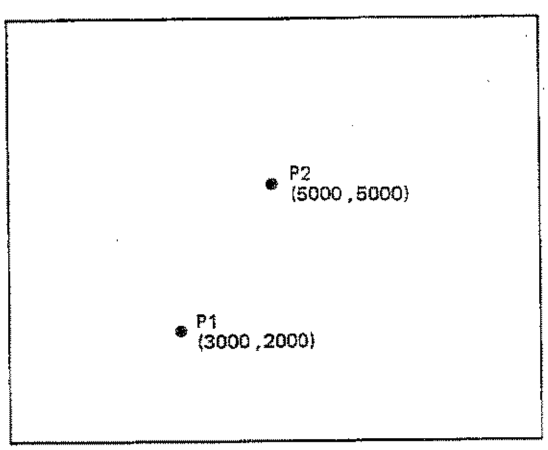
If either coordinate of P2 is set equal to the corresponding coordinate of P1, then that coordinate of P2 is incremented by one plotter unit.

An IP instruction without parameters (IP;) sets P1 and P2 to the default coordinate values for the currently selected size of paper. Refer to The Reference Coordinate System, Chapter 2, for the default coordinate values of P1 and P2.

Upon receipt of a valid IP instruction, bit position 1 of the plotter status byte is set true (1). Refer to The Output Plotter Status Instruction, OS, Chapter 12.

The following IP instruction relocates the scaling points P1 and P2 to the positions shown in the figure.

```
"IP3000,2000,5000,5000;"
```



The Output P1 and P2 Instruction, OP

USES: The OP instruction outputs the current plotter unit coordinates of P1 and P2. This information can be used with the IW instruction to set the soft-clip limits to P1 and P2, to compute the number of plotter units per user unit when plotter scaling is on, or to determine the numeric coordinates of P1 and P2 when they have been set manually.

SYNTAX: *OP* terminator

PARAMETERS: None

EXPLANATION: When an OP instruction is received, the current coordinates of P1 and P2 are output in plotter units as four numbers in ASCII, separated by commas.

$P1_X, P1_Y, P2_X, P2_Y$ CRLF

Upon completion of output, bit position 1 of the plotter status byte is cleared (set to 0).

The Scale Instruction, SC

USES: The SC instruction establishes a user-unit coordinate system by mapping values onto the scaling points P1 and P2. This instruction is used to enable you to plot in user units convenient to your application. Scaling can be anisotropic (unequal in X and Y) or isotropic (equal in X and Y). In addition, you can create mirror images of the plot and enlarge or reduce the plot size by changing the positions of P1 and P2.

SYNTAX: *SC* $X_{min}, X_{max}, Y_{min}, Y_{max}$ terminator
or
SC terminator

Parameter	Format	Range	Default
X,Y	integer	−32 768 to 32 767	plotter units

EXPLANATION: Executing an SC instruction without parameters (SC;) turns scaling off and subsequent parameters of plot instructions are interpreted as plotter units.

When parameters are used, all four parameters are required. Decimal parameters in an SC instruction are truncated to integers. The parameters X_{min} and Y_{min} define the user-unit coordinates of P1; the parameters X_{max} and Y_{max} define the user-unit coordinates of P2. P1 and P2 may be any two opposite corners of a rectangle. No matter where P1 and P2 are, the parameters of the SC instruction are always mapped onto the current P1 and P2 location. Scaling points P1 and P2 retain the assigned user-unit coordinate values until scaling is turned off or another SC instruction redefines their user-unit coordinate values. Therefore, the physical size of a user unit will change when any change is made in the relative position and distance between P1 and P2. Once an SC instruction has been executed, all parameters of HP-GL instructions PA, PR, and the response to an OC instruction are interpreted as user units.

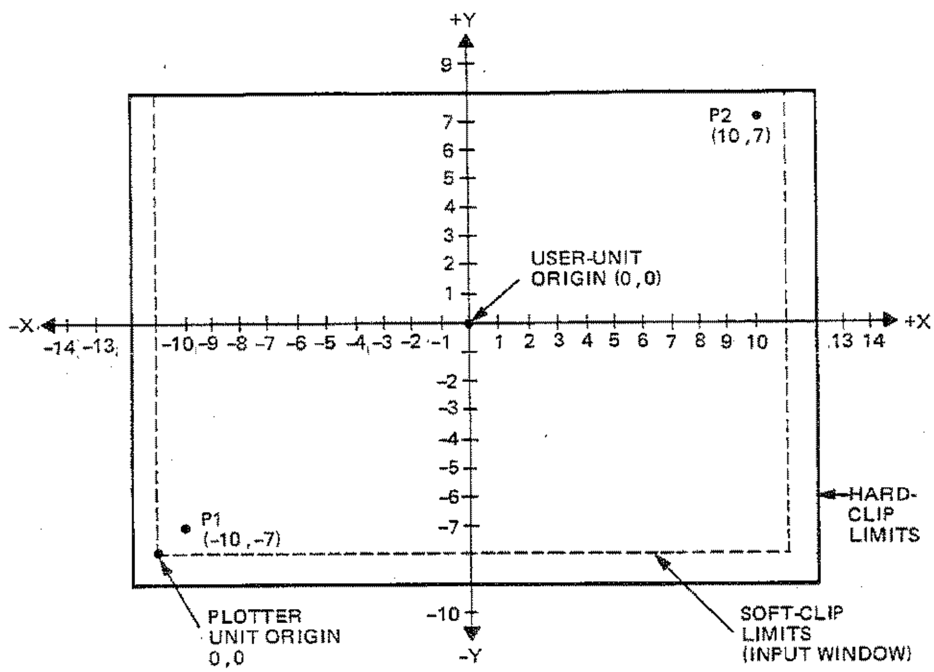
The user-unit coordinate system that is mapped onto the reference coordinate system by the SC instruction is not limited to the rectangle defined by P1 and P2; it extends over the entire range of the reference coordinate system.

It is not possible to scale an area such that P1 and P2 are assigned values larger than 32 767 or less than −32 768. To plot data with values beyond these limits, reduce your data to acceptable ranges by an arithmetic process before sending it to the plotter. This can be accomplished by dividing the data by some factor of 10 so that the integer portions fall between ±32 767.

Specifying $X_{max} = X_{min}$ or $Y_{max} = Y_{min}$ or parameters less than −32 768 or greater than 32 767 will set error 3, and the instruction is ignored. If more than four parameters are specified, the instruction is executed with the first four parameters, error 2 is set, and the rest of the parameters are ignored.

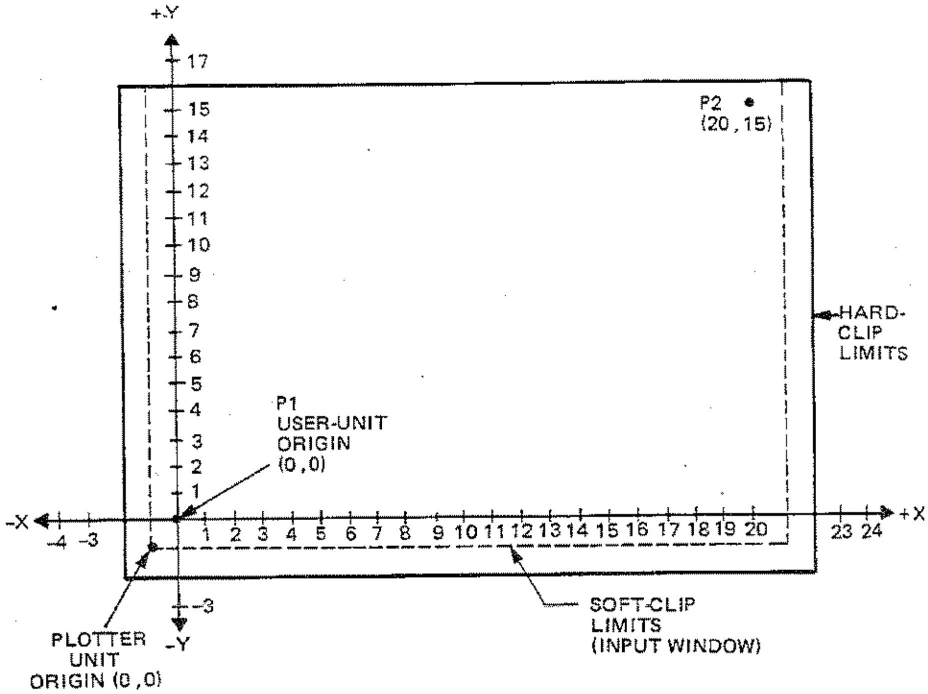
The following illustration shows the user-unit grid that is mapped onto the reference coordinate system as a result of executing the indicated SC instruction. P1 and P2 are shown at their default locations for A/A4 size paper, but the scaling would be the same for any P1/P2 frame.

```
"SC-10,10,-7,7;"
```



In the next illustration, P1 and P2 are still set at their default locations, but the indicated SC instruction places the user-unit origin at P1.

"SC0,20,0,15;"



One factor to consider when determining scaling parameters is the required smoothness of the plot. The plotter can only move to integer locations in the plotter area. The smallest addressable move is one plotter unit or one user unit in the X- or Y-direction. This is the resolution of the plot. Resolution is limited in that 0.025 mm (0.000 98 in.) is the smallest move the plotter can make. This is true even when finer resolution is specified by the parameters of the SC instruction and the positions of P1 and P2. When resolution finer than 0.025 mm is specified, the plotter uses conventional rounding and positions the pen as close as possible to the specified coordinates in increments of 0.025 mm.

When the range of the scale parameters is small, resolution is coarse and may be insufficient to produce a satisfactory plot. While scaling could establish a user unit as large as the hard-clip limits, the plotting area is usually divided into several hundred or several thousand units. The limitation on coarse resolution is determined by the minimum number of user units that will allow creation of a satisfactory plot.

A second factor to consider is whether the plots represent geometric or actual shapes. If so, isotropic scaling (equal in X and Y) must be established. The means to assure sufficient resolution and set isotropic scaling are demonstrated in the following examples.

The examples are based on the equations for a circle in polar coordinate form.

$$X = R\cos T \quad \text{and} \quad Y = R\sin T$$

where R is the radius of the circle and T is the angle in radians. Since there are π radians in 180 degrees, T must vary from 0 to 2π to draw a circle.

Example of Insufficient Resolution

This example shows that an indiscernible figure may result if user units are too large. More often the figure will be recognizable but the curves will not be sufficiently smooth. The above equations are used to draw a circle in the center of the plotting area. The resolution of the plot as established by the SC instruction is as follows: X-axis resolution = 5 mm since there are 5000 plotter units in the area divided into 25 user units in X.

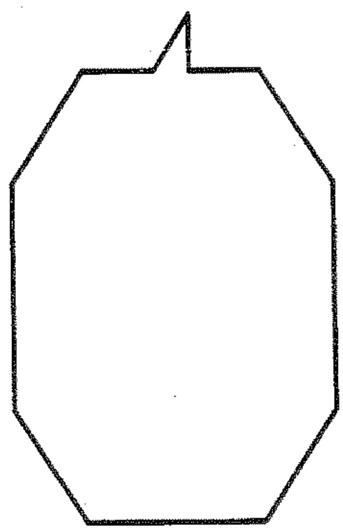
$$5000 \times \frac{0.025\text{ mm}}{25} = 5\text{ mm}$$

For the Y-axis resolution

$$5000 \times \frac{0.025\text{ mm}}{16} = 7.8\text{ mm}$$

This is 200 times coarser than plotter-unit resolution.

```
10     ASSIGN @Hp7090 TO 705
20     OUTPUT @Hp7090;"IN;SP1;IP1000,1000,6000,6000;"
30     OUTPUT @Hp7090;"SC0,25,0,16;"
40     FOR T=0 TO 2*PI STEP PI/20
50     X=4.5*COS(T)+12.5
60     Y=4.5*SIN(T)+8
70     OUTPUT @Hp7090 USING 80;"PA";X;Y;";PD;"
80     IMAGE 2A,2(MDD),4A
90     NEXT T
100    OUTPUT @Hp7090;"PU;"
110    END
```



Indiscernible Circle

Program Description

Line 20 initializes the plotter, selects pen number 1, and positions P1 and P2.

Line 30 scales the X-axis into 25 user units and the Y-axis into 16 user units.

Lines 40 and 90 define a loop in which lines 50 through 80 are executed 40 times as T ranges from 0 to 2π radians, in steps of $\pi/20$ radians.

Line 50 computes the X-coordinate of a point on a circle with a radius of 4.5 user units, centered horizontally in the scaled area at 12.5.

Line 60 computes the Y-coordinate of a point on the circle and centers the circle vertically at 8.

Line 70 plots the X- and Y-coordinates computed in lines 50 and 60. Line 80 ensures that the values are rounded to integers before they are sent to the plotter.

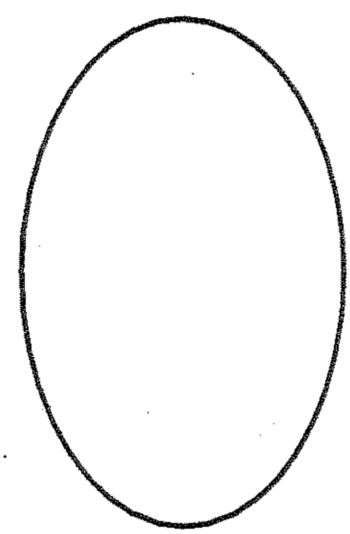
Line 100 raises the pen.

Example of Distortion

This example solves the resolution problem by using a multiplier of 1000 in both the scaling parameters and the equations for X and Y. The multiplier is also used in the offset values to center the plot.

Notice that the plot is an ellipse, rather than the expected circle. This is a result of the unequal scaling in X and Y established by the parameters of the SC instruction.

```
10      ASSIGN @Hp7090 TO 705
20      OUTPUT @Hp7090;"IN;SP1;IP1000,1000,6000,6000;"
30      OUTPUT @Hp7090;"SC0,25000,0,16000;"
40      FOR T=0 TO 2*PI STEP PI/20
50      X=4.5*1000*COS(T)+12500
60      Y=4.5*1000*SIN(T)+8000
70      OUTPUT @Hp7090 USING 80;"PA";X;Y;";PD;"
80      IMAGE 2A,2(MDDDDD),4A
90      NEXT T
100     OUTPUT @Hp7090;"PU;"
110     END
```

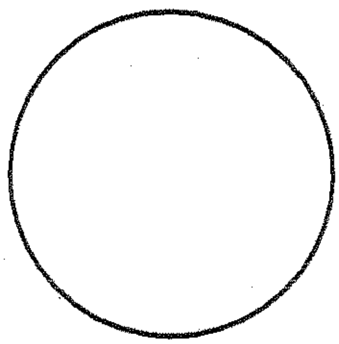


Distorted Circle

Examples of Equal Scaling

When P1 and P2 define a square area as they do here, the way to compensate for the distortion shown in the prior example is to set an equal number of user units in both X and Y. The plot will be a perfect circle.

```
10    ASSIGN @Hp7090 TO 705
20    OUTPUT @Hp7090;"IN;SP1;IP1000,1000,6000,6000;"
30    OUTPUT @Hp7090;"SC0,25000,0,25000;"
40    FOR T=0 TO 2*PI STEP PI/20
50    X=4.5*1000*COS(T)+12500
60    Y=4.5*1000*SIN(T)+12500
70    OUTPUT @Hp7090 USING 80;"PA";X;Y;";PD;"
80    IMAGE 2A,2(MDDDDD),4A
90    NEXT T
100   OUTPUT @Hp7090;"PU;"
110   END
```



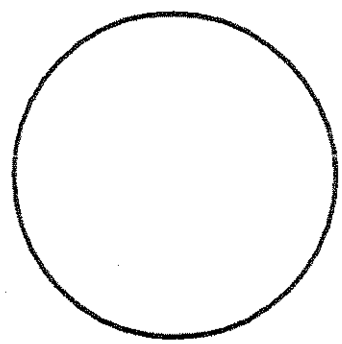
Perfect Circle

When the area defined by P1 and P2 is not square and the scaling parameters do not establish isotropic scaling, a geometrically accurate plot can be created by using multipliers in the equations for X and Y or by adjusting the parameters of the SC instruction. This is frequently done in programs which may run on a variety of graphics devices or paper sizes. The constant would be calculated elsewhere in the program, once the device to be used for that plot was determined. The multiplier would then appear in lines 50 or 60 as a variable instead of the constant 800 that appears in line 60 of the second program and the SC instruction in line 30 of the second program would depend on that constant.

The same circle will be drawn by both the following programs in the same area as the ellipse drawn earlier. Note the area defined by P1 and P2 is not square. The first program adjusts the SC parameters to match the IP instruction and one scaled unit equals one plotter unit. One scaled unit could equal any number of plotter units. The second program adjusts the multiplier in the equation for Y to correspond to the ratio $(P2_X - P1_X)/(P2_Y - P1_Y)$.

```
10     ASSIGN @Hp7090 TO 705
20     OUTPUT @Hp7090;"IN;SP1;IP1000,2875,6000,4125;"
30     OUTPUT @Hp7090;"SC0,5000,0,1250;"
40     FOR T=0 TO 2*PI STEP PI/20
50     X=4.5*200*COS(T)+2500
60     Y=4.5*200*SIN(T)+625
70     OUTPUT @Hp7090 USING 80;"PA";X;Y;";PD;"
80     IMAGE 2A,2(MDDDD),4A
90     NEXT T
100    OUTPUT @Hp7090;"PU;"
110    END
```

```
10     ASSIGN @Hp7090 TO 705
20     OUTPUT @Hp7090;"IN;SP1;IP1000,2875,6000,4125;"
30     OUTPUT @Hp7090;"SC0,5000,0,5000;"
40     FOR T=0 TO 2*PI STEP PI/20
50     X=4.5*200*COS(T)+2500
60     Y=4.5*800*SIN(T)+2500
70     OUTPUT @Hp7090 USING 80;"PA";X;Y;";PD;"
80     IMAGE 2A,2(MDDDD),4A
90     NEXT T
100    OUTPUT @Hp7090;"PU;"
110    END
```

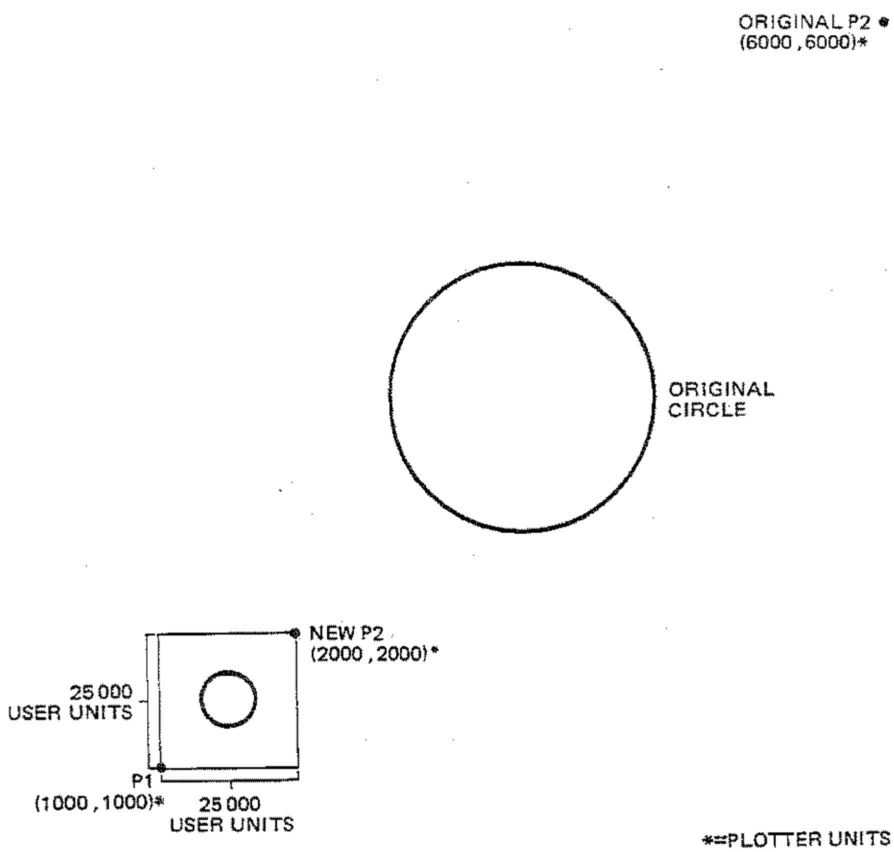


Scaling can also be used to expand or contract a graphics plot. To do this, the locations of P1 and P2 are changed.

Example of a Contracted Plot

The following listing retraces the perfect circle plotted in example 3, then relocates P2 closer to P1 and generates a second circle. Notice that the second circle is much smaller, but is centered within the rectangle defined by P1 and the new P2 location.

```
10    ASSIGN @Hp7090 TO 705
20    OUTPUT @Hp7090;"IN;SP1;IP1000,1000,6000,6000;"
30    OUTPUT @Hp7090;"SC0,25000,0,25000;"
40    GOSUB 80
50    OUTPUT @Hp7090;"IP1000,1000,2000,2000;"
60    GOSUB 80
70    STOP
80    FOR T=0 TO 2*PI STEP PI/20
90    X=4.5*1000*COS(T)+12500
100   Y=4.5*1000*SIN(T)+12500
110   OUTPUT @Hp7090 USING 120;"PA";X;Y;";PD;"
120   IMAGE 2A,2(MDDDDD),4A
130   NEXT T
140   OUTPUT @Hp7090;"PU;"
150   RETURN
160   END
```



A third phenomena which can be accomplished using scaling is plotting of mirror images.

Example of Mirror Images

Mirror images of normal vector plots can be produced by changing the locations of scaling points P1 and P2 to one of their unusual orientations after the SC instruction has been executed. The following listing illustrates how mirror images of vectors are produced. The resultant plot is shown in four stages to track the sequence of the program listing.

```
10     ASSIGN @Hp7090 TO 705
20     OUTPUT @Hp7090;"IN;SP1;IP5000,3600,7500,6100;"
30     OUTPUT @Hp7090;"SC-15,15,-10,10;"
40     GOSUB 120
50     OUTPUT @Hp7090;"IP5000,3600,2500,6100;"
60     GOSUB 120
70     OUTPUT @Hp7090;"IP5000,3600,2500,1100;"
80     GOSUB 120
90     OUTPUT @Hp7090;"IP5000,3600,7500,1100;"
100    GOSUB 120
110    STOP
120    OUTPUT @Hp7090;"PA1,2,PD,1,4,3,4,3,7,2,7;"
130    OUTPUT @Hp7090;"PA4,9,6,7,5,7,5,4;"
140    OUTPUT @Hp7090;"PA12,4,12,5,14,3;"
150    OUTPUT @Hp7090;"PA12,1,12,2,1,2,PU;"
160    RETURN
170    END
```

Program Description

Line 20 initializes the plotter, selects pen number 1, and positions P1 and P2 in their normal orientation.

Line 30 scales the plot into user units

Lines 40 and 120 through 150 produce a normal plot of the arrow.

Line 50 relocates P2 above and to the left of P1.

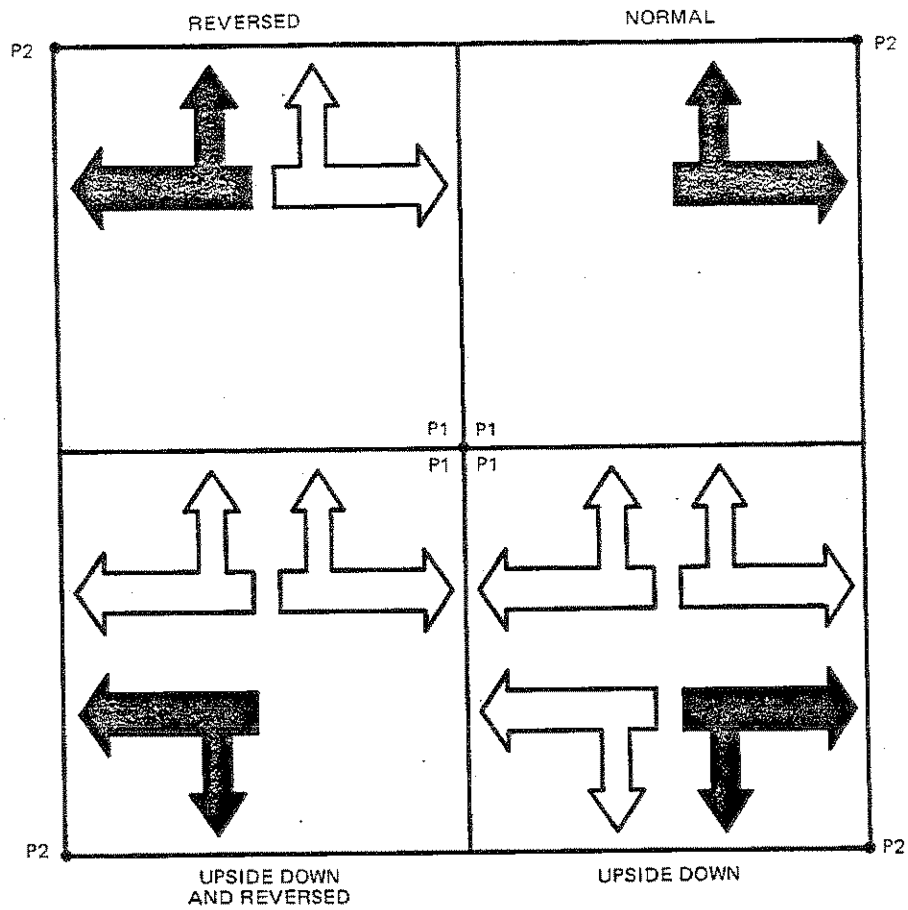
Lines 60 and 120 through 150 draw a second arrow which is a reversed mirror image of the normal plot.

Line 70 relocates P2 below and to the left of P1.

Lines 80 and 120 through 150 draw a third arrow which is an upside down and reversed mirror image of the normal plot.

Line 90 relocates P2 below and to the right of P1.

Lines 100 and 120 through 150 draw a fourth arrow which is an upside down mirror image of the normal plot.



The Input Window Instruction, IW

USES: The IW instruction is used to restrict programmed pen motion to a specific rectangular area. This area is called the window. Use the instruction to establish a soft-clip area, i.e., restrict plotting to a certain area of the paper. The instruction is especially useful when your data should fall in a certain range but your scaling is larger (perhaps you have left room for labels) and you don't want lines outside the normal data area. It is also useful when hatching (shading rectangular areas). The window has no effect on direct or buffered recording traces or front-panel initiated labels.

SYNTAX: *IW* $X_{lower\ left}, Y_{lower\ left}, X_{upper\ right}, Y_{upper\ right}$ terminator
or
IW terminator

Parameter	Format	Range	Default
X,Y	integer	−32 768 to 32 767	dependent on current paper size

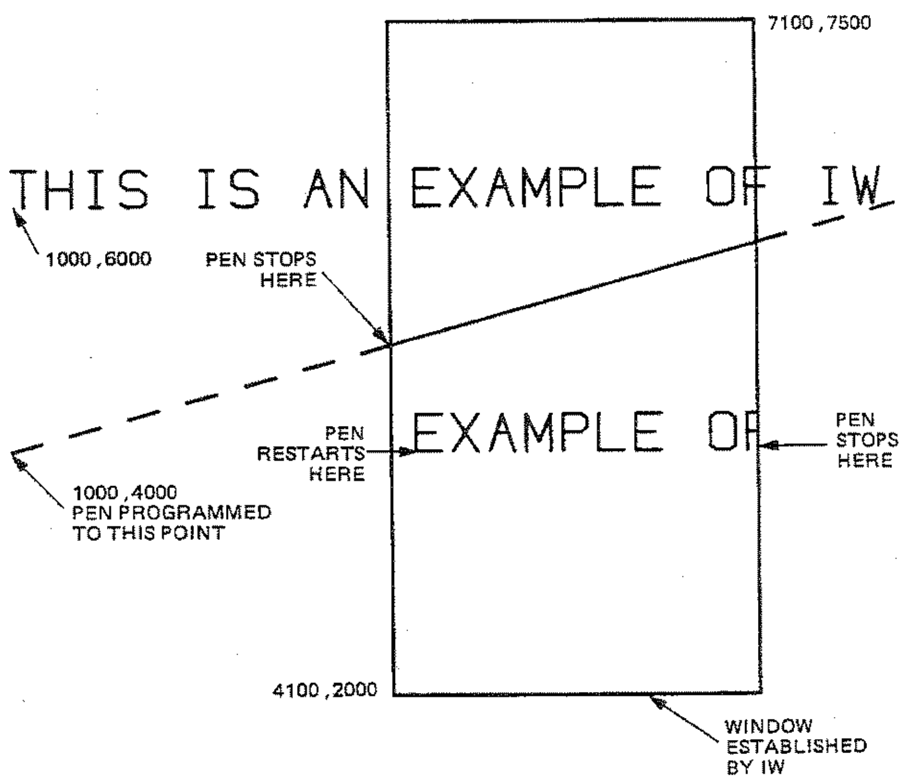
EXPLANATION: When parameters are used, all four parameters are required and are interpreted as plotter units. The parameters specify the X- and Y-coordinates of the lower-left and upper-right corners of the window area. If the location of either corner is specified outside of the hard-clip limits, its coordinates are changed automatically to the hard-clip limits and no error is set. Parameters less than −32 768 or greater than 32 767 will cause error 3 and the instruction is ignored.

If X- or Y-coordinates of the lower-left corner are specified to be greater than the X- or Y-coordinates of the upper-right corner, the coordinates will be automatically interchanged. For example, IW 6000, 3000, 5000, 4000 will be converted to IW 5000, 3000, 6000, 4000.

An IW instruction without parameters (IW;) sets the window area to the default coordinate values for the currently selected size of paper. Refer to The Reference Coordinate System, Chapter 2, for the default coordinate values of the window area.

The following example demonstrates the effects of specifying a window on labels and vectors.

```
10    ASSIGN @Hp7090 TO 705
20    OUTPUT @Hp7090;"IN;SP1;SI.5,.75;"
30    OUTPUT @Hp7090;"PA1000,6000;"
40    OUTPUT @Hp7090;"LBTHIS IS AN EXAMPLE OF IW␃"
50    OUTPUT @Hp7090;"IW4100,2000,7100,7500;"
60    OUTPUT @Hp7090;"PD1000,4000PU;"
70    OUTPUT @Hp7090;"LBTHIS IS AN EXAMPLE OF IW␃"
80    END
```



Program Description

Line 20 initializes the plotter, selects pen number 1, and sets the character size.

Line 30 moves the pen to the position where the label will begin and line 40 sends the label string.

Line 50 establishes a new window to limit the plotting area.

Line 60 moves the pen below the starting point of the first label and line 70 sends the label string again. Note that the vector and the second label string are clipped and are only plotted within the new window area.

The Output Window Instruction, OW

USES: The OW instruction is used to output the X- and Y-coordinates of the lower-left and upper-right corners of the window area in which plotting and I/O labeling can currently occur.

SYNTAX: *OW* terminator

PARAMETERS: None

EXPLANATION: When an OW instruction is received, the plotter will output the coordinates of opposite corners of the window area in plotter units as four numbers in ASCII, separated by commas.

$X_{lower\ left}, Y_{lower\ left}, X_{upper\ right}, Y_{upper\ right}$ CRLF

The range of the coordinates is limited to the hard-clip limits of the currently selected size of paper.

Chapter 7
Plot Enhancement

What You'll Learn in This Chapter

In this chapter you will learn how to enhance your plots by using HP-GL instructions to draw tick marks on axes or create grids, draw a symbol or character of your choice at each data point, and draw dashed or dotted lines. All these enhancements will make your data easier to interpret.

Instructions Covered

LT The Line Type Instruction
SM The Symbol Mode Instruction
TL The Tick Length Instruction
XT The X-Tick Instruction
YT The Y-Tick Instruction

The Tick Instructions, XT and YT

USES: The XT instruction draws a vertical X-tick at the current pen location. The YT instruction draws a horizontal Y-tick at the current pen location. These instructions can be used to draw tick marks on axes, draw grid lines by making the tick length 100%, or draw horizontal or vertical lines either centered on or ending at the current pen position.

SYNTAX: *XT* terminator
or
YT terminator

PARAMETERS: None

EXPLANATION: These instructions include an automatic pen down feature, and draw a tick at the current pen position. After the tick is drawn, the pen assumes the pen state (up or down) that was in effect prior to the tick instruction.

The tick length is specified by the tick length instruction, TL. If no tick length is specified, the length defaults to 0.5% of ($P2_X - P1_X$) for YT or 0.5% of ($P2_Y - P1_Y$) for XT. This percentage applies to both the positive and negative portion of the tick. Refer to The Tick Length Instruction, TL, which follows.

The following example draws a horizontal line 3000 plotter units long, places X-ticks at the endpoints and at X-locations 1200 and 2200, and raises and stores the pen.

```
"IN;SP2;PR200,500;PD;XT;PR1000,0;XT;"
"PR1000,0;XT;PR1000,0;XT;PU;SP0;"
```

The Tick Length Instruction, TL

USES: The TL instruction specifies the length of the tick marks drawn by the plotter. The tick lengths are specified as a percentage of the horizontal and vertical distances between the scaling points P1 and P2. The instruction can be used to set the length of both positive and negative portions of tick marks. The instruction can be used with only one parameter to suppress the negative portion of a tick mark, or with a first parameter of zero to suppress the positive portion of the tick. Setting the tick length, tp, to 100 enables the user to draw grids easily, using the XT and YT instructions.

SYNTAX: *TL* tp (, tn) terminator
or
TL terminator

Parameter	Format	Range	Default
tp, tn	real	−128.0000 to 127.9999	tp and tn = 0.5% of ($P2_Y - P1_Y$) for XT and 0.5% of ($P2_X - P1_X$) for YT

EXPLANATION: Both parameters must be between −128.0000 and 127.9999. Use of positive parameters is recommended. For most applications, parameters will be between 0 and 100.

The up and right tick length, tp, determines the length of the upward portion of the tick marks drawn along the X-axis and the right-side portion of the tick marks drawn along the Y-axis, taking P1 as the lower-left corner.

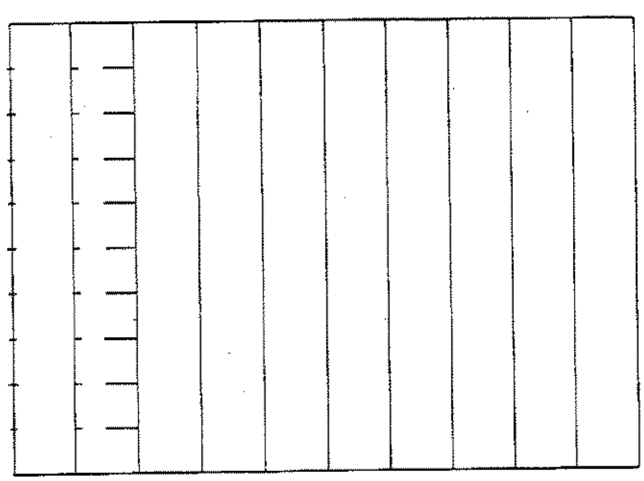
The down and left tick length, tn, determines the length of the downward portion of the tick marks drawn along the X-axis and the left-side portion of the tick marks drawn along the Y-axis, taking P1 as the lower-left corner.

The values specified by parameters tp and tn are a percentage of the vertical scale length ($P2_Y - P1_Y$) when used with the XT instruction, and a percentage of the horizontal scale length ($P2_X - P1_X$) when used with the YT instruction. Note the actual tick length is a function of the scaling established by P1 and P2, and the length of ticks on the X- and Y-axes will be different even if the same tick length percentage value is specified for both XT and YT, unless the area defined by P1 and P2 is square.

The plotter, when initialized, automatically sets the tick length values to 0.5% of the scaling lengths ($P2_Y - P1_Y$) and ($P2_X - P1_X$). A TL instruction with no parameters will default to the same values. A TL instruction with only one parameter specifies the length of tp, and tn will be zero. A negative tp parameter will draw a negative tick just as would be drawn by a tn with a positive parameter. Likewise, a negative tn parameter will draw a positive tick. Use of negative parameters is not recommended both because the results are more difficult to visualize and programs with negative parameters will not be compatible with other HP plotters. A TL instruction remains in effect until another TL instruction with valid parameters is executed or an IN or DF instruction is executed.

The following example draws both tick marks and grid lines. The grid lines are a result of specifying 100% tick length. The horizontal tick marks on the left-most grid line are drawn using the default tp,tn. The tick marks on the second grid line have a positive tick length of 1% and no negative tick. The tick marks on the third grid line have no positive tick and a negative tick length of 5%. Note that these last tick marks are drawn by the YT instruction even though the PU instruction is in effect. However, the moves to the next tick location are made with the pen up, and hence, the grid line is not retraced. A reduced version of the plot follows.

```
10    ASSIGN @Hp7090 TO 705
20    OUTPUT @Hp7090;"IN;IP300,300,10300,7500;"
30    OUTPUT @Hp7090;"PA300,300;SP1;PD;TL100;XT;"
40    FOR I=1 TO 10
50    OUTPUT @Hp7090;"PR1000,0;XT;"
60    NEXT I
70    OUTPUT @Hp7090;"TL;PU;PA300,300;PD;"
80    GOSUB 150
90    OUTPUT @Hp7090;"TL1,0;PU;PA1300,300;PD;"
100   GOSUB 150
110   OUTPUT @Hp7090;"TL0,5;PU;PA2300,300;"
120   GOSUB 150
130   OUTPUT @Hp7090;"PA300,7500;TL100;YT;PU;SP0;"
140   STOP
150   !SUBROUTINE TO DRAW TICKS
160   FOR J=1 TO 9
170   OUTPUT @Hp7090;"PR0,720;YT;"
180   NEXT J
190   RETURN
200   END
```



The Symbol Mode Instruction, SM

USES: The SM instruction is used with PA and PR instructions, and provides the means to draw a single character which is centered at the end of each vector. Symbol mode plotting can be used to create scattergrams, geometric drawings, or multiple-line graphs where lines are easy to differentiate.

SYNTAX: *SM* character terminator
or
SM terminator

Parameter	Format	Range	Default
character	character	all printing characters except semicolon	off

EXPLANATION: An SM instruction without parameters (SM;) turns off symbol mode. When a parameter is present, it is limited to a single character, which must be one of the printing characters of the character set currently selected.

NOTE: Remember that the first character after the mnemonic will be interpreted as the parameter. ■

After an SM instruction has been executed, subsequent PA and PR instructions function as described in Chapter 5, except that the specified symbol mode character is drawn at the end of each vector and is centered on the plotted point. (A character drawn at a point using the label instruction, LB, would not be centered on the point.) Drawing of the character is independent of the current pen state (up or down); the character is always drawn at each point specified in the PA and PR instruction.

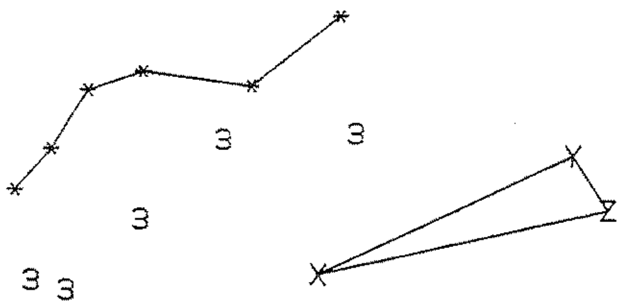
The character is drawn according to the character set selected when the SM instruction is executed. The character does not change even if a new set is selected. An SM instruction remains in effect until another valid SM instruction is executed or an IN or DF instruction is executed. The size (SI and SR), slant (SL), and direction (DI and DR) instructions affect the character drawn.

An SM instruction can specify any printing character (decimal values 33 through 126). The semicolon (decimal value 59) is used only to cancel symbol mode (SM;) and cannot be selected as the symbol to be drawn at the endpoint of each vector. Specifying a space (decimal value 32) or any control character also cancels symbol mode.

The following example shows symbol mode plotting with the pen up and the pen down as might be used in line graphs, geometric drawings, and scattergrams.

```
10    ASSIGN @Hp7090 TO 705
20    OUTPUT @Hp7090;"IN;SP1;SM*;PA200,1000;"
30    OUTPUT @Hp7090;"PD400,1230,600,1560,900,1670;"
40    OUTPUT @Hp7090;"PA1500,1600,2000,2000;"
50    OUTPUT @Hp7090;"PU;SM;PA100,300;SM3;"
60    OUTPUT @Hp7090;"PA300,500,500,450,900,850;"
70    OUTPUT @Hp7090;"PA1350,1300,2100,1350PU;"
80    OUTPUT @Hp7090;"SM;PA1900,560;PD;SMY;PA3300,1250;"
90    OUTPUT @Hp7090;"SMZ;PA3500,950;SMX;"
100   OUTPUT @Hp7090;"PA1900,560;PU;SP0;"
110   END
```

Plot showing symbol mode:



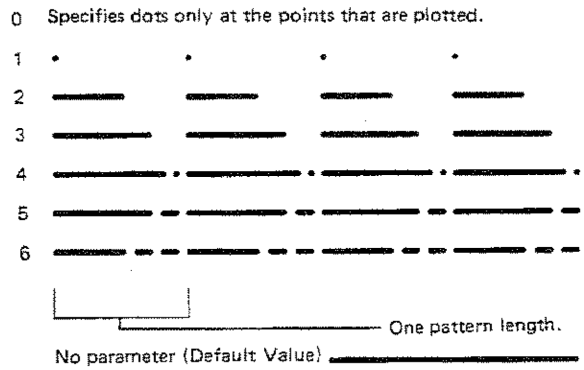
The Line Type Instruction, LT

USES: The LT instruction specifies the type of line that will be used with PA and PR instructions. Use it to draw dashed or dotted lines. This facilitates trace differentiation on multiple-line graphs and enables emphasis or deemphasis of plotted lines or grids. One line type causes only dots to be plotted at each data point.

SYNTAX: *LT* pattern number (, pattern length) terminator
or
LT terminator

Parameter	Format	Range	Default
pattern number	real	−128.0000 to 127.9999	solid line
pattern length	real	−128.0000 to 127.9999	4% of diagonal (P2 − P1)

EXPLANATION: Shown below are the line patterns and their pattern numbers.



The shaded portion of each of the line patterns is one complete segment of the pattern.

The pattern number parameter is in real format but any decimal fraction is truncated. This parameter should be between 0 and 6; a parameter in this range sets the line type as shown in the preceding illustration. A parameter in the range 7 to 127.9999 is ignored; the line type does not change and no error is set. A parameter 128 or greater sets error 3 and the line type does not change. A negative parameter between 0 and −128 defaults to a solid line type and no error is set. A negative parameter less than −128 sets error 3 and the line type does not change.

When the first parameter is between 0 and 127.9999, the second parameter is used. This optional pattern length parameter is in real format. Both integer and fractional parts are used. The parameter specifies the length of one complete pattern and is expressed as a percentage of the diagonal distance between the scaling points P1 and P2. When this parameter is positive and less than 127.9999, the pattern length is set to this length. When this parameter is negative or is greater than or equal to 128, the previous pattern length is used and error 3 is set. If a pattern length parameter is not specified, a length of 4% is used.

NOTE: If a vector ends in the pen-up portion of the pattern, a pen down instruction, PD, will not physically put the pen down until the next vector instruction is executed and the pen has moved so it is in a pen-down portion of a pattern segment. The pen up instruction clears the carry-over portion of a pattern segment. ■

Notes

Chapter 8
Labeling

What You'll Learn in This Chapter

In this chapter you will learn about character sets and labels used to create effective annotated graphics. You will learn how to designate and select character sets, how to use the label instruction with both constant and variable parameters, and how to set the size, slant, and direction of labels. Character spacing, moving the pen any number of character widths and/or lines, and automatic label positioning are also explained.

Instructions Covered

CA The Designate Alternate Character Set Instruction
CP The Character Plot Instruction
CS The Designate Standard Character Set Instruction
DI The Absolute Direction Instruction
DR The Relative Direction Instruction
DT The Define Terminator Instruction
LB The Label Instruction
LO The Label Origin Instruction
SA The Select Alternate Character Set Instruction
SI The Absolute Character Size Instruction
SL The Character Slant Instruction
SR The Relative Character Size Instruction
SS The Select Standard Character Set Instruction

Terms You Should Understand

Label Terminator — the final character in every label string; it takes the plotter out of label mode so that characters are no longer drawn but are again interpreted as HP-RL or HP-GL instructions and parameters. Its default value is the ASCII character ETX (decimal equivalent 3), but it may be redefined using the DT instruction.

Character Space Field — the space occupied by a single character, together with the space between it and the next character and the space above the character which separates it from the previous text line.

Label Start Point — the current pen position. Before executing the LB instruction, move the pen to the location where labeling is to begin. You can do this by using, for example, a PA, PR, or CP instruction or by using the front-panel controls.

The Label Instruction, LB

USES: The LB instruction is used to letter text, expressions, or string variables using the currently defined character set. Use it to annotate graphs or create text-only overhead transparencies.

SYNTAX: *LB* c...c t

where t is the label terminator, either the default ETX character (decimal equivalent 3), or another character defined by the DT instruction.

Parameter	Format	Range	Default
c...c	character	all ASCII characters	off

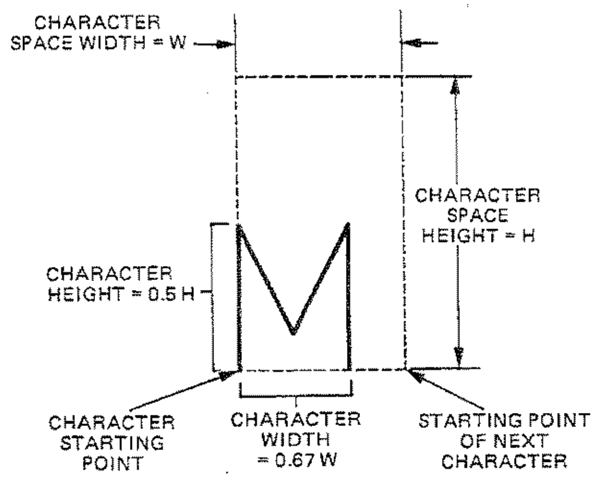
EXPLANATION: All printing characters following the LB mnemonic are drawn using the currently selected character set. The set used is specified by the CA or CS instructions and selected by the SA or SS instructions, or the ASCII control characters shift-out or shift-in (decimal equivalent 14 and 15 respectively). If not specified, the default character set (set 0) is used.

The direction, size, and slant of the characters assume default values if not previously specified by DI, DR, SI, SR, or SL instructions. The label begins at the current pen position unless its placement has been set using an LO instruction.

The label mode can be terminated only by sending a label terminator at the end of the character string. Refer to The Define Terminator Instruction, DT. Unless a label string is terminated, subsequent HP-RL and HP-GL instructions will appear as labels in your plot.

Before executing the LB instruction, move the pen to the location where labeling is to begin using, for example, a PA, PR, or CP instruction or by using the front-panel controls. This establishes the lower-left corner of the first character space and the carriage-return point.

Character spacing and line spacing are functions of character size. In the following diagram, you can see the relative position of a character, in this case M, within the character space. The character-space field is set indirectly by the SI or SR instructions, since the character space height is twice the character's height and the character-space is 1½ times the character's width. The space above and beside a drawn character becomes the spacing between lines and characters. The character space is illustrated as follows.



When you specify the height of a character in an SI or SR instruction, however, you should specify the character height, not the height of a character space.

Control characters, such as carriage return, line feed, or inverse line feed (VT) can be included in the label character string. These characters are not labeled, but cause the specified function to be executed. For example, a carriage return character causes the pen to return to a defined carriage return point prior to labeling subsequent characters in the label character string. The carriage return point is established as follows:

- The endpoint of the last vector drawn becomes the carriage return point.
- The point moved to using a front-panel control becomes the carriage return point.
- The current pen position when a DI or DR instruction is executed becomes the carriage return point. Refer to the DI instruction example in this chapter.

The Define Terminator Instruction, DT

USES: The DT instruction provides the means to specify the character to be used as the label terminator. Use it to change the label terminator from its default value if ETX (decimal equivalent 3) cannot be used by your computer.

SYNTAX: *DT* t terminator

where t is the label terminator

Parameter	Format	Range	Default
t	character	ASCII characters with decimal values 1 through 127	ETX (decimal value 3)

EXPLANATION: The label mode can only be terminated by sending a label terminator at the end of the label character string. ASCII control characters (decimal equivalent 1 through 32 and 127) can be defined as label terminators and will not print when invoked, although the function normally performed by the character will be performed (i.e., LF will terminate a label but will also cause a line feed). ASCII characters with decimal equivalent values 33 through 126 can also be defined as the terminator, but the character will also be printed at the end of the label character string. The ASCII character "null" (decimal equivalent 0) cannot be defined as the label terminator.

NOTE: A DT instruction with no parameter (DT;) does not establish ETX as the default terminator, since the character immediately following the mnemonic DT is taken as a parameter. Only a DF or IN instruction or use of the ETX character itself as the instruction's parameter can be used to reestablish ETX as the label terminator. ■

The following examples of text in a label instruction demonstrate the use of the label terminator.

NOTE: Remember to use the equivalent code for your computer whenever you encounter the ASCII code, ETX, in a program. On all HP Series 200 computers, **CTRL C** will access the ETX character. On many other computers, you can use CHR$(3). ■

```
10  ASSIGN @Hp7090 TO 705
20  OUTPUT @Hp7090;"IN;SP1;SC0,5000,0,5000;"
30  OUTPUT @Hp7090;"PA0,4500;"
40  OUTPUT @Hp7090;"LBDefault control character ETX␍␊␃"
50  OUTPUT @Hp7090;"LBterminates by performing end-␍␊␃"
60  OUTPUT @Hp7090;"LBof-text function.␃"
70  OUTPUT @Hp7090;"PA0,3900;DT#;"
80  OUTPUT @Hp7090;"LBPrinting characters terminate,␍␊#"
90  OUTPUT @Hp7090;"LBbut are also printed.#"
100 OUTPUT @Hp7090;"PA0,3400;DT␍;"
110 OUTPUT @Hp7090;"LBControl characters terminate␊␍"
120 OUTPUT @Hp7090;"LBand perform their function.␍"
130 END
```

Default control character ETX
terminates by performing end-
of-text function.

Printing characters terminate,
#but are also printed.#

Control characters terminate
and perform their function.

Labeling with Variables

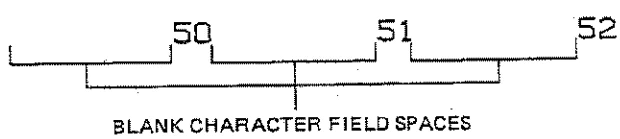
In some applications, it is desirable to label the plot using variables rather than literals to define the label string. Many different conventions are used in different computer languages and computers to define variable length and the character field format in which these variables will be printed. To avoid unexpected placement of the labels defined by variables, refer to your computer manual for a definition of the conventions used for PRINT and OUTPUT statements.

On HP series 200 computers, quotation marks are used by PRINT and OUTPUT statements to define the literal characters that are to be sent, but variables are not included within quotation marks. With PRINT statements the comma is used as a separator between variables to cause the label string to be right-justified in a specific character-field width. The unused character positions in this field are sent as leading blank spaces to establish fixed spacing between label strings. For close spacing of label strings, the blank spaces can be suppressed by substituting a semicolon as a separator between variables.

With OUTPUT statements, variables are sent only with a leading sign space. Variables followed by a semicolon are sent with nothing following them. Numeric variables followed by a comma are sent with a comma following them. String variables followed by a comma are sent with a CRLF following them.

The following example illustrates using PRINT with a comma to establish fixed spacing when using variables for labeling. When the value of X is 50, the labels shown are produced by the given HP-GL instructions. The first string causes the plotter to label the value of X, X + 1, and X + 2. Blank spaces between the printed integers include space for the sign, but only a minus sign is printed. The number of blank character-field spaces may vary with different computers.

```
"LB",X,X+1,X+2,"␃"
```



The following example illustrates the closer spacing achieved with PRINT when semicolons separate variables in labeling commands. The semicolons between the variables cause suppression of blank spaces. The space between the printed integers varies with different computers, but normally includes only the sign space.

```
"LB";X;X+1;X+2;"␃"
```

50 51 52

Any spaces required to fit into the context of the item being labeled must normally be sent enclosed in quotes. The following example labels the same variables as before, but with four extra spaces between each of the integers. Note that four spaces enclosed in quotes are sent between each variable, but the semicolon suppresses unwanted blank spaces.

```
"LB";X;"    ";X+1;"    ";X+2;"␃"
```

Character Sets

The HP 7090 has the capability of lettering with any of 5 internal character sets. Most of the character sets have identical upper- and lowercase alphabetic characters and identical numerals. The symbols and punctuation marks vary from set to set, making annotation in several languages possible. The plotter, when initialized, automatically sets both the standard and alternate character sets to character set 0 which follows:

```
CHARACTER SET 0
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~
```

Some examples of annotation in foreign languages are found below. Notice that the label string in the HP-GL label instruction uses the CHR$ function if that ASCII character code is not available on the computer's keyboard.

```
"PR5000,5000;SP1;SI.37,.50;"
"CS2;LB60 & DRU";CHR$(123);"BER␃"
```

60 & DRÜBER

```
"PR5000,4000;SP1;SI.37,.50;"
"CS4;LB#su compan";CHR$(124);"ia?␃"
```

¿su compañia?

```
"PR5000,3000;SP1;SI.37,.50;"
"CS3;LB35-50 A";CHR$(124);"R␃"
```

35-50 ÅR

The HP 7090 will perform an automatic backspace before drawing an accent above the letter. Therefore, when an accented letter is required, enter the letter first, followed by the accent.

For a complete listing of all 5 character sets, refer to Appendix C.

The Designate Standard Character Set Instruction, CS

USES: The CS instruction is used to designate one of the 5 character sets (0–4) as the standard set. Use it to change the standard character set to one with characters appropriate for your application. It is especially useful when labels are in a language other than English.

SYNTAX: *CS* set number terminator
or
CS terminator

Parameter	Format	Range	Default
set number	integer	0 to 4	0

EXPLANATION: The character set number can be 0 to 4. The set designated by the CS instruction is used for all labeling operations when the standard set is selected by the SS instruction or by the control character shift-in (decimal 15) in a label string. Character set 0 is automatically designated as the standard character set whenever the plotter is initialized or set to default values.

A CS instruction executed while the standard set is selected will immediately change the character set used for labeling. CS instructions executed while the alternate set is selected will not change the set used for labeling until the standard set is selected.

A CS instruction with no parameters (CS ;) defaults to set 0. A CS instruction with invalid parameters sets error 5 (unknown character set), and the instruction is ignored.

The Designate Alternate Character Set Instruction, CA

USES: The CA instruction is used to designate one of the 5 character sets (0–4) as the alternate character set. Use it to provide an additional character set that can be easily accessed from a program, especially when a single label contains characters found in two different sets.

SYNTAX: *CA* set number terminator
or
CA terminator

Parameter	Format	Range	Default
set number	integer	0 to 4	0

EXPLANATION: The character set number may be from 0 to 4. The set designated by the CA instruction is used for all labeling operations when the alternate set is selected by the SA instruction or by the control character shift-out (decimal 14) in a label string. Character set 0 is automatically designated as the alternate character set whenever the plotter is initialized or set to default values.

A CA instruction executed while the alternate set is selected will immediately change the character set used for labeling. CA instructions executed while the standard set is selected will not change the set used for labeling until the alternate set is selected.

A CA instruction with no parameters (CA ;) defaults to set 0. A CA instruction with invalid parameters sets error 5 (unknown character set), and the instruction is ignored.

The Select Standard Set Instruction, SS

USES: The SS instruction is used to select the standard set designated by the CS instruction as the character set to be used for all labeling. Use it to shift from the currently designated alternate character set to the currently designated standard character set so characters in that set may be accessed. Using the control character shift-in (decimal 15) inside a label string is equivalent to executing this instruction.

SYNTAX: *SS* terminator

PARAMETERS: None

EXPLANATION: Any parameters which follow the instruction set error 2, and the standard set is selected.

The standard ASCII character set (set 0) is automatically selected when the plotter is first turned on, initialized, or set to default values. The standard set can be selected within a label instruction by sending the ASCII control character for shift-in (decimal 15).

The Select Alternate Set Instruction, SA

USES: The SA instruction is used to select the alternate set designated by the CA instruction as the character set to be used for all labeling. Use it to shift from the currently designated standard character set to the currently designated alternate character set to access characters in the second set. Sending the control character shift-out (decimal 14) inside a label string is equivalent to executing this instruction.

SYNTAX: *SA* terminator

PARAMETERS: None

EXPLANATION: Any parameters which follow the instruction set error 2, and the alternate set is selected.

The instruction should be executed before executing a label instruction whenever the alternate character set is to be used. The alternate set can be selected within a label instruction by sending the ASCII control character for shift-out (decimal 14). Shift-in and shift-out are particularly useful when a line of text must be composed with symbols from two character sets.

The following instructions label using two different character sets where the underline is drawn with and without a backspace. The shift-out character is used to change from the standard to the alternate set.

```
"PA5000,5000;"
"SP2;CS0;CA4;SS;LBS_E_T_O_␎S_E_T_4_␏"
```

```
S_E_T_O SET4
```

The Absolute Direction Instruction, DI

USES: The DI instruction specifies the direction in which characters are lettered. Use it to change the direction of labeling to a new absolute direction; by absolute we mean independent of P1, P2 settings. It is expecially useful for labeling a Y-axis or labeling a vertical graph.

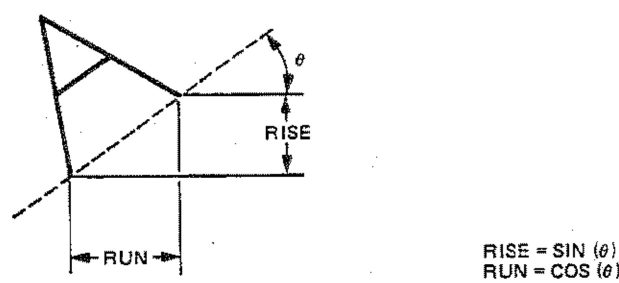
SYNTAX: *DI* run, rise terminator
or
DI terminator

Parameter	Format	Range	Default
run	real	−128.0000 to 127.9999	1
rise	real	−128.0000 to 127.9999	0

EXPLANATION: Run and rise are in real format in the range −128.0000 to 127.9999 and specify the direction according to the relationship:

$$\theta = \tan^{-1}\left(\frac{\text{rise}}{\text{run}}\right)$$

where:



At least one parameter must be effectively nonzero; i.e., $|\geqslant 0.0004|$.

A DI instruction with a rise parameter of zero will produce horizontal labeling. A DI instruction with a run parameter of zero will produce vertical labeling.

A DI instruction with no parameters (DI;) will default to the values DI 1,0 (horizontal). A DI instruction with only one parameter will set error 2, and the instruction will be ignored. A DI instruction with more than two parameters will set error 2, and the instruction will be executed using the first two parameters.

A change in the orientation of P1 and P2 will not affect the direction of labeling. A DI instruction remains in effect until another DI or DR instruction is executed, or the plotter is initialized.

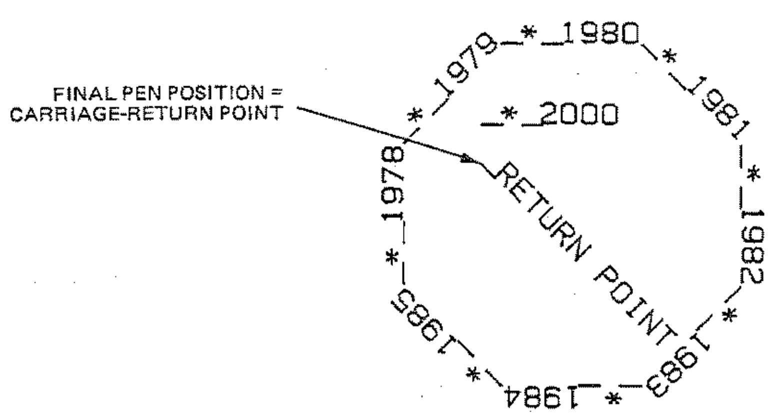
A DI instruction updates the carriage-return point to the current pen position.

When the angle, θ, necessary to establish the desired label direction is known, the instruction DI $\cos\theta$, $\sin\theta$ can be used to establish label direction.

The following example labels the years 1978 through 1985, in a circular pattern starting with vertical labeling. The direction in which each year is labeled is changed by 45 degrees. Then the labels in the center are drawn to illustrate the use of cosine and sine values as parameters. The label _*_2000 contains both a carriage return and a line feed character before the label terminator, ETX, so the pen position at the end of that label is one line below the beginning of that label. The fact that DI instructions update the carriage-return point can be clearly seen by observing the pen's position at the end of the program. The final character in the last label is a carriage return and the pen returns to the carriage-return point, the position of the pen at the last DI instruction.

NOTE: Remember to use the equivalent code for your computer whenever you encounter the ASCII code, ETX, in a program. On all HP Series 200 computers, CTRL C will access the ETX character. On many other computers, you can use CHR$(3). ■

```
10  ASSIGN @Hp7090 TO 705
20  DEG
30  OUTPUT @Hp7090;"IN;SP1;PA1050,4450;"
40  OUTPUT @Hp7090;"DI0,1;LB_*_1978␃DI1,1;LB_*_1979␃"
50  OUTPUT @Hp7090;"DI1,0;LB_*_1980␃DI1,-1;LB_*_1981␃"
60  OUTPUT @Hp7090;"DI0,-1;LB_*_1982␃DI-1,-1;"
70  OUTPUT @Hp7090;"LB_*_1983␃"
80  OUTPUT @Hp7090;"DI-1,0;LB_*_1984␃DI-1,1;"
90  OUTPUT @Hp7090;"LB_*_1985␃"
100 OUTPUT @Hp7090;"PA1500,5350;DI;";COS(0);SIN(0)
110 OUTPUT @Hp7090;"LB_*_2000␍␊␃"
120 OUTPUT @Hp7090;"DI";COS(-45);SIN(-45)
130 OUTPUT @Hp7090;"LB_RETURN POINT␍␃"
140 END
```



The Relative Direction Instruction, DR

USES: The DR instruction specifies the direction in which characters are lettered. Use it to change the direction of lettering from its default direction, horizontal, to a direction which is relative to P1,P2 settings. It is useful when creating graphs which will be plotted in several sizes and you want labels to have the same relationship to the data on all plots.

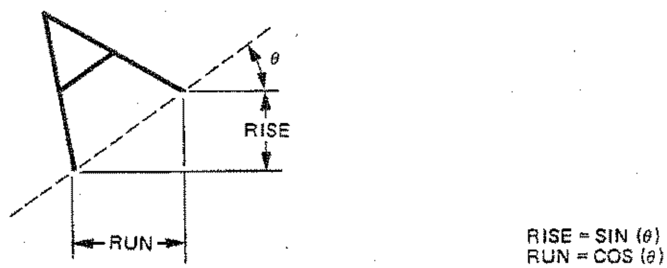
SYNTAX: *DR* run, rise terminator
or
DR terminator

Parameter	Format	Range	Default
run	real	−128.0000 to 127.9999	1
rise	real	−128.0000 to 127.9999	0

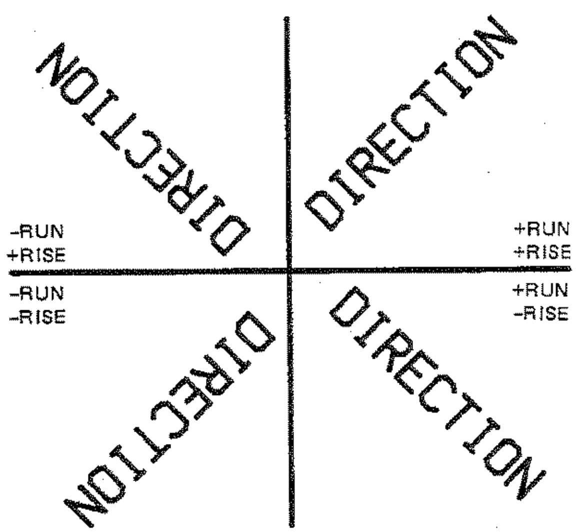
EXPLANATION: Run and rise are in real format in the range −128.0000 to 127.9999 and specify the label direction according to the relationship:

$$\theta = \tan^{-1}\left(\frac{\text{rise}}{\text{run}}\right)$$

where:



If you imagine the current pen position to be the origin, the sign of the parameters determines in which quadrant the lettering will be. In the example below, run and rise assume all combinations of ±1 with default P1 and P2.



A change in P1 or P2 will affect the direction of lettering. Refer to the section Parameter Interaction in Labeling Instructions in this chapter.

A DR instruction remains in effect until another DR or DI instruction is executed or the plotter is initialized.

A DR instruction with no parameters (DR;) will default to the values DR 1,0 (horizontal).

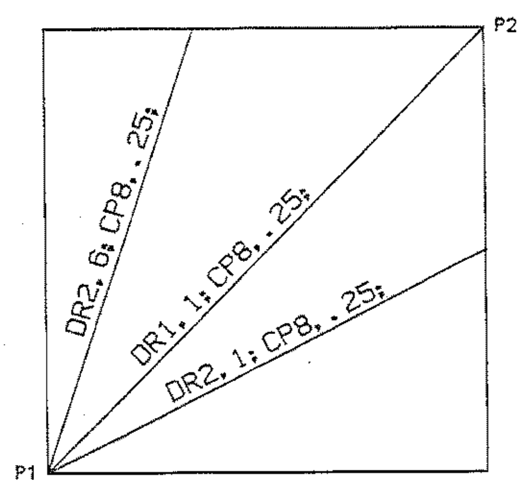
Specifying both parameters as zero will set error 3, and the instruction will be ignored. Specifying only one parameter will set error 2, and the instruction will be ignored. Specifying more than two parameters will set error 2, and the instruction will be executed.

The following description may help you visualize the direction of labeling using the DR instruction with various parameters. Think of directional lines as being parallel to a line starting at the lower-left scaling point (usually P1) and intersecting the opposite side or the top edge of the current P1/P2 frame.

Form a fraction in lowest terms $\leqslant 1$ using the run and rise parameters. If run = rise, the fraction will equal 1 and the directional line will go from P1 to P2. If run > rise, the directional line will intersect the side of the frame that fraction of the way up toward P2. If rise > run, the directional line will intersect the top of the frame that fraction of the way across toward P2. Remember, since lettering starts at the current pen position, labels will be parallel to these lines, not necessarily along them.

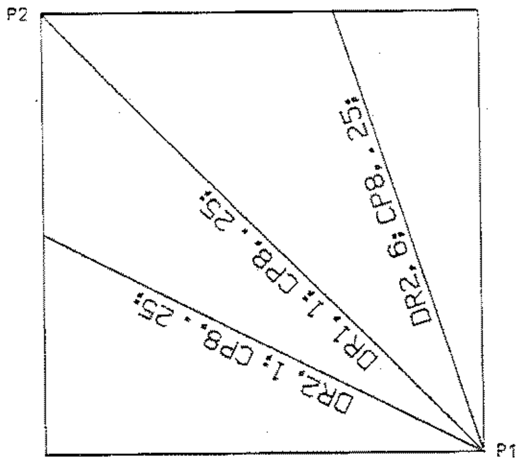
The accompanying program illustrates the DR instruction with a given P1 and P2.

```
10  ASSIGN @Hp7090 TO 705
20  OUTPUT @Hp7090;"IN;SP1;IP2500,1250,5500,4250;"
30  OUTPUT @Hp7090;"SC0,5000,0,5000;SI.20,.30;"
40  OUTPUT @Hp7090;"PA0,0;PRPD0,5000,5000,0;"
50  OUTPUT @Hp7090;"PR0,-5000,-5000,0;"
60  OUTPUT @Hp7090;"PA5000,5000,PU,0,0;DR1,1;CP8,.25;"
70  OUTPUT @Hp7090;"LBDR1,1;CP8,.25;␃"
80  OUTPUT @Hp7090;"PR0,0,PD,1667,5000,PU;"
90  OUTPUT @Hp7090;"DR2,6;PA0,0;CP8,.25;"
100 OUTPUT @Hp7090;"LBDR2,6;CP8,.25;␃"
110 OUTPUT @Hp7090;"PR0,0,PD,5000,2500,PU;"
120 OUTPUT @Hp7090;"DR2,1;PA0,0;CP8,.25;"
130 OUTPUT @Hp7090;"LBDR2,1;CP8,.25;␃"
140 END
```



Substitute the following instruction string in line 20. Note that changing the position of P1 and P2 will cause the plot and the label's direction to change as shown in the following illustration.

```
"IN;SP1;IP5500,1250,2500,4250;"
```



The Absolute Character Size Instruction, SI

USES: The SI instruction specifies the actual size of characters and symbols in centimetres. Use it to change the character size from its default value or to another value and establish absolute character sizing in centimetres so character size is not dependent on the settings of P1 and P2.

SYNTAX: *SI* width, height terminator
or
SI terminator

Parameter	Format	Range	Default	
			A/A4	**B/A3**
width	real	−128.0000 to 127.9999	0.187	0.285
height	real	−128.0000 to 127.9999	0.269	0.375

EXPLANATION: If parameters are included, two parameters are required, width and height. The defined width and height are interpreted as centimetres, must be in real format, and may have any value between −128.0000 and 127.9999.

An SI instruction with no parameters (SI;) will default character size to the values shown in the following table.

Paper Size	Width	Height
A/A4	0.187 cm	0.269 cm
B/A3	0.285 cm	0.375 cm

An SI instruction remains in effect until another valid SI or SR instruction is executed or the plotter is initialized or set to default conditions. An SI instruction with only one parameter sets error 2, and the instruction is not executed. An SI instruction with more than two parameters sets error 2, and the instruction is executed.

The following example draws the instrument's model number, HP 7090, at the specified width of 1 cm and height of 1.5 cm.

```
"SI1,1.5;LBHP 7090␃"
```

Negative SI parameters will produce mirror images of labels. A negative SI width parameter will mirror labels in the right-to-left direction.

INSTRUCTION	RESULTING LABEL
"SI-.35,.6;LBHP␃"	

A negative height parameter will mirror labels in the top-to-bottom direction.

INSTRUCTION	RESULTING LABEL
"SI.35,-.6;LBHP␃"	

Two negative SI parameters will mirror the label in both directions and the label will appear to be rotated 180 degrees.

INSTRUCTION	RESULTING LABEL
"SI-.35,-.6;LBHP␃"	

For further information on the effects of negative parameters, refer to the section Parameter Interaction in Labeling Instructions later in this chapter.

To produce legible characters, parameters should be greater than 0.1. Parameter values above 18 allow no more than two characters to be drawn on the paper.

The Relative Character Size Instruction, SR

USES: The SR instruction specifies the size of characters and symbols as a percentage of the distance between scaling points P1 and P2. Use it to define character size relative to the distance between P1 and P2 so that if the P1,P2 distance changes, character size will adjust to occupy the same "relative" amount of space.

SYNTAX: *SR* width, height terminator
or
SR terminator

Parameter	Format	Range	Default
width	real	−128.0000 to 127.9999	0.75
height	real	−128.0000 to 127.9999	1.5

EXPLANATION: If parameters are included, two parameters are required, width and height. The defined width and height are interpreted as a percentage of the algebraic distance between the X- or Y-coordinates of P1 and P2. The parameters are in real format and may have any value between −128.0000 and 127.9999.

An SR instruction with no parameters (SR;) will default to the values 0.75 for width and 1.5 for height, which, when P1 and P2 are at default values, produces letters the same size as an SI instruction without parameters.

An SR instruction remains in effect until another valid SR or SI instruction is executed or the plotter is initialized or set to default conditions. An SR instruction with only one parameter sets error 2, and the instruction is ignored. An SR instruction with more than two parameters sets error 2, and the instruction is executed.

The following example shows how changes in P1 and P2 affect labels drawn while an SR instruction is in effect. The upper label is written with default character size. Then P1 and P2 are changed to define a square area with 6000-plotter-unit sides. A new label is drawn. Next a new SR instruction is executed with both width and height parameters set to three percent. Because the area established by P1 and P2 is square, equal parameters create square letters. With default P1 and P2 settings, equal parameters do not create square letters.

```
10  ASSIGN @Hp7090 TO 705
20  OUTPUT @Hp7090;"IN;SP1;PA500,7000;LBDEFAULT SIZE␃"
30  OUTPUT @Hp7090;"IP1000,1000,7000,7000;PA500,6500;"
40  OUTPUT @Hp7090;"LBNEW P1 AND P2 CHANGE LABEL SIZE␃"
50  OUTPUT @Hp7090;"SR3,3;"
60  OUTPUT @Hp7090;"PA500,6000;LBNEW SR INSTRUCTION␍␊␃"
70  OUTPUT @Hp7090;"LBCHANGES LABEL SIZE␃"
80  END
```

DEFAULT SIZE

NEW P1 AND P2 CHANGE LABEL SIZE

NEW SR INSTRUCTION CHANGES LABEL SIZE

Either negative SR parameters or switching the relative positions of P1 and P2 will produce mirror images of labels. Refer to The Absolute Size Instruction, SI, and Parameter Interaction in Labeling Instructions for more information on mirroring.

With default P1 and P2, the useful range of width and height parameters which produces legible characters and a label of suitable length is approximately 0.6 to 5.

The Character Slant Instruction, SL

USES: The SL instruction specifies the slant with which characters are lettered. Use it to create slanted text, for emphasis, or to reestablish upright labeling after an SL instruction with parameters has been in effect.

SYNTAX: *SL* tan θ terminator
or
SL terminator

Parameter	Format	Range	Default
tan θ	real	−128.0000 to 127.9999	0

EXPLANATION: The instruction may be used with or without a parameter. When a parameter is included, it is interpreted as the tangent of the angle from vertical as shown below. The parameter is in real format and may have any value between −128.0000 and 127.9999.

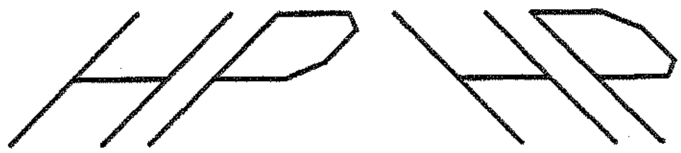
An SL instruction without a parameter (SL;) defaults to the same value as SL0, and labels are not slanted.

An SL instruction remains in effect until another valid SL instruction is executed or the HP 7090 is initialized or set to default conditions. An SL instruction with more than one parameter sets error 2, and the instruction is executed.

The useful parameter range is ±0.05 to ±2 when using default-size characters and up to ±3.5 for large letters.

The following example letters HP at a slant of +45 degrees and −45 degrees.

```
"DF;SP1;SI1.3,1.8;PR3000,6000;"
"SL1;LBHP␃"
"SL-1;PR1300,0;LBHP␃"
```



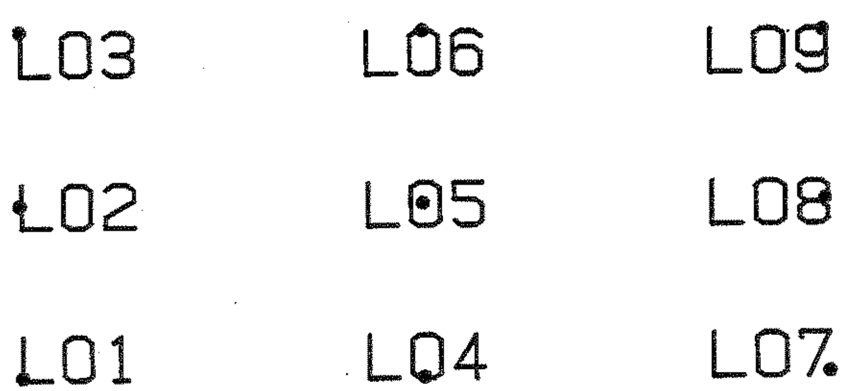
The Label Origin Instruction, LO

USES: The LO instruction is used to position labels relative to the current pen position. Use it to center labels or justify them to the left or right of the current pen position. Positioning can be above or below the current pen position and can also be offset by an amount equal to ½ the character's width and height.

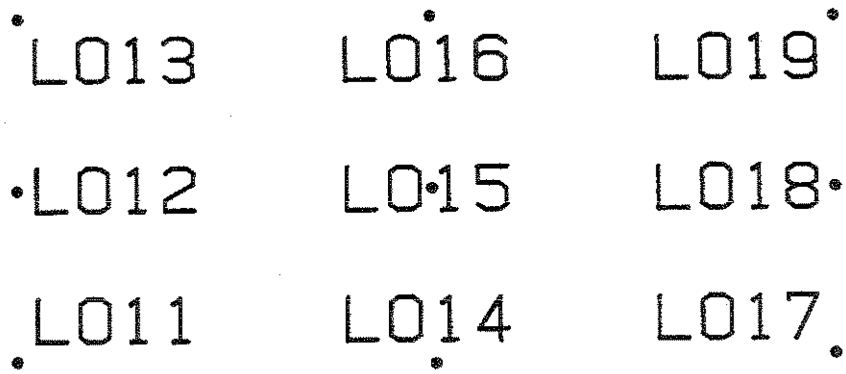
SYNTAX: *LO* position number terminator
or
LO terminator

Parameter	Format	Range	Default
position number	integer	1–9 and 11–19	1

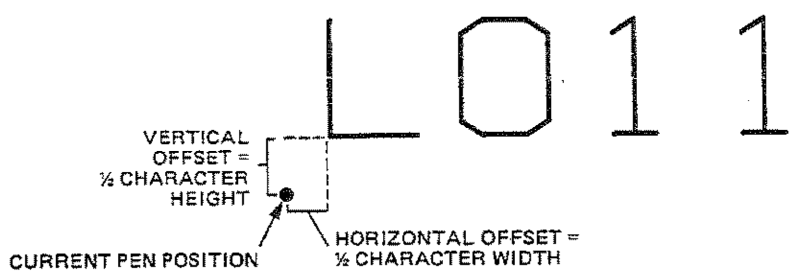
EXPLANATION: The position number can be an integer from 1 to 9, or 11 to 19, and determines the position of the label with respect to the current pen position. The illustration below graphically summarizes this relationship by showing the instructions LO 1 through LO 9 in the labeled position which they produce. The dots represent the current pen position in each case.



The positions resulting from the instructions LO 11 through LO 19 are the same as shown for the instruction LO 1 through LO 9, except that the labels are offset from the current pen position as shown next.



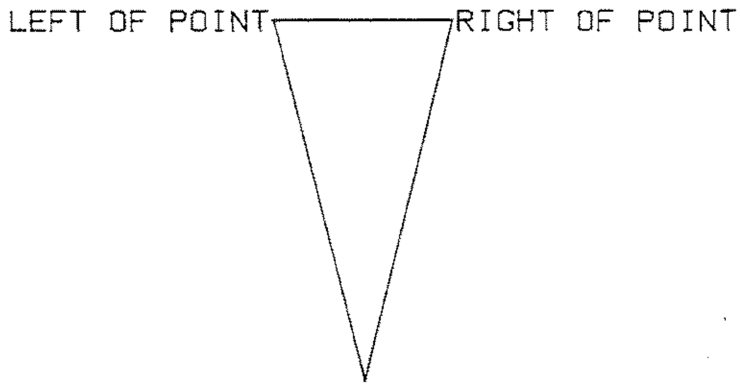
The amount of offset is equal to ½ of the character's width and ½ of the character's height as specified by the most recent SI or SR instruction. The offset is shown below.



An LO instruction remains in effect until another LO instruction is received or the plotter is initialized or set to default conditions. An LO instruction with no parameters (LO;) sets the default label origin, LO 1.

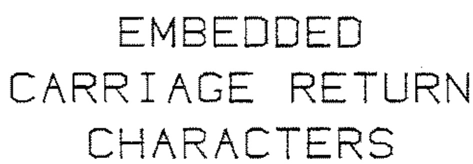
The pen position is updated after each character is drawn, and the pen automatically moves to the next character origin in anticipation of additional characters. If the pen position is to be returned to its location prior to the label, a carriage return character must be sent following the label string and before the label terminator. The following example demonstrates the use of the carriage return with an LO instruction.

```
10  ASSIGN @Hp7090 TO 705
20  OUTPUT @Hp7090;"IN;SP1;PA4000,3000;PRPD-500,2000;"
30  OUTPUT @Hp7090;"LO18;LBLEFT OF POINT␍␃"
40  OUTPUT @Hp7090;"PR1000,0;LO12;LBRIGHT OF POINT␍␃"
50  OUTPUT @Hp7090;"PR-500,-2000;PU;SP0;PA10000,7500;"
60  END
```



When carriage return characters are embedded in an LB label string, each portion of the label preceding, between, and following a carriage return character is positioned according to the label origin. This is shown in the following illustration.

```
10  ASSIGN @Hp7090 TO 705
20  OUTPUT @Hp7090;"IN;SP1;PA5000,3500;"
30  OUTPUT @Hp7090;"LO14;PD;PU;SI.29,.39;"
40  OUTPUT @Hp7090;"LBEMBEDDED␍␊CARRIAGE RETURN␍␊␃"
50  OUTPUT @Hp7090;"LBCHARACTERS␃"
60  END
```



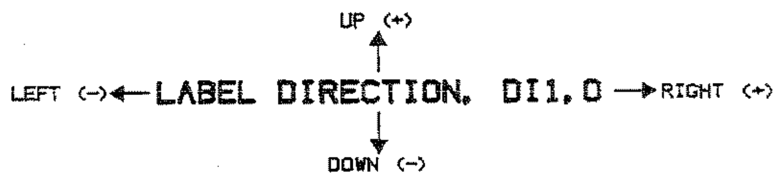
The Character Plot Instruction, CP

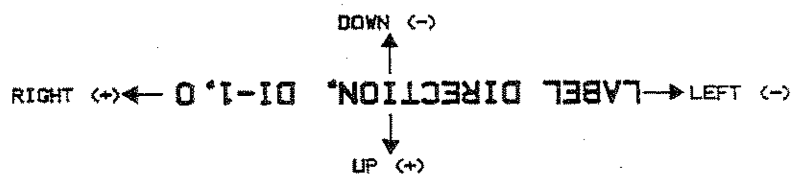
USES: The CP instruction is used to move the pen the specified number of character-space widths (spaces) and heights (lines). Use it to move the pen any number of character spaces or lines from a point on the plotting surface, to align with a left-margin, or to center or right-justify a label. Thus, the label can be moved slightly above or below a line, spaces or lines can be inserted in text, or labels can be centered.

SYNTAX: *CP* spaces, lines terminator
or
CP terminator

Parameter	Format	Range	Default
spaces	real	−128.0000 to 127.9999	Carriage return point
lines	real	−128.0000 to 127.9999	

EXPLANATION: The space and line parameters are in terms of character-space width and height. The spaces parameter moves the pen the specified number of character-space widths to the right (a positive value) or the left (a negative value). The lines parameter moves the pen the specified number of character-space heights up (a positive value) or down (a negative value). Note that right, left, up, and down are relative to the label direction as shown below.



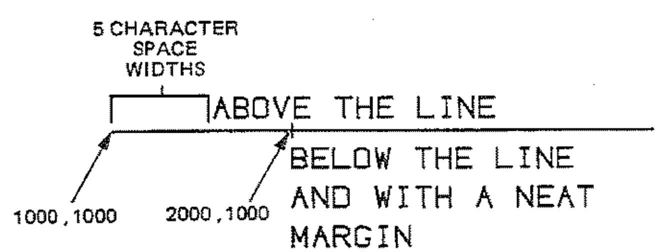


A CP instruction without parameters (CP ;) performs a carriage return and line feed operation, moving one line down and returning to the carriage return point. Refer to The Label Instruction, LB, in this chapter.

A CP instruction which has a nonzero lines parameter shifts the carriage return point up or down by the amount specified. However, a nonzero spaces parameter does not change the carriage-return point.

The use of the CP instruction to produce lettering along a line, but not on top of it and alignment with a left-hand margin is illustrated in the following program. The CP instruction in line 40 moves the label five spaces to the right and slightly above the line. The CP instruction in line 60 moves the label slightly below the line and the CP instruction in line 80 performs a carriage return, line feed to the margin established by the plot instruction in line 40. Inserting carriage return and line feed characters directly into the label string in line 60 causes the same effect as the CP; instruction in line 80.

```
10  ASSIGN @Hp7090 TO 705
20  OUTPUT @Hp7090;"IN;SP1;PA1000,1000,PD;PR3000,0;"
30  OUTPUT @Hp7090;"PUPR-3000,0;"
40  OUTPUT @Hp7090;"CP5,.35;LBABOVE THE LINE"
50  OUTPUT @Hp7090;"PA2000,1000;"
60  OUTPUT @Hp7090;"XT;CP0,-.95;LBBELOW THE LINE"
70  OUTPUT @Hp7090;"LBAND WITH A NEAT"
80  OUTPUT @Hp7090;"CP;LBMARGIN"
90  END
```



Parameter Interaction in Labeling Instructions

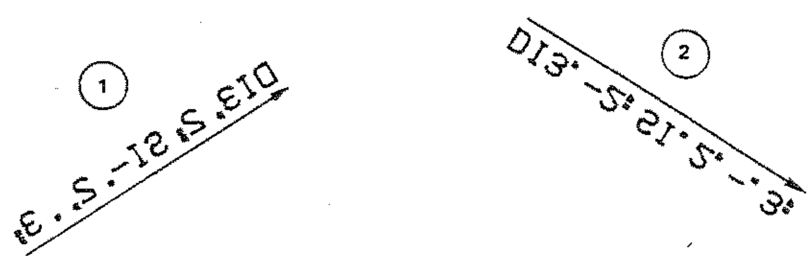
There are three factors which interact and affect the direction and mirroring of labels; the label direction as specified by DI or DR instructions or default direction, the sign of the parameters for the size instructions SI or SR, and the relative positions of P1 and P2. These interactions are complex. This section considers the four possible combinations of DI, DR, SI, and SR and illustrates the effects of various parameters and settings of P1 and P2 on labels.

The labels used in the illustrations are the instructions which cause the direction, size, and mirroring of the label. All descriptions are in terms of the reference coordinate system. An arrow is shown for each label; this arrow is the baseline along which labeling occurs and shows the left-to-right direction that is the standard direction of a label without mirroring. During the course of the illustrations, P1 and P2 are assigned to opposite corners of the plotting area in all possible ways. The values used for X-coordinates of P1 and P2 are 250 and 10 250; the values used for the Y-coordinates of P1 and P2 are 500 and 7500.

Use of DI and SI

When DI and SI instructions are used together, the DI instruction establishes the label's direction and the SI instruction establishes its size. The direction serves as the axis along and about which labels (written with negative SI parameters) are mirrored. Positions of P1 and P2 do not affect the labels. Refer to The Absolute Direction Instruction, DI, and The Absolute Size Instruction, SI.

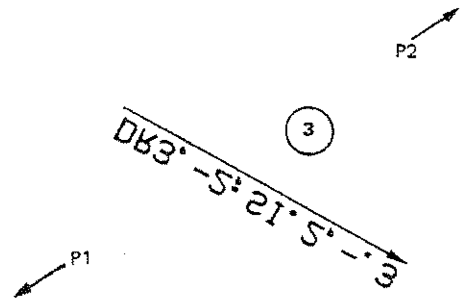
Two examples of mirrored labels are shown below. In the first example, the DI parameters 3,2 place the directional line in the first quadrant. The negative width parameter of the SI instruction mirrors the label in the right-to-left direction. In the second example, the DI parameters 3,−2 place the directional line in the fourth quadrant. The negative height parameter of the SI instruction mirrors the label top-to-bottom.



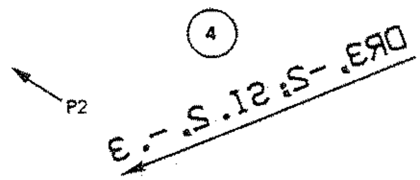
Use of DR and SI

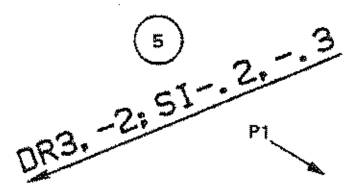
When DR and SI instructions are used together, the label size is determined by the SI instruction and does not change with changes in the settings of P1 and P2. However, changes in the settings of P1 and P2 will affect the label direction. The algebraic differences ($P2_X - P1_X$) and ($P2_Y - P1_Y$) are multiplied by the run and rise parameters of the DR instruction. The resulting parameters, when applied to the reference coordinate system, determine the label baseline. Mirroring about this baseline is determined by the signs of the SI parameters.

In illustration 3, P1 and P2 are at their default settings so the algebraic differences ($P2_X - P1_X$) and ($P2_Y - P1_Y$) are both positive. The DR parameters 3,−2 are used as is and establish the directional line in the fourth quadrant. The negative SI height parameter mirrors the label from top to bottom.



In illustrations 4 and 5, P1 is moved to the lower-right corner and P2 becomes the upper-left corner. Now ($P2_X - P1_X$) is negative. The DR instruction as given is DR3,−2; the run parameter of the DR instruction is multiplied by −1 and the effective DR instruction becomes DR−3,−2 placing the directional line in the third quadrant. The negative SI height parameter mirrors the label from top to bottom. In illustration 5, both SI parameters are negative and the label is mirrored in both directions, making it appear upright.

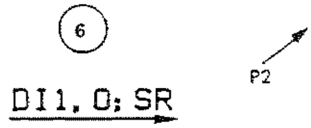


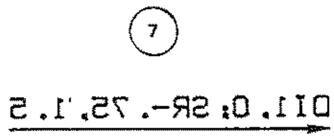


Use of DI and SR

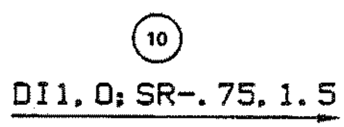
When the DI instruction is used with SR, only the DI instruction affects the directional baseline of labels; changes in the relative positions of P1 and P2 do not affect the baseline. Mirroring about this baseline will occur when either a negative SR width or height parameter with a positive difference ($P2_X - P1_X$) or ($P2_Y - P1_Y$) or a positive SR parameter and a negative difference are present. If respective parameters and differences are both positive or both negative, no mirroring will occur.

Label direction is horizontal for all illustrations in this section. The first three illustrations are drawn with P1 and P2 at their power-on settings. In example 6, the SR; instruction is the same as SR .75, 1.5. Since the parameters are positive, there is no mirroring. In example 7, the negative width parameter causes mirroring right-to-left. In example 8, the negative height parameter causes mirroring top-to-bottom.

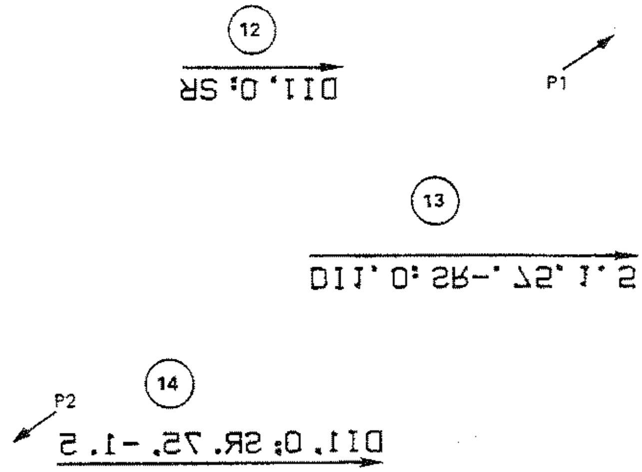




In the next three illustrations, P1 and P2 have been changed so P1 is lower right and P2 is upper left. Hence $(P2_X - P1_X)$ is negative and anything with a positive SR width parameter is mirrored right-to-left, e.g., illustrations 9 and 11. The effect of the negative width parameter in illustration 10 is cancelled by the negative difference $(P2_X - P1_X)$.



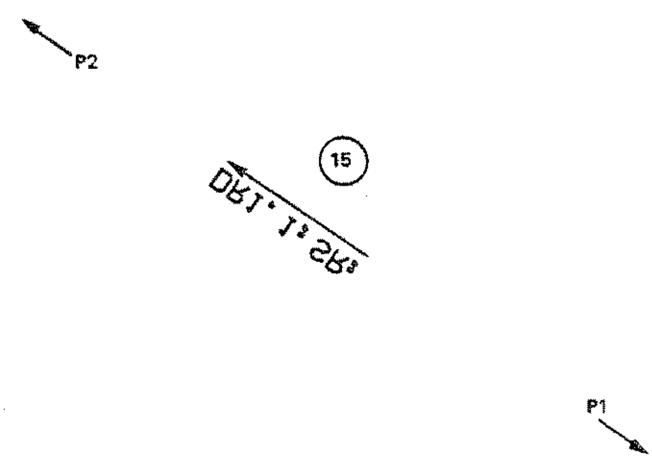
In the next illustrations, P1 and P2 have both been flipped so P1 is upper right and P2 is lower left. Now any positive parameter causes mirroring and any negative parameter cancels mirroring. This can be seen in examples 12, 13, and 14.

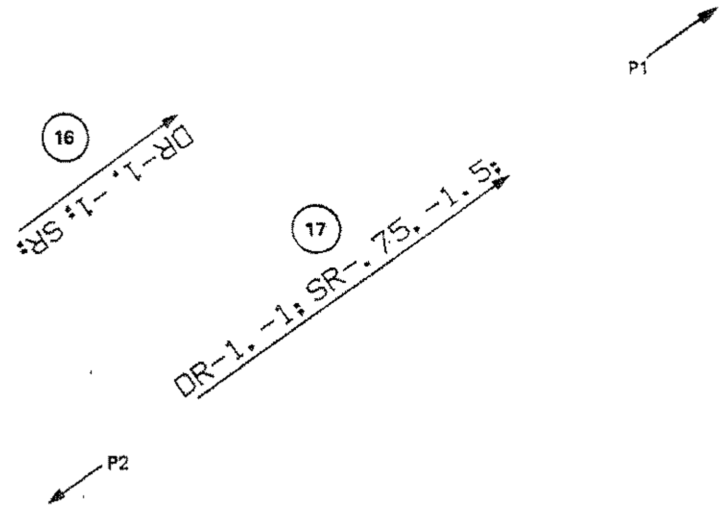


Use of DR and SR

When the DR and SR instructions are used together, interactions are most complex. Using only standard settings of P1 and P2, where P1 is the lower-left corner and P2 is the upper-right corner, will make it easier for you to establish the direction and mirroring of labels you desire. DR parameters interact with the albegraic differences $(P2_X - P1_X)$ and $(P2_Y - P1_Y)$ to establish label direction, and SR parameters interact with these differences to create mirroring. Signs of both parameters and differences are important. A negative sign in either the parameter or the distance will affect both DR and SR instructions. Having both parameter and distance either positive or negative will cause standard direction or no mirroring.

The following examples show the most complex cases, with P1 and P2 in nonstandard locations. Label 15 is drawn with the instructions DR 1 , 1 ; SR in effect, P1 in the lower-right corner and P2 in the upper-left corner. The label baseline is in the second quadrant, not the first, because $(P2_X - P1_X)$ is negative and the DR run parameter is positive. Likewise, the label is mirrored left-to-right because that distance is negative while the parameter is positive. In labels 16 and 17, the label direction baseline is in the third quadrant because both $(P2_X - P1_X)$ and $(P2_Y - P1_Y)$ are negative. Label 16 is mirrored in both directions. (Rotate the manual so the arrow points to +45 degrees to see this more clearly.) In label 17, the label is not mirrored because both parameters and distances are negative.




P1
16
DR-1,-1;SR;
17
DR-1,-1;SR-.75,-1.5;
P2

Chapter 9
Putting the Plotter Instructions to Work

What You'll Learn in This Chapter

In this chapter you'll learn how to put instructions together to develop a plot. The following examples are designed to show you how to integrate many instructions into a complete program, how data might be handled, and how subroutines might be used to program a task which would be common to many plots and used in several programs.

This first program draws a line graph, one of the most common types of plots. While this line graph shows quality control data, line graphs can be used to plot almost any kind of data — factory output, data from laboratory experiments, etc. The concepts of plotting and labeling demonstrated here are applicable in almost any application.

The second example shows a generalized approach to drawing and shading a bar chart.

A variety of allowable separators and terminators have been used in the program listings. In applications where it is important to minimize the number of characters sent over the interface, the spaces between instructions and the semicolon preceding the next mnemonic could and should be omitted. In applications where compatibility with other HP plotters is important, a semicolon or line feed should always be used as the terminator and parameters should be separated by commas.

Problem

Scale, draw, and label an X- and Y-axis in user units and plot test results by type and lot number. Use a different line type for each failure type and place a legend on the graph. The complete program is in the Listing section, following the Solution section.

Solution

Setup and Scaling

For emphasis and readability, the data and title will be drawn with a wide pen. The first step, then, is to set up the pen carousel with a narrow pen in pen stall 1 and a wide pen in pen stall 2. The second step is to set the plotter to known conditions, cancelling any parameters which may have been set in the previously run program. The IN or DF instruction may be used; IN resets P1 and P2; DF does not. IN is used here.

Next, a pen is selected (SP 1) and the scaling for this plot is established. The parameters of the IP instruction determine the location of the scaling points, P1 and P2. In this graph, all data will be plotted within this P1/P2 frame. The points have been chosen to allow room for labels, titles, and margins outside the P1/P2 frame. The scaling statement SC1,23,0,150; assigns user unit values to the scaling points. Although we are plotting test results for 12 lots, we have scaled the X-axis from 1 to 23. This places the center of the P1/P2 frame at an integer location so labels can be easily centered using the LO instruction. The Y-axis is scaled from 0 to 150 since all types of rejections combined should be less than three percent.

Having established our scaling, we shall draw a frame for the data area. This is done by moving to the point 1,0 with the pen up, lowering the pen and drawing to the four corners 23,0; 23,150; 1,150; and 1,0. The coordinates are interpreted as absolute (instead of relative) moves since absolute plotting is established by the IN instruction. The first three program lines with HP-GL instructions are:

```
20 OUTPUT @Hp7090;"IN;SP1;IP1250,1000,9250,6000;"
30 OUTPUT @Hp7090;"SC1,23,0,150;"
40 OUTPUT @Hp7090;"PU1,0PD23,0,23,150,1,150,1,0PU;"
```

NOTE: If compatibility with HP 9872 plotters is desired, PA should be used to begin plotting, and raising and lowering the pen should be controlled with separate PU and PD instructions. In addition, the stricter syntax of HP 9872 plotters would be required. ■

The Axes and Their Labels

We are now ready to draw and label the axes. The label size is set by the absolute size instruction SI.2,.3;. This creates characters which are slightly larger than the default character size specified by the IN instruction. The tick length is established by the tick length instruction TL1.5,0. The resulting ticks will be 1.5% of the horizontal or vertical distances between the scaling points. No negative portion of the tick will be drawn; ticks will be entirely above the X-axis and to the right of the Y-axis.

Axes are commonly drawn using a loop; this program in BASIC uses FOR... NEXT loops. First, we shall draw the X-axis. Let X range from 1 to 23 in steps of 2, representing the 12 lots for which we have data. In the loop we will do four things: move to the integer location on the axis, draw a tick mark, position the pen below the axis, and draw the label. Note that the X-parameter of the plot instruction is a variable. You will need to know how to send a variable between strings of fixed characters. The method will differ from computer to computer; consult your computer's documentation and Plotting with Variables in Chapter 5 of this manual. The XT instruction draws a tick, whether the pen is up or down. The pen is up here so we do not draw the axis line again. You might want to use PD, drawing over the frame line if you want your axis line a bit darker, or you might want to redraw the axis again later with a wide pen.

There are several techniques used here to draw the alphabetic labels. First, so we can use a looping technique, we have placed the labels in a data statement. (At some point, you might want to access data for other lots. If your data were stored together with a date code, you could use a similar technique to read the label and data from some file and properly label your graph for the data you were then plotting.) Secondly, we have used the CI and LO instructions together with HP Series 200 BASIC formatting (using the image specifier "K" to suppress blank characters) to center the label under the tick. The base of the tick mark is the pen position after the tick is drawn. By moving one-half line down and specifying centered strings, both single and double character labels are centered under the tick with enough space so it can be easily read. Finally, the axis title, LOT NUMBER (10000/LOT), is centered and drawn under the axis.

The loop to draw the X-axis and the statements to set character and tick length and to label and title the X-axis are shown next. Note the "K" image specifier; it is used to suppress the sign space in the variable so the label values are centered under the X-axis ticks.

```
50  OUTPUT @Hp7090;"SI.2,.3;TL1.5,0"
60  FOR X=1 TO 23 STEP 2
70     OUTPUT @Hp7090;"PA";X;",0;XT;"
80     READ A
90     OUTPUT @Hp7090 USING "K";"CP0,-.5;LO5;LB";A;"␃"
100 NEXT X
110 DATA 1,2,3,4,5,6,7,8,9,10,11,12
120 OUTPUT @Hp7090;"PA12,-12;LBLOT NUMBER␍␊(10000/LOT)␃"
```

The Y-axis is created in a similar manner, except the loop's index is used for the label value. The Y-axis title is positioned above the axis.

Following the axis routine are the instructions which draw and label the data legend. It is drawn now while the label size is small and the narrow pen is selected.

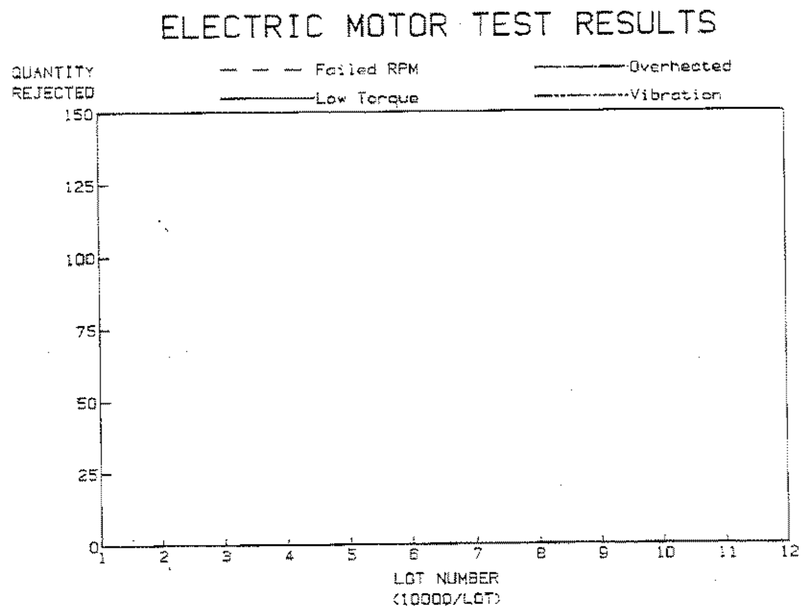
The lines which draw the Y-axis, label it, and draw the legend labels follow:

```
130 FOR Y=0 TO 150 STEP 25
140    OUTPUT @Hp7090;"PA1,";Y;";YT;"
150    OUTPUT @Hp7090;"CP-.5,0;LO8;LB";Y;"␃"
160 NEXT Y
170 OUTPUT @Hp7090;"PA1,165;LO18;LBQUANTITY␍␊REJECTED␃"
180 OUTPUT @Hp7090;"LT2;PA5,165PD8,165PU;LO12;"
190 OUTPUT @Hp7090;"LBFailed RPM␃"
200 OUTPUT @Hp7090;"LT4;PU15,165PD18,165PU;"
210 OUTPUT @Hp7090;"LBOverheated␃"
220 OUTPUT @Hp7090;"LT;PU5,155PD8,155PU;"
230 OUTPUT @Hp7090;"LBLow Torque␃"
240 OUTPUT @Hp7090;"LT6;PU15,155PD18,155PU;"
250 OUTPUT @Hp7090;"LBVibration␃"
```

Because the most important part of the graph is the data and title, we will emphasize these by using a wide pen. The program lines to title the graph using a wide pen follow:

```
260 OUTPUT @Hp7090;"PU12,180;SP2;SI.4,.6;LO5;"
270 OUTPUT @Hp7090;"LBELECTRIC MOTOR TEST RESULTS␃"
```

Here's what the graph looks like so far.



Plotting Your Data

Each of the four data lines on this graph is drawn using a different technique. The first two lines are drawn by plot instructions with parameters included when the program was written. Hence, if the data changes, it will be necessary to change the plot instructions in the program.

The first line (the bottom-most line on the graph) is drawn with pen 2 using a dashed line type. The program takes full advantage of the plotter's relatively free syntax and uses spaces to delineate parameters. Send the character strings to the plotter exactly as shown. Be sure to enter those spaces; if the spaces are removed, the plotter will try to plot one very large number and you won't plot the line.

The second line is also plotted using plot instructions with fixed parameters. These plot instructions use the stricter syntax of the HP 9872 plotter and would be accepted by any HP plotter programmed in HP-GL. The line type consists of long and short dashes. The program lines which plot the two lower lines are:

```
280 OUTPUT @Hp7090;"LT2,6;PA1 23PD3 25 5 18 7 22 9 23;"
290 OUTPUT @Hp7090;"PD11 27 13 27 15 25 17 24 19 28;"
300 OUTPUT @Hp7090;"PD21 27 23 27PU;"
310 OUTPUT @Hp7090;"LT6,8;PA1,45;PD;PA3,50,5,52;"
320 OUTPUT @Hp7090;"PA7,53,9,52;"
330 OUTPUT @Hp7090;"PD11,51,13,55,15,56,17,56,19,58;"
340 OUTPUT @Hp7090;"PD21,58,23,60PU;"
```

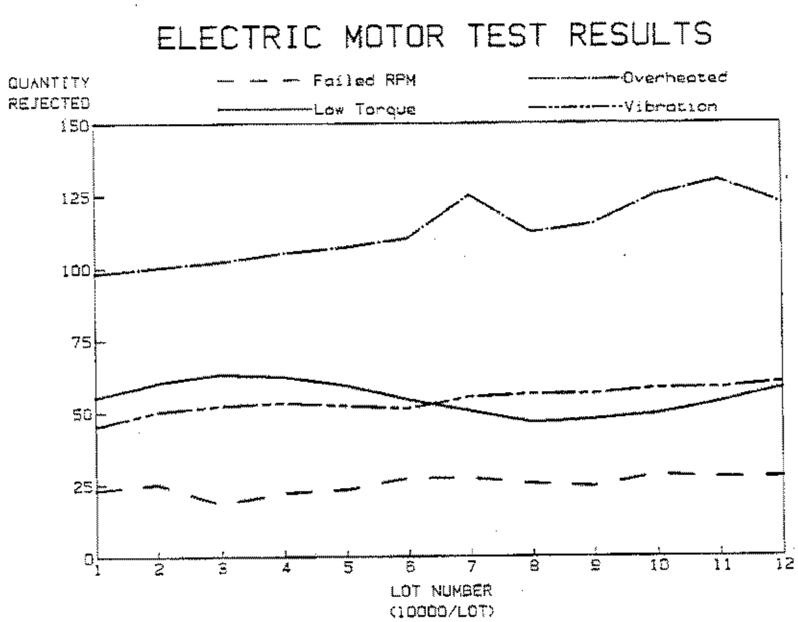
The third line is plotted from data read by the program at execution time using a FOR . . . NEXT loop and a READ statement. This technique would be used to plot a graph that will be replotted often with new data. If the necessary file statements were added, the data could be on a tape or disc file instead of in a DATA statement as shown here. The line type for this line is the default solid line, reverted to by the LT instruction with no parameters. Since we are using variables as plot parameters, you need to be sure they are sent to the plotter with a space between numeric variables. Computers often send a leading and or trailing blank or allow for a sign space before numeric variables. The HP 7090 will treat a blank or comma as a separator between numeric parameters. Know your computer before sending variables with plot instructions. The loop to plot this third line follows:

```
350 OUTPUT @Hp7090;"LT;"
360 FOR X=1 TO 23 STEP 2
370     READ Y
380     OUTPUT @Hp7090;"PA";X;Y;"PD;"
390 NEXT X
400 DATA 55,60,63,62,59,54,50,46,47,49,53,58
```

The last line is drawn using a subroutine. The subroutine is designed to read data that have been stored with a third value for pen control. This third value controls a branch to two different plot statements, one with the pen up and the other with the pen down. In this program, a zero as a pen control parameter results in a pen up move, a 1 causes plotting with the pen down, and 3 signifies the end of the data.

The program lines to change the line type and the subroutine itself are listed here, followed by a reduced version of the completed plot.

```
410 OUTPUT @Hp7090;"LT4,6;"
420 GOSUB 450
430 OUTPUT @Hp7090;"PU25,200;SP0;"
440 STOP
450 REM PLOTTING SUBROUTINE
460 READ X,Y,P
470 IF P=1 THEN OUTPUT @Hp7090;"PD";X;Y
480 IF P=0 THEN OUTPUT @Hp7090;"PU";X;Y
490 IF P=3 THEN 540
500 DATA 1,98,0,3,100,1,5,102,1,7,105,1,9,107,1,11
510 DATA 110,1,13,125,1,15,112,1,17,115,1,19
520 DATA 125,1,21,130,1,23,122,1,0,0,3
530 GOTO 460
540 RETURN
```



Listing

A complete listing of the program follows. This listing contains all the BASIC statements necessary to have this program run on HP Series 200 computers with an HP-IB interface and the plotter set to address 5. In some statements, semicolons or commas are used to ensure that HP-GL instructions will have the necessary separators or no extra spaces. You may need to make changes for your computer's BASIC, or you can use some other programming language and send the strings of HP-GL instructions using your language's output statements and looping techniques.

```
10  ASSIGN @Hp7090 TO 705
20  OUTPUT @Hp7090;"IN;SP1;IP1250,1000,9250,6000;"
30  OUTPUT @Hp7090;"SC1,23,0,150;"
40  OUTPUT @Hp7090;"PU1,0PD23,0,23,150,1,150,1,0PU;"
50  OUTPUT @Hp7090;"SI.2,.3;TL1.5,0"
60  FOR X=1 TO 23 STEP 2
70     OUTPUT @Hp7090;"PA";X;",0;XT;"
80     READ A
90     OUTPUT @Hp7090 USING "K";"CP0,-.5;LO5;LB";A;"␃"
100 NEXT X
110 DATA 1,2,3,4,5,6,7,8,9,10,11,12
120 OUTPUT @Hp7090;"PA12,-12;LBLOT NUMBER␍␊(10000/LOT)␃"
130 FOR Y=0 TO 150 STEP 25
140    OUTPUT @Hp7090;"PA1,";Y;";YT;"
150    OUTPUT @Hp7090;"CP-.5,0;LO8;LB";Y;"␃"
160 NEXT Y
170 OUTPUT @Hp7090;"PA1,165;LO18;LBQUANTITY␍␊REJECTED␃"
180 OUTPUT @Hp7090;"LT2;PA5,165PD8,165PU;LO12;"
190 OUTPUT @Hp7090;"LBFailed RPM␃"
200 OUTPUT @Hp7090;"LT4;PU15,165PD18,165PU;"
210 OUTPUT @Hp7090;"LBOverheated␃"
220 OUTPUT @Hp7090;"LT;PU5,155PD8,155PU;"
230 OUTPUT @Hp7090;"LBLow Torque␃"
240 OUTPUT @Hp7090;"LT6;PU15,155PD18,155PU;"
250 OUTPUT @Hp7090;"LBVibration␃"
260 OUTPUT @Hp7090;"PU12,180;SP2;SI.4,.6;LO5;"
270 OUTPUT @Hp7090;"LBELECTRIC MOTOR TEST RESULTS␃"
280 OUTPUT @Hp7090;"LT2,6;PA1 23PD3 25 5 18 7 22 9 23;"
290 OUTPUT @Hp7090;"PD11 27 13 27 15 25 17 24 19 28;"
300 OUTPUT @Hp7090;"PD21 27 23 27PU;"
310 OUTPUT @Hp7090;"LT6,8;PA1,45;PD;PA3,50,5,52;"
320 OUTPUT @Hp7090;"PA7,53,9,52;"
330 OUTPUT @Hp7090;"PD11,51,13,55,15,56,17,56,19,58;"
340 OUTPUT @Hp7090;"PD21,56,23,60PU;"
350 OUTPUT @Hp7090;"LT;"
360 FOR X=1 TO 23 STEP 2
370    READ Y
380    OUTPUT @Hp7090;"PA";X;Y;"PD;"
390 NEXT X
400 DATA 55,60,63,62,59,54,50,46,47,49,53,58
410 OUTPUT @Hp7090;"LT4,6;"
420 GOSUB 450
430 OUTPUT @Hp7090;"PU25,200;SP0;"
440 STOP
450 REM PLOTTING SUBROUTINE
460 READ X,Y,P
470 IF P=1 THEN OUTPUT @Hp7090;"PD";X;Y
480 IF P=0 THEN OUTPUT @Hp7090;"PU";X;Y
490 IF P=3 THEN 540
500 DATA 1,98,0,3,100,1,5,102,1,7,105,1,9,107,1,11
```

```
510 DATA 110,1,13,125,1,15,112,1,17,115,1,19
520 DATA 125,1,21,130,1,23,122,1,0,0,3
530 GOTO 460
540 RETURN
550 END
```

Filling and Hatching

Two kinds of area fill are commonly used in graphics plots; solid fill and hatching. Solid fill totally covers the area with color, whereas hatching fills the area with evenly spaced parallel lines. If there are lines in two directions at 90 degree angles, we call the hatching "crosshatching." Sometimes a graph will have both narrow and wide hatching or crosshatching; the wide hatching having more space between the lines than the narrow.

Filling a Bar

The following program illustrates solid fill of a bar which represents the lot 3 data for line 1 in the example just given, i.e., Y = 18 (see program line 280). To create an aesthetically pleasing and easily comprehensible bar graph, the bar is centered over the X-data point and is one-half as wide as the distance between data points on the X-axis. The increment variable P determines the spacing between fill lines; a value of P = 10 plotter units is suitable for a wide pen and 5 for a narrow pen.

The basic program should be used when plotting on paper. Notice the pen does not lift during the fill routine; this speeds up filling and prolongs pen-tip life by limiting up/down moves.

The basic program performs the following tasks:

1. Draws around the P1/P2 frame and labels the X-axis.
2. Obtains, in plotter units, the coordinates of the bar's corners.
3. Turns scaling off so plotting is in plotter units. This algorithm can, therefore, be used in any program, and there is no need to recompute the increment P for different scaling in different graphs.
4. Beginning at the X,Ymin value, draws a line to the top of the bar, moves over slightly less than one pen width, and draws to the bottom of the bar.
5. Increments the X-value one pen width and repeats step 4 until the bar is filled.

The lines shown below the program listing should be substituted in the program when plotting on transparency film. In this case, pen velocity is set to 10 cm/s and all fill lines are drawn in the same direction to achieve uniform ink distribution.

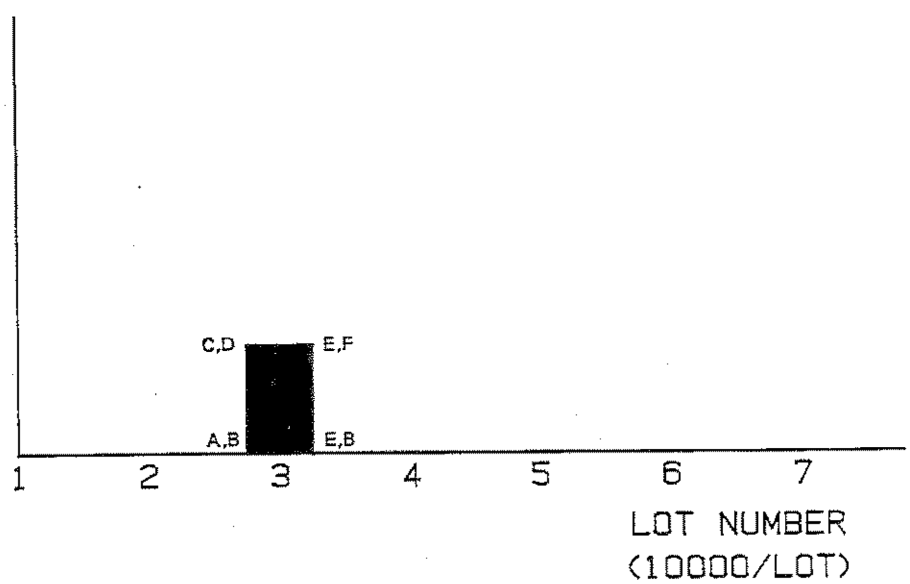
Plotting on Paper

```
10   ASSIGN @Hp7090 TO 705
20   OUTPUT @Hp7090;"IN;SP1;IP1250,1000,9250,6000;"
30   OUTPUT @Hp7090;"SC1,45,0,150;"
40   OUTPUT @Hp7090;"PU1,0PD45,0,45,150,1,150,1,0PU;"
50   OUTPUT @Hp7090;"SI.2,.3;TL1.5,0"
60   FOR X=1 TO 45 STEP 4
70       OUTPUT @Hp7090;"PA";X;",0;"
80       READ A
90       OUTPUT @Hp7090 USING "K";"CP0,-.5;LO5;LB";A;"␃"
100  NEXT X
110  DATA 1,2,3,4,5,6,7,8,9,10,11,12
120  OUTPUT @Hp7090;"PA23,-12;LBLOT NUMBER␍␊␃"
130  OUTPUT @Hp7090;"LB(10000/LOT)␃"
140  OUTPUT @Hp7090;"PA8,0,PD,8,18,10,18,10,0,8,0;PU;"
150  OUTPUT @Hp7090;"OA;"
160  ENTER @Hp7090;A,B
170  OUTPUT @Hp7090;"PA8,18;OA;"
180  ENTER @Hp7090;C,D
190  OUTPUT @Hp7090;"PA10,18;OA;"
200  ENTER @Hp7090;E,F
210  OUTPUT @Hp7090;"PA8,0;SC;"
220  P=5
230  FOR X=A TO E-P STEP 2*P
240      OUTPUT @Hp7090;"PD";X;B;X;D
250      OUTPUT @Hp7090;"PD";X+P;D;X+P;B
260  NEXT X
270  OUTPUT @Hp7090;"PU;SP0;"
280  END
```

Plotting on Transparency Film

```
30   OUTPUT @Hp7090;"VS10;SC1,45,0,150;"
230  FOR X=A TO E STEP P
240     OUTPUT @Hp7090;"PU";X;B;"PD";X;D
250  NEXT X
260  OUTPUT @Hp7090;"PU;SP0;"
270  END
```

Plot Showing Filled Bar



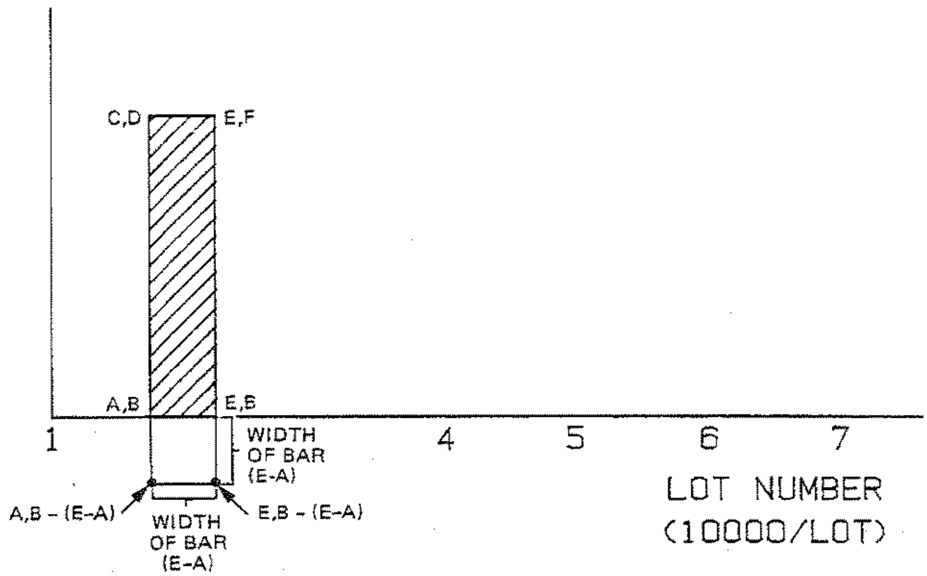
Hatching a Bar

The following program illustrates hatching a bar. In this program, the increment variable P is the distance between hatching lines and determines whether a wide or narrow hatch pattern is drawn. You may want to make further refinements depending on pen width and the bar's dimensions. The locations of the variables are shown and should help you understand the program listing.

The program performs the following tasks:

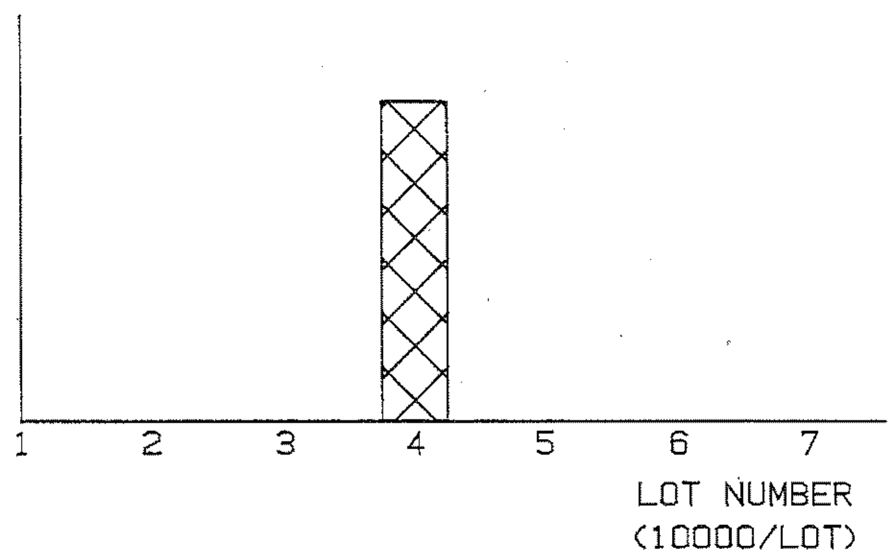
1. Draws around the P1/P2 frame and labels the X-axis.
2. Obtains, in plotter units, the coordinates of the bar's corners.
3. Turns scaling off so plotting is in plotter units. This algorithm can, therefore, be used in any program, and there is no need to recompute the increment P for different scaling in different graphs.
4. Using the output obtained in step 2, sets the window to be the bar we wish to hatch.
5. Beginning the width of the bar below the Ymin value, plots a line at a 45-degree angle to the opposite side of the bar, increments the Y-value and continues the process until the top of the bar is reached.

```
10  ASSIGN @Hp7090 TO 705
20  OUTPUT @Hp7090;"IN;SP1;IP1250,1000,9250,6000;"
30  OUTPUT @Hp7090;"SC1,45,0,150;"
40  OUTPUT @Hp7090;"PU1,0PD45,0,45,150,1,150,1,0PU;"
50  OUTPUT @Hp7090;"SI.2,.3;TL1.5,0"
60  FOR X=1 TO 45 STEP 4
70      OUTPUT @Hp7090;"PA";X;",0;"
80      READ A
90      OUTPUT @Hp7090 USING "K";"CP0,-.5;LO5;LB";A;"␃"
100 NEXT X
110 DATA 1,2,3,4,5,6,7,8,9,10,11,12
120 OUTPUT @Hp7090;"PA23,-12;LBLOT NUMBER␍␊␃"
130 OUTPUT @Hp7090;"LB(10000/LOT)␃"
140 OUTPUT @Hp7090;"PA4,0,PD,4,50,6,50,6,0,4,0;PU;"
150 OUTPUT @Hp7090;"OA;"
160 ENTER @Hp7090;A,B
170 OUTPUT @Hp7090;"PA4,50;OA;"
180 ENTER @Hp7090;C,D
190 OUTPUT @Hp7090;"PA6,50;OA;"
200 ENTER @Hp7090;E,F
210 OUTPUT @Hp7090;"PA4,0;SC;"
220 P=100
230 OUTPUT @Hp7090;"IW";A;B;E;F
240 FOR Y=B-(E-A) TO D STEP P
250     OUTPUT @Hp7090;"PU";A;Y;"PD";E;Y+(E-A)
260 NEXT Y
270 OUTPUT @Hp7090;"PU;SP0;"
280 END
```



To crosshatch a bar, change the program as shown below. The increment variable P is set to 300 plotter units to reduce hatching line density.

```
140 OUTPUT @Hp7090;"PA12,0,PD,12,53,14,53,14,0,12,0;"
170 OUTPUT @Hp7090;"PUPA12,53;OA;"
190 OUTPUT @Hp7090;"PA14,53;OA;"
210 OUTPUT @Hp7090;"PA12,0;SC;"
220 P=300
251     OUTPUT @Hp7090;"PU";E;Y;"PD";A;Y+(E-A)
```



Notes

Chapter 10
General Device Querying

What You'll Learn in This Chapter

Up to this time we have mainly been concerned with the HP-RL and HP-GL instructions which relate to either the recording or plotting functions. Sometimes, however, information of a general nature is required. In this chapter you will learn how to obtain information concerning the current versus the commanded pen position and the most recently received instruction sequence. You will also learn about factors (addressable resolution), model number verification, and the instrument's ability to define its capabilities.

Instructions Covered

OA The Output Actual Position and Pen Status Instruction
OC The Output Commanded Position and Pen Status Instruction
OF The Output Factors Instruction
OI The Output Identification Instruction
OO The Output Options Instruction
OZ The Output Command String Instruction

The Output Actual Position and Pen Status Instruction, OA

USES: The OA instruction is used to output the X- and Y-coordinates and pen status (up or down) associated with the actual pen position. Use it to determine the pen's current position in plotter units. You might use that information to position a label or figure, or determine the parameters of some desired window.

SYNTAX: *OA* terminator

PARAMETERS: None

EXPLANATION: When an OA instruction is received, the pen position and status are output as three numbers in ASCII, separated by commas.

X,Y,P CRLF

where X is the X-coordinate in plotter units,
Y is the Y-coordinate in plotter units,
P is the pen status (0 = pen up, 1 = pen down).

The ranges of the X- and Y-coordinates are the hard-clip limits of the currently selected size of paper.

The Output Commanded Position and Pen Status Instruction, OC

USES: The OC instruction is used to output the X- and Y-coordinates and pen status (up or down) associated with the last valid pen position instruction. Use it to determine the pen's last valid commanded position in plotter units or user units depending on whether scaling is off or on. This instruction is especially useful when the pen is physically at the plotting limits and the pen position does not coincide with the commanded position, or when output in user units is desired.

SYNTAX: *OC* terminator

PARAMETERS: None

EXPLANATION: When an OC instruction is received, the commanded pen position and status are output as three numbers in ASCII, separated by commas.

X,Y,P CRLF

where X is the X-coordinate in plotter units or user units,
Y is the Y-coordinate in plotter units or user units,
P is the pen status (0 = pen up, 1 = pen down).

When scaling is off, X- and Y-coordinates are in plotter units. When scaling is on, X- and Y-coordinates are in user units. Ranges of the X- and Y-coordinates are −32 768 to 32 767 whether scaling is on or off.

The Output Factors Instruction, OF

USES: The OF instruction is used to output the number of plotter units per millimetre in each axis. Use it when software must find the size of a plotter unit.

SYNTAX: *OF* terminator

PARAMETERS: None

EXPLANATION: When an OF instruction is received, the following numbers are always output in ASCII.

40,40 CRLF

These factors indicate there are approximately 40 (actually 40.2) plotter units per millimetre in the X-axis and in the Y-axis. Remember, a plotter unit equals 0.025 mm.

The Output Identification Instruction, OI

USES: The OI instruction is used to output the instrument's model number.

SYNTAX: *OI* terminator

PARAMETERS: None

EXPLANATION: When an OI instruction is received, the following character string is always output in ASCII.

7090A CRLF

The Output Options Instruction, OO

USES: The OO instruction is used to output the instrument's capabilities.

SYNTAX: *OO* terminator

PARAMETERS: None

EXPLANATION: When an OO instruction is received, the HP 7090 will always output the following eight integers in ASCII, separated by commas.

```
0,1,0,0,0,0,0,0   CRLF
  └─INDICATES PEN SELECT CAPABILITY IS INCLUDED.
```

The Output Command String Instruction, OZ

USES: The OZ instruction outputs the character string received immediately prior to execution of OZ. Use it as a program debugging aid.

SYNTAX: *OZ* terminator

PARAMETERS: None

EXPLANATION: When an OZ instruction is executed, OZ and the fourteen characters received immediately before OZ are output when the HP 7090 is instructed to talk. Carriage return and line feed characters are ignored when the string is stored. Therefore, if the string PA0,0␍␊PD; PU␍␊SP4;OZ was received, the HP 7090 would return PA0,0PD; PUSP4;OZ.

Chapter 11
HP-IB Functions

What You'll Learn in This Chapter

In this chapter you will learn which HP-IB functions are implemented in the HP 7090 and how to use the remote/local and serial/parallel polling functions. The HP 7090's reaction to HP-IB bus commands are also explained.

This chapter assumes you have a working knowledge of the HP-IB; however, if you wish to refresh your memory on HP-IB structure, refer to Appendix A.

Instructions Covered

RL The Remote/Local Instruction

HP-IB Function Implementation

The HP-IB conforms to ANSI/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 HP 7090 are as follows:

AH1 Complete acceptor handshake capability
C0 No controller capability
DC1 Complete device clear capability
DT1 Complete device trigger capability
L3 Basic listener, listen only mode, unaddress if MTA
LE0 No extended listener capability
PP0 No parallel poll capability when in listen only mode
PP1 Parallel poll with remote configuration if address > 8
PP2 Parallel poll with local configuration if address < 7
RL0 No remote/local capability (power-up default)
RL1 Complete remote/local capability when enabled by RL1 instruction

NOTE: RL1 cannot be invoked in listen only mode.

SH1 Complete source handshake capability
SR1 Complete service request capability
T6 Basic talker, serial poll, no talk only mode, unaddress if MLA
TE0 No extended talker capability

Device Reaction to HP-IB Bus Commands

The computer can set all devices on the HP-IB 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 will obey SDC only if they are addressed to listen, whereas DCL clears all devices on the bus. The interface clear command, IFC, is used by the computer to override all bus operations and return the bus to a known quiescent state. The HP 7090's reaction to bus commands is as follows:

- Upon receipt of either a DCL or SDC command, the HP 7090 resets the I/O to begin accepting a new instruction, and disables any current output. Any partially parsed HP-RL or HP-GL instruction will be lost. The IFC command resets both talker and listener functions to their respective idle states (TIDS and LIDS) per IEEE-488 specifications.

  The commands DCL, SDC, and IFC do not reset parameters to their default values. They are not the same as the HP-GL instructions DF or IN.

- The group execute trigger, GET, is recognized only when the HP 7090 is in either its data acquisition or recorder mode and external trigger mode is selected. Otherwise, the GET command is ignored.

- The end message is sent, via the EOI line, when the last byte of data from the HP 7090 is output to the computer.

The Remote/Local Instruction, RL

USES: The RL instruction provides the means to disable the HP 7090's front-panel controls. Use it to prevent inadvertent interruption of I/O measurements by someone touching the front-panel controls.

SYNTAX: *RL* state terminator
or
RL terminator

Parameter	Format	Range	Default
state	integer	0 = local state 1 = remote state	0

EXPLANATION: When the HP 7090 receives an RL1 instruction it enters the remote state. In this state, the front-panel is inactivated and the word **REMOTE** is displayed on the LCD. Pressing the **SHIFT/LOCAL** buttons will return the HP 7090 to the normal (local) state of operation. The **REMOTE** message is cleared and the front panel is reactivated. However, the next time the computer sends an MLA (my listen address) message, the HP 7090 is put back into the remote state.

NOTE: The MLA message is sent each time the computer begins a PRINT or OUTPUT statement. ■

With RL1 invoked, the HP Series 200 HP-IB commands "LOCAL 7" and "LOCAL LOCKOUT 7" can be used. Executing LOCAL LOCKOUT 7 will inactivate the entire front panel, including the SHIFT/LOCAL buttons. Executing LOCAL 7 will reactivate the front panel, but prevent the HP 7090 from responding to HP-RL or HP-GL instructions. Executing REMOTE 7 will counteract the effects of LOCAL 7.

An RL instruction with no parameter (RL ;), or a parameter of zero (RL 0 ;), returns the HP 7090 to the local state (default). The HP 7090 remains in the local state until RL1 is again executed. In the local state the HP 7090 will respond to either front-panel or I/O instructions. LOCAL 7 and LOCAL LOCKOUT 7 do not affect the local state. However, they will be in effect for any subsequent RL1 instruction.

An out-of-range parameter sets error 3, and the instruction is ignored.

Serial and Parallel Polling

Polling is the process used by the computer to determine which device on the HP-IB bus has initiated a require service message. The conditions which will cause the require service message to be sent to the computer are defined by the input mask instruction, IM, in Chapter 12.

The Serial Poll

A serial poll enables the computer to learn the status or condition of devices on the bus. It is commonly used by the computer to determine who is requesting service.

The serial poll is so named because the computer polls devices one at a time rather than all at once as in the parallel poll. The HP 7090 will respond to a serial poll by sending the status byte selected by the XS instruction (see Chapter 12). The S-mask parameter of the IM instruction is used to specify which status byte conditions will send the service request message. Unless the user changes the S-mask value from the default setting of 0, the HP 7090 will never send a service request. When any of the conditions designated by the S-mask are true, the service request message is sent concurrently with setting bit position 6 of the status byte. Executing a serial poll resets bit 6 to 0 and arms the HP 7090 so it can generate another service request. It should be noted that the latched bits of the status byte are not reset by a serial poll. Therefore, an OS or QS instruction should be included in the serial poll service routine to reset the latched bits. This will prevent generating a service request on old status conditions.

A computer must issue special commands to initiate and terminate a serial poll. During a serial poll, a device must be instructed to talk and the computer to listen. Therefore, a serial poll cannot be executed when the HP 7090 is in listen only mode.

The Parallel Poll

The parallel poll is the fastest method of determining which device has requested service, since all devices (with this capability) can be polled in one operation. The devices respond with only one bit of status, but if the response is affirmative, a serial poll can be conducted to obtain the device's specific status.

The P-mask parameter of the input mask instruction, IM, is used to specify the staus byte conditions that will cause the HP 7090 to respond affirmatively to a parallel poll. The response to a parallel poll is limited to setting the appropriate data line to a logical 1. If the HP 7090's address is 0 through 7 the data line used is determined by the address as shown in the following table. With address settings > 7, a response will not be sent unless the HP 7090 has been remotely configured to do so by the computer. In this case, the response is sent on the data line specified by the computer.

Plotter 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	Plotter Preset Address
6	1	2	
7	0	1	

It is important to remember that the HP 7090 will not send a logical 1 unless the P-mask bit value has been changed from the default value of 0 and some condition included in the new P-mask value is true. The HP 7090 does not respond to a parallel poll in listen only mode.

Positive responses to parallel polls will continue to occur until all bits of the status byte included in the P-mask value have been reset to 0.

To execute a parallel poll, the computer sets the ATN and EOI lines to 1. The computer reads the eight data lines, and determines from these lines which instrument on the bus is requesting service. The computer then sends the parallel poll disable command. Not all computers have parallel poll capability.

Notes

Chapter 12

Error and Status Reporting

What You'll Learn in This Chapter

In this chapter you will learn how to select the recorder or plotter status byte for use as the HP-IB status byte and how to mask the conditions that will generate an error message, a service request message, or a parallel poll response. You will also learn how to locate syntax errors that may have caused the error message. Finally, a complete program example is given to illustrate how to use the HP 7090's status reporting capabilities in combination.

Instructions Covered

IM The Input Mask Instruction
OE The Output Error Instruction
OS The Output Plotter Status Instruction
OY The Locate Syntax Error Instruction
QC The Query Channel Status Instruction
QS The Query Recorder Status Instruction
XS The Set Plotter or Recorder Status Instruction

The Set Plotter or Recorder Status Instruction, XS

USES: The XS instruction selects either the plotter status byte (OS) or the recorder status byte (QS) for use as the HP-IB status byte. Use it to permit generating service requests or positive parallel poll responses on either plotter or recorder conditions.

SYNTAX: *XS* status type terminator
or
XS terminator

Parameter	Format	Range	Default
status type	integer	0 = plotter status byte 1 = recorder status byte	0

EXPLANATION: Upon power-up or after an IN instruction, the plotter status byte is the HP-IB status byte that is output during an HP-IB serial poll. An XS instruction with no parameter (XS;), or a parameter of zero (XS 0;), also selects the plotter status byte for use as the HP-IB status byte.

An XS instruction with a parameter of one (XS 1;) selects the recorder status byte for use as the HP-IB status byte.

An out-of-range parameter sets error 3, and the instruction is ignored.

The Output Plotter Status Instruction, OS

USES: The OS instruction is used to output the decimal equivalent of the plotter status byte in ASCII. Use it to obtain the current status of various plotter functions.

SYNTAX: *OS* terminator

PARAMETERS: None

EXPLANATION: When an OS instruction is received, the internal eight-bit plotter status byte is converted to a number between 0 and 255. This decimal value is output in ASCII in the form:

plotter status CRLF

NOTE: The number should be read into a numeric variable to allow bit checking using the HP Series 200 BIT function. ■

The meaning of each plotter status bit is defined in the following table. The bits which remain set until cleared are identified by the term "latched." The latched bits can be cleared only by the indicated methods. The other bits are constantly updated to reflect current conditions.

Bit Value	Bit Position	Meaning
1	0	Pen is down.
2	1	P1 or P2 changed (latched): cleared by reading OP output or by executing IN instruction, but not by a serial poll.
4	2	Not used.
8	3	Initialized (latched): cleared by reading OS output, but not by a serial poll.
16	4	Ready for data; pinch wheels are down and servo is operational.
32	5	Error (latched): cleared by reading OE output or by executing IN instruction, but not be a serial poll.
64	6	Require service (SRQ) message set (latched); always 0 for OS, but 0 or 1 for HP-IB status byte: cleared by a serial poll.
128	7	Next byte available for output: cleared when output or by executing IN instruction (always 0 for QS; 0 or 1 for HP-IB status byte).

Upon power up with the pinch wheels down, the status is decimal 24, the sum of 8 (initialized) and 16 (ready for data).

When default conditions are in effect, this plotter status byte is the HP-IB status byte that is output during an HP-IB serial poll. The HP-IB status byte is masked by the current S- and P-mask parameters of the IM instruction to define which of the above status conditions will generate a service request and/or a positive response to a parallel poll.

The XS instruction can substitute the QS recorder status byte in place of the plotter status byte for use as the HP-IB status byte.

The Query Recorder Status Instruction, QS

USES: The QS instruction is used to output the decimal equivalent of the recorder status byte in ASCII. Use it to obtain the current status of various recorder functions.

SYNTAX: *QS* terminator

PARAMETERS: None

EXPLANATION: When a QS instruction is received, the internal eight-bit recorder status byte is converted to a number between 0 and 255. This decimal value is output in ASCII in the form:

recorder status CRLF

NOTE: The number should be read into a numeric variable to allow bit checking using the HP Series 200 BIT function. ■

The meaning of each recorder status bit is defined in the following table. The bits which remain set until cleared are identified by the term "latched." The latched bits can be cleared only by the indicated methods. The other bits are constantly updated to reflect current conditions. Note that a QS instruction can be executed without terminating an in-process direct or buffered measurement.

Bit Value	Bit Position	Meaning
1	0	Trigger has been found (latched): cleared by reading QS output or by executing IN instruction, but not by a serial poll.
2	1	Channel overload has occurred (latched): cleared by reading QS output or by executing IN instruction, but not by a serial poll.
4	2	Next sample is ready: cleared by output of last sample byte or by IN instruction.
8	3	Sample overrun has occurred (latched: cleared by reading QS output or by executing IN instruction, but not by a serial poll.
16	4	Ready for data; pinch wheels are down and servo is operational.
32	5	Error (latched): cleared by reading OE output or by executing IN instruction, but not be a serial poll.
64	6	Require service (SRQ) message set (latched); always 0 for QS, but 0 or 1 for HP-IB status byte: cleared by a serial poll.
128	7	Next byte available for output: cleared when output or by executing IN instruction (always 0 for QS; 0 or 1 for HP-IB status byte).

Upon power up with the pinch wheels down, the status is 16 (ready for data).

When default conditions are in effect, the plotter status byte (OS) is the HP-IB status byte that is output during an HP-IB serial poll. However, the XS instruction can substitute the recorder status byte in place of the plotter status byte for use as the HP-IB status byte.

The HP-IB status byte is masked by the current S- and P-mask parameters of the IM instruction to define which status byte conditions (plotter or recorder) will generate a service request and/or a positive response to a parallel poll.

The Query Channel Status Instruction, QC

USES: The QC instruction is used to output the decimal equivalent of the channel status byte in ASCII. Use it to obtain the current status of the input signal on each channel and to determine if the trigger is out-of-range or a measurement is in progress.

SYNTAX: *QC* terminator

PARAMETERS: None

EXPLANATION: When a QC instruction is received, the channel status byte is converted to a number between 0 and 255. This decimal value is output in ASCII in the form:

channel status CRLF

NOTE: The number should be read into a numeric variable to allow bit checking using the HP Series 200 BIT function. ■

The channel status bits defined in the following table are constantly updated to reflect current conditions. These status conditions can be obtained without terminating an in-process direct or buffered measurement. Any time an overload condition occurs on any channel, bit 1 of the recorder status byte is set and remains latched so the condition can be reported via a serial poll.

Bit Value	Bit Position	Meaning
1	0	Channel 1 overload.
2	1	Channel 2 overload.
4	2	Channel 3 overload.
8	3	Channel 1 offset invalid (out-of-range).
16	4	Channel 2 offset invalid (out-of-range).
32	5	Channel 3 offset invalid (out-of-range).
64	6	Trigger out-of-range.
128	7	Measurement in progress (direct or buffered only).

The Output Error Instruction, OE

USES: The OE instruction is used to obtain the number of the currently set HP-RL or HP-GL error (if any). Use it when debugging programs to determine the type of error that has occurred.

SYNTAX: *OE* terminator

PARAMETERS: None

EXPLANATION: When an OE instruction is received, the HP 7090 converts the first HP-RL or HP-GL error to a positive integer in ASCII, which is output in the form:

error number CRLF

The error number is defined as follows:

Error Number	Meaning
0	No error.
1	Instruction not recognized.
2	Wrong number of parameters.
3	Parameter out-of-range.
4	Not used.
5	Unknown character set.
6	Position overflow.
7	Not used.
8	Vector move requested while pinch wheels raised or servo is inoperative.

After the carriage return has been sent, bit position 5 of the plotter and recorder status bytes are cleared (if set) and the ERROR message (if displayed) is turned off.

You should note that most of the errors are related to instruction syntax, and that only the first error encountered is output. Therefore, placing an OE instruction at the end of a program allows the program to be run repeatedly until all syntax errors are located and corrected. Refer to the OY instruction for additional information on locating syntax errors.

The Locate Syntax Error Instruction, OY

USES: The OY instruction outputs the character string received immediately prior to an HP-RL or HP-GL error. Use it as a program debugging aid to locate syntax errors.

SYNTAX: *OY* terminator

PARAMETERS: None

EXPLANATION: When an error is sensed, the sixteen characters preceding the error are stored in memory. Carriage return and line feed characters are ignored when the string is stored. For example, if the string PA 100,100␍␊PD;PU␍␊S4 was received prior to the error, executing an OY instruction would return PA 100,100PD;PUS4.

Executing an OY instruction and reading the string into the computer does not clear the string from memory. Therefore, if multiple errors are received before the string is cleared, the string will contain the 16 characters preceding the first error. The string is cleared by executing an OE or an IN instruction.

The Input Mask Instruction, IM

USES: The IM instruction controls the conditions under which HP-RL and HP-GL error status is reported, the conditions that can cause an HP-IB service request message, and the conditions that can cause a positive response to an HP-IB parallel poll. Refer to Serial and Parallel Polling in Chapter 11.

SYNTAX: *IM* E-mask (, S-mask (, P-mask)) terminator
or
IM terminator

Parameter	Format	Range	Default
E-mask	integer	0 to 255	223
S-mask	integer	0 to 255	0
P-mask	integer	0 to 255	0

EXPLANATION: The E-mask can be the sum of any combination of the bit values shown in the following table. When an HP-RL or HP-GL error occurs, the error number is tested against the specified E-mask bits to determine if the error bit (bit 5) of the selected status byte (plotter or recorder) is to be set and the ERROR display turned on. If the corresponding E-mask bit is not specified, there is no way to ever determine if that error occurred.

E-Mask Bit Value	Bit	Error Number	Meaning
1	0	1	Instruction not recognized.
2	1	2	Wrong number of parameters.
4	2	3	Parameter out-of-range.
8	3	4	Not used.
16	4	5	Unknown character set.
32	5	6	Position overflow.
64	6	7	Not used.
128	7	8	Vector move requested while pinch wheels raised or servo is inoperative.

The default E-mask value of 223 (128+64+16+8+4+2+1) specifies that all errors except error 6 will set the error bit in the selected status byte and turn on the ERROR display whenever they occur. Error 6 will not set the error bit or turn on the ERROR display if it occurs, since it is not included in the E-mask value. Errors 4 and 7 never occur, so setting the E-mask to 151 defines the same conditions as the default value 223.

The S-mask can be the sum of any combination of the bit values of either the plotter or recorder status byte. When a bit of the selected status byte is set and the corresponding bit value is included in the S-mask, bit 6 of the status byte is set and the service request message is sent. Refer to the OS (plotter) and QS (recorder) instructions for the meaning of each bit value.

For example, an S-mask value of 2 specifies that moving P1/P2 or a channel overload condition will cause a service request message to be sent, depending on whether the plotter or recorder status byte is selected.

The P-mask can be the sum of any combination of the bit values of either the plotter of recorder status byte. It specifies which of the status byte conditions will result in a logical 1 response to an HP-IB parallel poll. Refer to the OS (plotter) and QS (recorder) instructions for the meaning of each bit value.

For example, a P-mask value of 48 specifies that only bits 4 and 5 (16 + 32) of the status byte can cause the HP 7090 to respond to a parallel poll with a logical 1 on the appropriate data line.

The HP 7090, when set to default values or initialized, automatically sets the E-mask to 223, the S-mask to 0, and the P-mask to 0. An IM instruction without parameters (IM ;) or with invalid parameters also sets the masks to the default values 223 , 0 , 0.

Status Reporting Example

The following program shows how to combine the HP 7090 status reporting capabilities with the interrupt and polling capabilities of the HP Series 200 computers. In this program, status reporting is structured around a buffered measurement of channels 1 and 2 versus time. The IM instruction specifies service requests are to be generated when the trigger is found or a channel overload occurs.

The program interrupts on these service requests and branches to a service routine which performs a parallel poll to determine which device on the bus needs service. If the HP 7090 needs service, a serial poll is performed to read in the recorder status byte. Bit checking is performed on the status byte and either "TRIGGER FOUND" is displayed to indicate the buffers are filling or the program branches to the Channel subroutine to determine which channel has an overload condition.

The channel subroutine uses the QC instruction to read in the channel status byte. If the bit checking determines the overload is on channel 1, "MEASUREMENT TERMINATED" is displayed and the program is immediately terminated. If channel 2 has an overload, "Channel 2 Overload" is displayed and the program returns to the serial subroutine. The QS instruction is then executed to clear the latched bits in the recorder status byte. The SRQ interrupts are again enabled and the buffers are allowed to continue filling.

Notice the two idle loops in lines 220 through 270. The first loop will not be exited until the trigger is found. The second loop ensures that service requests can be generated on overloads until the buffers are completely filled. "BUFFERS FILLED" is displayed when the second loop is exited. The buffer contents are then plotted to provide a hardcopy record of the measurement.

```
10    !This program takes a buffered measurement of channels 1 and 2 versus
20    !time.  Service requests are generated when the trigger is found or a
30    !channel overload occurs.  The program interrupts on SRQ, and parallel
40    !polling is used to determine which device needs service.  Serial
50    !polling is used to determine the type of service required.  "TRIGGER
60    !FOUND" is displayed when the buffers start filling and "BUFFERS FILLED"
70    !is displayed when the measurement is completed.  The buffer contents
80    !are then plotted.  An overload on channel 1 immediately terminates
90    !the measurement, but an overload on channel 2 only causes "Channel 2
100   !Overload" to be displayed.
110   PRINTER IS 1
120   PPOLL CONFIGURE 705;12 !PPOLL response on line 4 with logic 1
130   ON INTR 7,15 GOSUB Serv_rtn
140   ENABLE INTR 7;2 !Enables SRQ interrupts
150   OUTPUT 705;"IN;XS1;IM223,3,3;"  !SRQ on trigger or channel overload
160   OUTPUT 705;"RE13;"  !Sets channels 1 and 2 versus time
170   OUTPUT 705;"IR10,1;"  !Sets RANGE for channels 1 and 2
180   OUTPUT 705;"TB10,0;"  !Sets TOTAL TIME to 10 seconds
190   OUTPUT 705;"TM2,0;"  !Sets TRIGGER MODE to INTERNAL, ABOVE LEVEL
200   OUTPUT 705;"TA3,0;"  !Sets TRIGGER LEVEL to 3 volts.
210   OUTPUT 705;"MS1;"  !Invokes buffered recording mode
220   IF BIT(Status,0) THEN 240 !Idle loop until trigger is found
230   GOTO 220
240   FOR N=1 TO 10 !Idle loop while buffers are filling
250   WAIT 1
260   IF N=10 THEN PRINT "BUFFERS FILLED"
270   NEXT N
280   OUTPUT 705;"PL;"  !Plot buffer contents
290   STOP
300   !
310   !
320 Serv_rtn:   Device=PPOLL(7)
330             IF BIT(Device,4) THEN GOSUB Serial  !HP 7090 needs service
340             RETURN
350   !
360   !
370 Serial:     Status=SPOLL(705)
380             IF BIT(Status,0) THEN PRINT "TRIGGER FOUND"
390             IF BIT(Status,1) THEN GOSUB Channel
400             OUTPUT 705;"QS;"
410             ENTER 705;Reset !Clears latched bits in recorder status byte
420             ENABLE INTR 7;2
430             RETURN
440   !
450   !
460 Channel:    OUTPUT 705;"QC;"
470             ENTER 705;Overload
480             IF BIT(Overload,0) THEN
490             OUTPUT 705;"MT;"
500             PRINT "MEASUREMENT TERMINATED"
510             GOTO 570
520             END IF
530             IF BIT(Overload,1) THEN PRINT "Channel 2 Overload"
540             RETURN
550   !
560   !
570   END
```

Chapter 13
Rotation

What You'll Learn in This Chapter

In this chapter you will learn how to rotate the coordinate systems by 90 degrees. This feature permits recordings and plots to be oriented so they can be read horizontally, regardless of how they are stored in a binder.

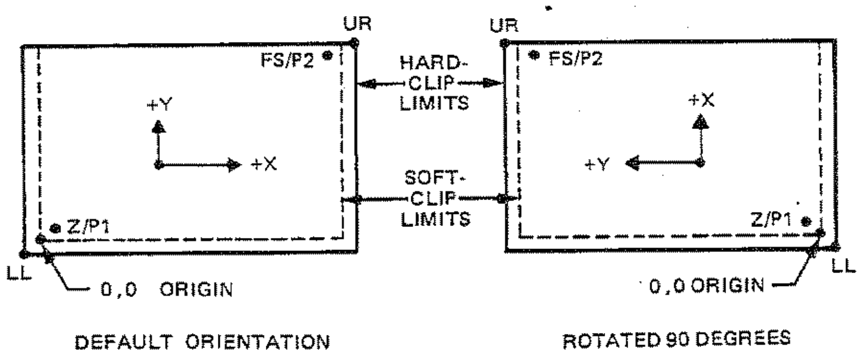
Instructions Covered

RO The Rotate Coordinate System Instruction

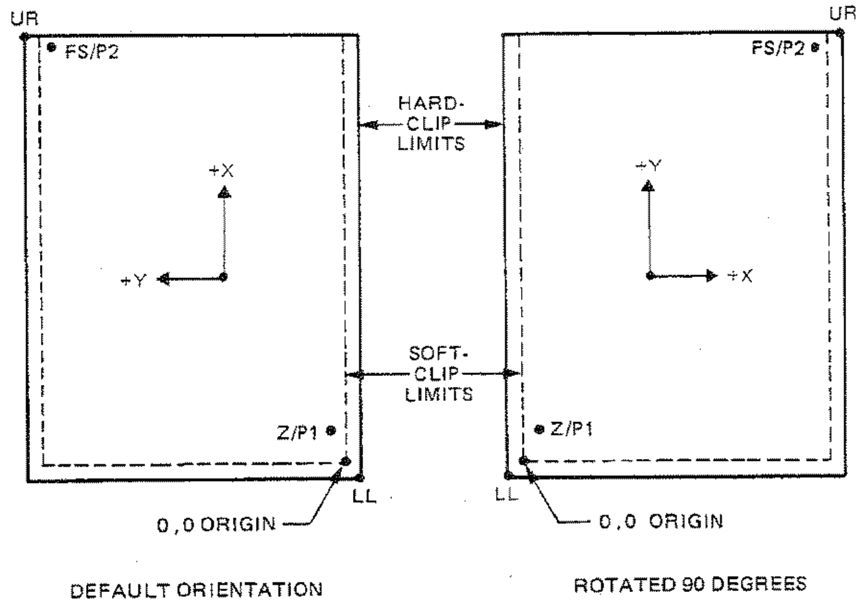
Rotating the Coordinate Systems

The reference coordinate system, recording area, and plotting area are normally oriented so the X-axis is plotted along the long length of the paper. However, the coordinate axes can be rotated 90 degrees so the X-axis is plotted along the short length of the paper. The following illustration shows the default and rotated axes orientations for A/A4 size and B/A3 size paper.

Front-Panel Rotation on A/A4 Size Paper



Front-Panel Rotation on B/A3 Size Paper



The rotate function can be invoked from the front panel, using the **FAST+SHIFT** button combination, or programmatically, using the HP-GL instruction, RO. The only difference between the two methods is the rotated locations of the soft-clip limits and both sets of scaling points (zero/full scale and P1/P2). The front-panel rotation automatically relocates the soft-clip limits and both sets of scaling points to the plotter unit coordinate locations shown in the following tables. These values correspond with the locations shown in the illustration. The RO instruction, however, allows both sets of scaling points to retain their coordinate values, and they may therefore be rotated outside the hard-clip limits. Refer to the RO instruction explanation for complete details.

Rotated Hard-Clip and Default Soft-Clip Limits

Paper Size	Hard-Clip Limits		Soft-Clip Limits	
	X_{LL}, Y_{LL}	X_{UR}, Y_{UR}	X_{LL}, Y_{LL}	X_{UR}, Y_{UR}
A 8.5 × 11 in.	−100, −333	7987, 10703	0, 0	7987, 10370
B 11 × 17 in.	−333, −475	10703, 16260	0, 0	10370, 16260
A4 210 × 297 mm	−100, −322	7785, 11400	0, 0	7785, 11078
A3 297 × 420 mm	−322, −525	11400, 15762	0, 0	11078, 15762

Paper Size	Zero and P1	Full Scale and P2
	$Z_X, Z_Y (P1_X, P1_Y)$	$FS_X, FS_Y (P2_X, P2_Y)$
A 8.5 × 11 in.	447, 160	7682, 10 210
B 11 × 17 in.	160, 865	10 210, 16 140
A4 210 × 297 mm	348, 514	7583, 10 564
A3 297 × 420 mm	514, 325	10 564, 15 600

Rotated Default Scaling Point Locations

The Rotate Coordinate System Instruction, RO

USES: The RO instruction programmatically rotates the coordinate systems 90 degrees. Use the instruction to orient plots vertically or horizontally, regardless of whether the paper is loaded with the short or long dimension along the pen axis.

SYNTAX: *RO* angle terminator
or
RO terminator

Parameter	Format	Range	Default
angle	integer	0 = normal 90 = rotated	0

EXPLANATION: The only allowable parameters are 0 and 90. An RO 90 instruction rotates the current coordinate systems 90 degrees from its default orientation as shown in the following diagrams for A/A4 and B/A3 paper sizes. Rotations are not cumulative, and the rotate function can only be toggled on and off. An RO 0 instruction is the same as an RO instruction without parameters (RO;) and turns off the rotate function.

When an RO 90 instruction is executed, both sets of scaling points retain their current coordinate values and may therefore be rotated outside the hard-clip limits. The current input window (soft-clip limits) is also rotated, and any portion that is rotated outside of the hard-clip limits is clipped to the hard-clip limits. The size of the clipped input window can be determined by executing the OW instruction. The input window, zero and full scale, and P1 and P2 can be defaulted to their rotated default coordinate values using the instructions IW, IZ, and IP without parameters. Refer to the tables on the previous page for the rotated default coordinate values.

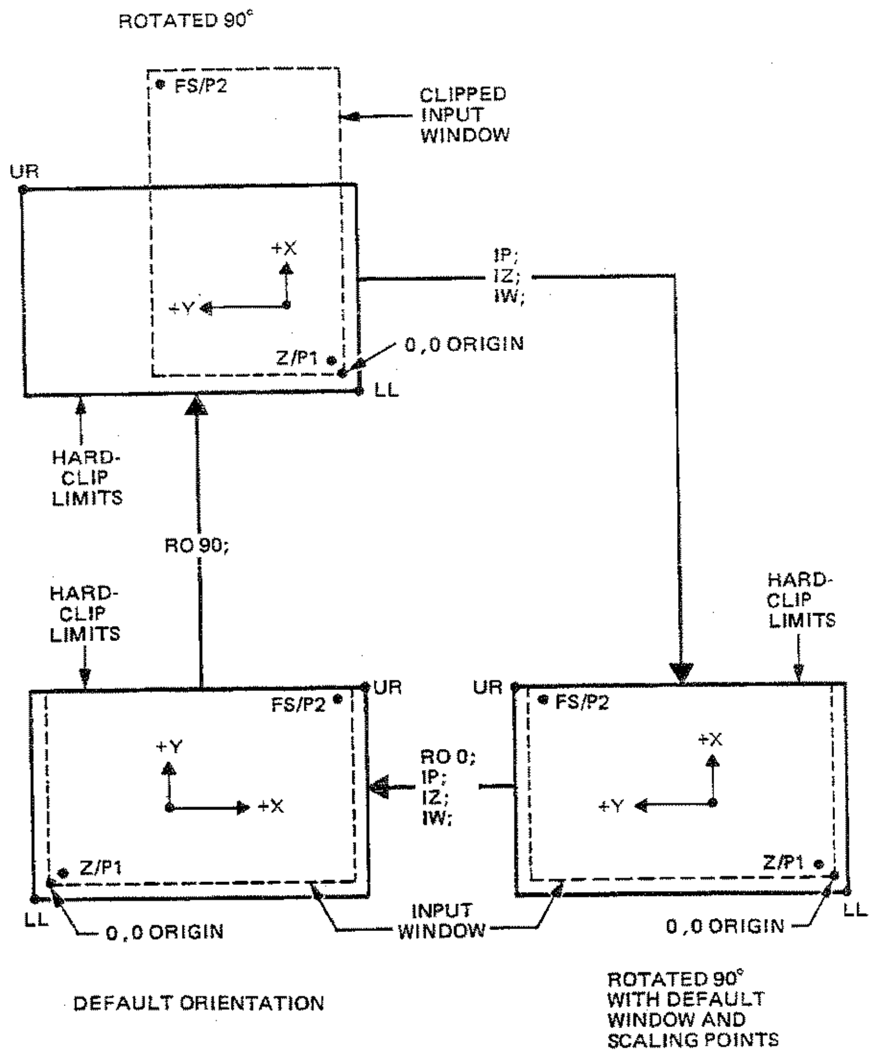
The 0, 0 origin moves when the coordinate system is rotated, but the physical size and location of the hard-clip limits are not affected. However, the defined lower-left (LL) and upper-right (UR) corners of the hard-clip limits are rotated to maintain the same relationship with respect to the 0, 0 origin. The coordinate values are determined by paper size and the state of the rotate function. The current values for LL and UR can be obtained by executing the OH instruction.

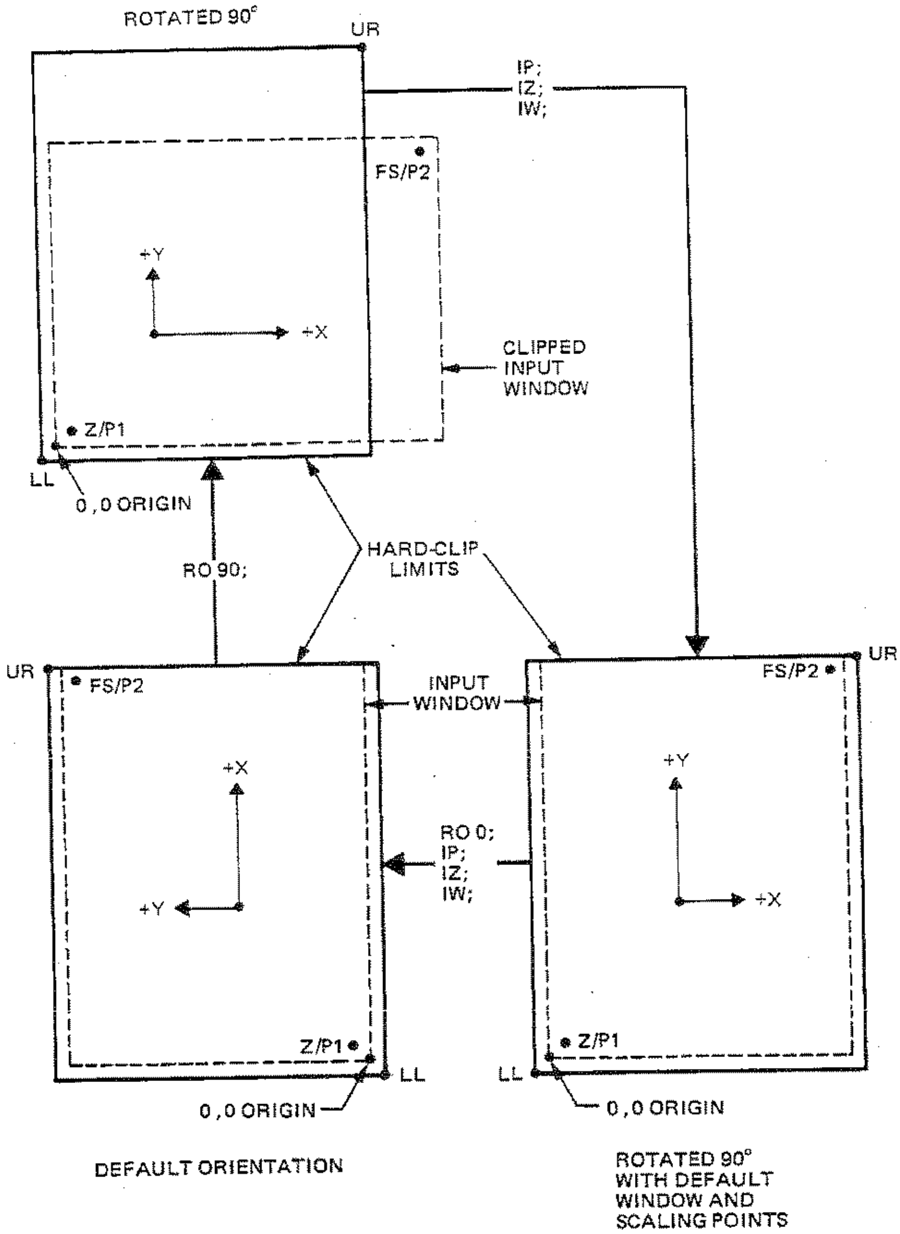
When the coordinate system is rotated, the logical pen position is changed to correspond with the current physical pen position. The coordinate values of the new logical pen position can be obtained by executing either an OA or OC instruction after the rotate instruction is executed.

Specifying parameters other than 0 or 90 sets error 3 and the instruction is ignored.

Remember, you can also turn rotation on and off via the front panel. Hold the FAST button down and press the SHIFT button to turn on rotation; press the buttons again to turn it off. Unlike the RO instruction, the front-panel rotation automatically defaults the input window and both sets of scaling points.

Rotation on A/A4 Size Paper




ROTATED 90°
UR
IP;
IZ;
IW;
FS/P2
+Y
+X
CLIPPED
INPUT
WINDOW
Z/P1
LL
0,0 ORIGIN
RO 90;
HARD-CLIP
LIMITS
UR
FS/P2
+X
+Y
Z/P1
LL
0,0 ORIGIN
INPUT
WINDOW
RO 0;
IP;
IZ;
IW;
UR
FS/P2
+Y
+X
Z/P1
LL
0,0 ORIGIN
DEFAULT ORIENTATION
ROTATED 90°
WITH DEFAULT
WINDOW AND
SCALING POINTS

Notes

Appendix A
An HP-IB Overview

HP-IB Terms and Concepts

The HP 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 ANSI/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 may vary from device to device.
5. **Operator** — The person that 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.

Interface Bus Concepts

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

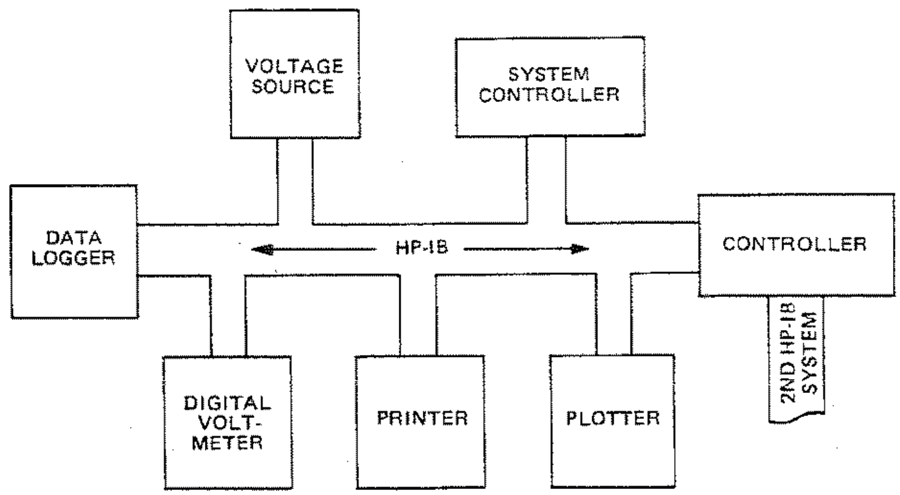
1. **Talkers** — Devices which send information on the bus when they have been addressed.
2. **Listeners** — Devices which receive information sent on the bus when they have been addressed.

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.

A typical HP-IB system is shown below.



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 more than one device. These quantities of information can easily be thought of as "messages."

In turn, the messages can be classified into 12 types. The list below gives the 12 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.

2. **The Trigger Message.** This message causes the listening device(s) to perform a device-dependent action when addressed.

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 all devices currently addressed to listen to switch from local front-panel control to remote program control.

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 devic operator from manually inhibiting remote program control.

7. **The Clear Lockout/Local Message.** This message causes all devices on the bus to be removed from Local Lockout and revert to Local. 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 by sending the device's Status Byte Message 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.

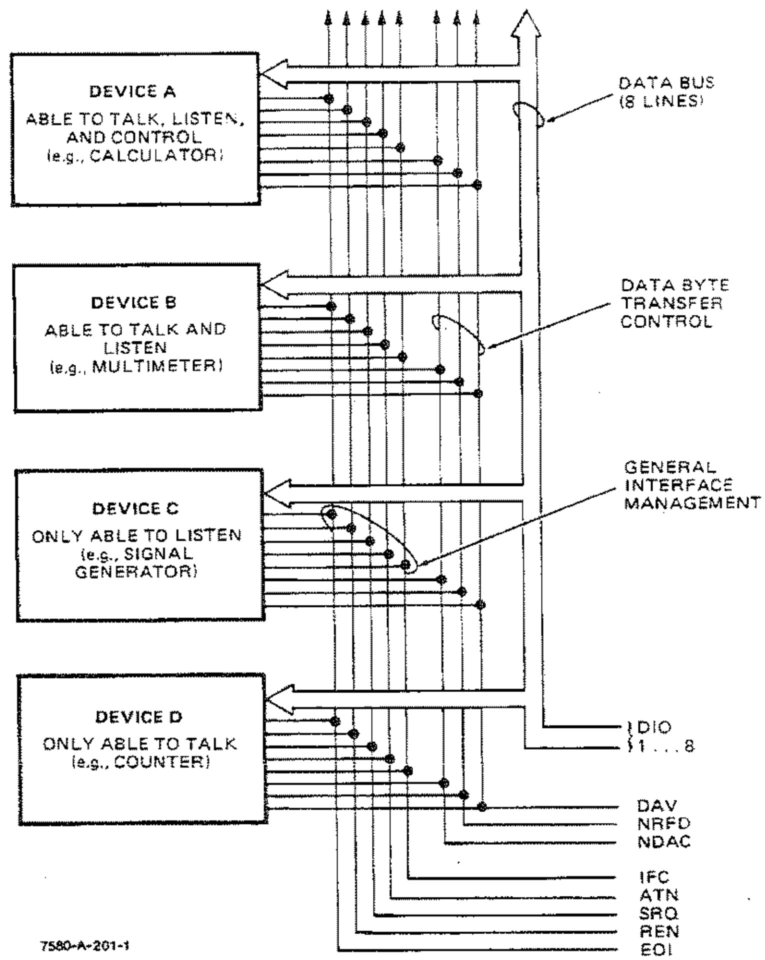
The HP Interface Bus

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 (DIO1 through DIO8) 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



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 is 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 8 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 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. Multi-line messages, which cannot exist concurrently with other multi-line messages, are sent over the eight data lines and the three handshake lines. Uni-line 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 multi-line messages.
- Universal commands cause every instrument equipped to do so to perform a specific interface operation. They include multi-line messages and three uni-line 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 multi-line commands. An instrument responds to an addressed command, however, only after an address has already told it to be talker or listener.
- Secondary commands are multi-line 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 7-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 ANSI/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. 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
Instruction Summary

HP-RL and HP-GL Syntax

This appendix lists the formal syntax for each instruction in alphabetical order of the instruction's two-letter mnemonic. The first list gives the HP-RL recorder instructions and the second list gives the HP-GL plotter instructions.

Each instruction is listed with its purpose, syntax, parameter or response type, range, and default condition. Refer to the indicated pages for details. The semicolon is included as the terminator for all instructions except the HP-GL label instruction, LB. The ASCII carriage return (CR) and line feed (LF) characters are sent at the end of all output responses.

HP-RL Syntax Summary

DG
The Draw Grid Instruction
Page 3-10

DG grid type ;
or
DG ;

Purpose: Draws a user-defined or a fixed centimetre grid. Omitting parameter defaults to user-defined grid. Grid is drawn according to conditions specified by GL instruction.

Parameter	Format	Range	Default
grid type	integer	0 = user-defined grid 1 = centimetre grid	0

DO
The Set Data Output Instruction
Page 4-2

DO channel(s) , # of samples , format , trigger status ;
or
DO ;

Purpose: Defines how data samples are to be output when data acquisition mode is subsequently initiated by the QD (direct) or QI (buffered) instructions. Omitting parameters defaults to continuous sampling of all channels in ASCII format with no trigger status.

Parameter	Format	Range	Default
channel(s)	integer	0= channels 1, 2, and 3 1= channel 1 2= channel 2 3= channel 3	0
# of samples	integer	0 to 32 767 (0= continuous sampling)	0
format	integer	0= ASCII 1= binary	0
trigger status	integer	0= no trigger status 1= send trigger status	0

GL
The Set Grid Divisions Instruction
Page 3-11

GL X-axis divisions , Y-axis divisions (, quadrants) ;
or
GL ;

Purpose: Specifies the number of divisions in the user-defined grid and the quadrants where it is to be drawn by the DG instruction. Omitting parameters sets default conditions.

Parameter	Format	Range	Default	
			A/A4	B/A3
X-axis divisions	integer	1 to 100	25	38
Y-axis divisions	integer	1 to 100	18	25
quadrants	integer	1 to 15	1	1

IR FS_1 (, FS_2 (, FS_3)) ;
or
IR ;

Purpose: Sets range of each input channel. Omitting parameters defaults all channels to 10.0 volts.

Parameter	Format	Range	Default
FS_1, FS_2, FS_3	real	0.0050 V to 100.0 V	10.0 V

IR
The Set Full Scale Range Instruction
Page 3-5

IT year, month, day, hours, minutes;

Purpose: Sets the time and date of the internal real-time clock. Seconds are set to zero when IT is executed.

Parameter	Format	Range	Default
year	integer	0 to 99	none
month	integer or character	1 to 12 or JAN to DEC	
day	integer	1 to 31	
hours	integer	0 to 23	
minutes	integer	0 to 59	

IT
The Set Real-Time Clock Instruction
Page 3-18

IZ Z_X, Z_Y (, FS_X, FS_Y) ;
or
IZ ;

Purpose: Sets location of zero (Z) and full scale (FS) points. Omitting parameters sets points to default locations.

Parameter	Format	Range	Default
X,Y coordinates	integer	−32 768 to 32 767	dependent on current paper size

IZ
The Set Zero and Full Scale Instruction
Page 3-3

MS
The Measurement Start Instruction
Page 4-1

MS measurement ;
or
MS ;

Purpose: Initiates either direct or buffered recording measurements. An MS with no parameter initiates direct recording.

Parameter	Format	Range	Default
measurement	integer	0 = direct recording 1 = buffered recording	0

MT
The Measurement Terminate Instruction
Page 4-2

MT ;

Purpose: Terminates either direct and buffered recording measurements or the data acquisition mode.

Parameter: None

PL
The Plot Buffer Instruction
Page 4-2

PL ;

Purpose: Plots buffer contents for each selected channel in order, using the pen with the same number as the channel(s).

Parameter: None

QA
The Query Trigger Level and Width Instruction
Page 3-15

QA ;

Purpose: Outputs the currently requested trigger level and width voltages and whether or not the values are valid.

Response: level , width , validity CRLF

QB
The Query Total Time Instruction
Page 3-10

QB ;

Purpose: Outputs the currently specified duration of a recording.

Response: time , units CRLF

where: time = 0.030 to 100 (seconds) units = 0 (seconds)
1.00 to 100 (minutes 1 (minutes)
1.00 to 24.0 (hours) 2 (hours)

QC
The Query Channel Status Instruction
Page 12-5

QC ;

Purpose: Outputs the decimal value (0 to 255) of the channel status byte in ASCII.

Response: channel status CRLF

QD ;

QD
The Query Direct A/D Data Sample Instruction
Page 4-5

Purpose: Initiates streaming of data samples from the A/D converters to a computer (data acquisition mode).

Response: Outputs number of samples specified by DO instruction at the rate specified by the SD instruction.

QG ;

QG
The Query Post- and Pre-Trigger Time Instruction
Page 3-17

Purpose: Outputs the currently requested post- and pre-trigger time and whether or not the time value is valid.

Response: time , units , validity CRLF

where: time = ±0.00 to 100 (seconds) units = 0 (seconds)
±1.00 to 100 (minutes) 1 (minutes)
±1.00 to 24.0 (hours) 2 (hours)

QI ;

QI
The Query Buffered A/D Data Sample Instruction
Page 4-6

Purpose: Initiates the transfer of the channel buffer contents to a computer.

Response: Outputs number of samples specified by DO instruction (SD instruction is ignored).

QL ;

QL
The Query Grid Divisions Instruction
Page 3-13

Purpose: Outputs the currently specified number of divisions in the user-defined grid and the quadrants where it is to be drawn.

Response: X-axis divisions , Y-axis divisions , quadrants CRLF

QM ;

QM
The Query Recording Mode Instruction
Page 3-3

Purpose: Outputs the current definitions of the independent (X) and dependent (Y) axes of the recording area.

Response: mode CRLF

QR ;

QR
The Query Full Scale Range Instruction
Page 3-6

Purpose: Outputs the current range settings for all three channels.

Response: range 1 , range 2 , range 3 CRLF

QS
The Query Recorder Status Instruction
Page 12-3

QS ;

Purpose: Output the decimal value (0–255) of the recorder status byte in ASCII.

Response: recorder status CRLF

QT
The Query Trigger Mode Instruction
Page 3-14

QT ;

Purpose: Outputs the currently selected trigger mode.

Response: mode , option CRLF

where: mode = 0 (manual)
1 (external)
2 (internal)

option = 0 (above level)
1 (below level)
2 (inside window)
3 (outside window)

QU
The Query Trigger Time Instruction
Page 3-16

QU ;

Purpose: Outputs the time the last trigger occurred.

Response: hours , minutes , seconds CRLF

where: hours = 0 to 23
minutes = 0 to 59
seconds = 0 to 59

QV
The Query DC Offset Instruction
Page 3-8

QV ;

Purpose: Outputs the currently requested dc offset for each channel and whether or not the values are valid.

Response: offset 1 , offset 2 , offset 3 , validity CRLF

QW
The Query Real-Time Clock Instruction
Page 3-18

QW ;

Purpose: Outputs the current time and date of the real-time clock.

Response: year , month , day , hours , minutes , seconds CRLF

where: year = 00 to 99
month = 01 to 12
day = 01 to 31
hours = 00 to 23
minutes = 00 to 59
seconds = 00 to 59

QZ ;

QZ
The Query Zero and Full Scale Instruction
Page 3-5

Purpose: Outputs the current X- and Y-coordinates of the zero and full scale points.

Response: Z_X, Z_Y, FS_X, FS_Y CRLF

RE mode ;
or
RE ;

RE
The Set Recording Mode Instruction
Page 3-2

Purpose: Defines the independent (X) and dependent (Y) axes of the recording area in terms of the input channels and/or time. Omitting parameter defaults to channel 1 versus time.

Parameter	Format	Range	Default
mode	integer	3 through 15 except 6, 10, and 14	9

RL state ;
or
RL ;

RL
The Remote/Local Instruction
Page 11-2

Purpose: Disables or enables the front-panel controls. An RL instruction with no parameter enables front-panel controls.

Parameter	Format	Range	Default
state	integer	0 = local state 1 = remote state	0

SD time ;
or
SD ;

SD
The Set Sample Delay Instruction
Page 4-4

Purpose: Sets the time interval between samples, or sample sets, for subsequent data transfers initiated by the QD instruction. Omitting parameter defaults time interval to 0.1 second.

Parameter	Format	Range	Default
time	real	0.002 to 1000 seconds	0.1 second

SV
The Set DC Offset Instruction
Page 3-7

SV offset 1 (, offset 2 (, offset 3)) ;
or
SV ;

Purpose: Sets zero voltage position of each channel's input signal. Omitting parameters sets the offset of all three channels to 0.0 volts.

Parameter	Format	Range	Default
offset 1, 2, 3	real	±100 volts	0.0 volts

TA
The Set Trigger Level and Width Instruction
Page 3-15

TA level (, width) ;
or
TA ;

Purpose: Specifies the trigger level for above and below level triggering and the trigger level and width for inside and outside window triggering. Omitting parameters defaults both trigger level and width to 0.0 volts.

Parameter	Format	Range	Default
level	real	±200 volts	0.0 volts
width	real	0 to 200 volts	0.0 volts

TB
The Set Total Time Instruction
Page 3-8

TB time (, units) ;
or
TB ;

Purpose: Specifies the duration of a recording. Omitting parameters defaults duration to 1.0 second.

Parameter	Format	Range	Default
time	real	seconds: 0.030 to 100 minutes: 1.00 to 100 hours: 1.00 to 24.0	1
units	integer	0 = seconds 1 = minutes 2 = hours	0

TD The Label Time and Date Instruction

Page 3-19

TD label type;
or
TD ;

Purpose: Provides the means to label the setup conditions for a recording, the current date and time, and the time that the trigger occurred. Omitting parameter specifies printing the time and date label.

Parameter	Format	Range	Default
label type	integer	0 = time and date 1 = time 2 = date 3 = trigger time 4 = set up conditions	0

TG The Set Post- and Pre-Trigger Time Instruction

Page 3-16

TG time (, units);
or
TG ;

Purpose: Specifies either a post-trigger time delay or the amount of total time that is used to record pre-trigger data. Omitting parameters defaults both post- and pre-trigger time to zero seconds.

Parameter	Format	Range	Default
time	real	±0.00 to 100 (seconds) ±1.00 to 100 (minutes) ±1.00 to 24.0 (hours)	0
units	integer	0 = seconds 1 = minutes 2 = hours	0

TM The Set Trigger Mode Instruction

Page 3-14

TM mode (, option);
or
TM ;

Purpose: Selects the triggering mode used to start a recording. Omitting parameters defaults to manual trigger mode.

Parameter	Format	Range	Default
mode	integer	0 = manual 1 = external 2 = internal	0
options	integer	0 = above level 1 = below level 2 = inside window 3 = outside window	0

XS
The Set Plotter or Recorder Status Instruction
Page 12-2

XS status type;
or
XS ;

Purpose: Selects either the plotter status byte (OS) or the recorder status byte (QS) for use as the HP-IB status byte. Omitting parameter defaults to plotter status byte.

Parameter	Format	Range	Default
status type	integer	0 = plotter status byte 1 = recorder status byte	0

HP-GL Syntax Summary

CA
The Designate Alternate Character Set Instruction
Page 8-7

CA set number;
or
CA ;

Purpose: Designates the alternate character set. Omitting parameter defaults to set 0.

Parameter	Format	Range	Default
set number	integer	0 to 4	0

CP
The Character Plot Instruction
Page 8-19

CP spaces, lines;
or
CP ;

Purpose: Move the pen the number of spaces and lines specified. Omitting parameters causes a carriage return and line feed operation.

Parameter	Format	Range	Default
spaces	real	−128.0000 to 127.9999	carriage return
lines	real	−128.0000 to 127.9999	point

CS set number;
or
CS ;

Purpose: Designates the standard character set. Omitting parameter defaults to set 0.

CS
The Designate Standard Character Set Instruction
Page 8-7

Parameter	Format	Range	Default
set number	integer	0 to 4	0

DF ;

Purpose: Sets certain plotter functions to a predefined state.

DF
The Default Instruction
Page 2-16

DI run, rise;
or
DI ;

Purpose: Sets the direction of labels. Omitting parameters causes horizontal labels and is the same as DI 1, 0. At least one parameter must be effectively nonzero; i.e., |≥ 0.0004|.

DI
The Absolute Direction Instruction
Page 8-9

Parameter	Format	Range	Default
run	real	−128.0000 to 127.9999	1
rise	real	−128.0000 to 127.9999	0

DR run, rise;
or
DR ;

Purpose: Sets the direction of labels. Omitting parameters causes horizontal labels and is the same as DR 1, 0. Run is % of ($P2_X - P1_X$), rise is % of ($P2_Y - P1_Y$).

DR
The Relative Direction Instruction
Page 8-11

Parameter	Format	Range	Default
run	real	−128.0000 to 127.9999	1
rise	real	−128.0000 to 127.9999	0

DT
The Define Terminator Instruction
Page 8-3

DT t;

Purpose: Defines the character which terminates label mode.

Parameter	Format	Range	Default
t	character	ASCII characters with decimal values 1 through 127	ETX (decimal value 3)

IM
The Input Mask Instruction
Page 12-7

IM E-mask (, S-mask (, P-mask)) ;
or
IM ;

Purpose: Controls the conditions under which HP-RL and HP-GL error status is reported, the conditions that can cause an HP-IB service request message, and the conditions that can cause a positive response to an HP-IB parallel poll. Omitting parameters defaults mask values to 223 , 0 , 0.

Parameter	Format	Range	Default
E-mask	integer	0 to 255	223
S-mask	integer	0 to 255	0
P-mask	integer	0 to 255	0

IN
The Initialize Instruction
Page 2-17

IN ;

Purpose: Sets plotter functions the same as DF and sets certain recorder functions to predefined states. In addition, the pen is raised, any error is cleared, the plotter status byte is the HP-IB status byte and bit position 3 is set true, channel buffers are flushed, rotation is set to 0 degrees, and all limits and scaling points are set to the default coordinate values for the currently selected paper size.

IP
The Input P1 and P2 Instruction
Page 6-2

IP $P1_X, P1_Y$ (, $P2_X, P2_Y$);

Purpose: Sets P1/P2 scaling points. Omitting parameters sets P1 and P2 to the default coordinates for the currently selected paper size.

Parameter	Format	Range	Default
P1 , P2	integer	−32 768 to 32 767	dependent on current paper size

IW The Input Window Instruction
Page 6-13

IW $X_{lower\ left}$, $Y_{lower\ left}$, $X_{upper\ right}$, $Y_{upper\ right}$;
or
IW ;

Purpose: Sets window (soft-clip limits) inside which plotting can occur. Omitting parameters sets window area to default coordinate values for the currently selected paper size.

Parameter	Format	Range	Default
X,Y	integer	−32 768 to 32 767	dependent on current paper size

LB The Label Instruction
Page 8-2

LB c...c t

where t is the label terminator defined by DT. Default is ETX, decimal equivalent 3.

Purpose: Draws the character string using the currently selected character set.

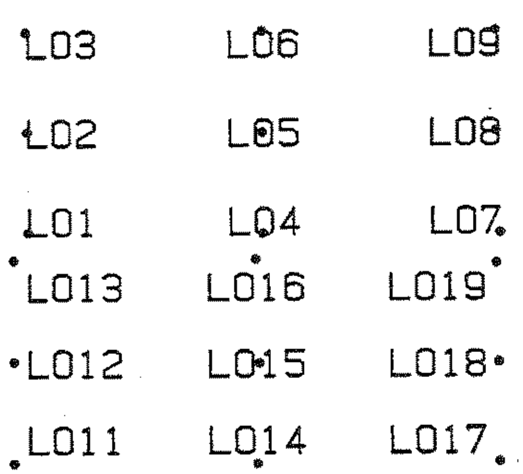
Parameter	Format	Range	Default
c...c	character	all ASCII characters	off

LO The Label Origin Instruction
Page 8-17

LO position number ;
or
LO ;

Purpose: Positions labels relative to current pen position. Omitting parameter defaults to LO 1.

Parameter	Format	Range	Default
position number	integer	1–9 and 11–19	1

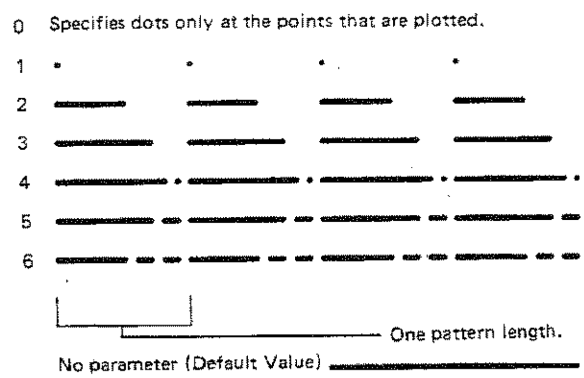


LT
The Line Type Instruction
Page 7-5

LT pattern number (, pattern length) ;
or
LT ;

Purpose: Sets the line type used for drawing lines. Omitting parameters defaults to a solid line.

Parameter	Format	Range	Default
pattern number	real	−128.0000 to 127.9999	solid line
pattern length	real	−128.0000 to 127.9999	4% of diagonal (P2 − P1)



OA
The Output Actual Position and Pen Status Instruction
Page 10-1

OA ;

Purpose: Outputs the pen's physical position.

Response: X,Y,P CRLF

where: X and Y are in plotter units; P is pen status (0 = up, 1 = down)

OC
The Output Commanded Position and Pen Status Instruction
Page 10-2

OC ;

Purpose: Outputs the pen's logical position and status.

Response: X,Y,P CRLF

where: X and Y are in plotter units if scaling is off or user units if scaling is on; P is pen status (0 = up, 1 = down)

OE
The Output Error Instruction
Page 12-5

OE ;

Purpose: Outputs the number of the first HP-RL or HP-GL error.

Response: error number CRLF

OF ;

OF
The Output Factors Instruction
Page 10-2

Purpose: Outputs the number of plotter units per millimetre in each axis.

Response: 40 , 40   CRLF

OH ;

OH
The Output Hard-Clip Limits Instruction
Page 2-4

Purpose: Outputs the lower-left and upper-right coordinates of the hard-clip limits in plotter units.

Response: $X_{lower\ left}$, $Y_{lower\ left}$, $X_{upper\ right}$, $Y_{upper\ right}$ CRLF

OI ;

OI
The Output Identification Instruction
Page 10-2

Purpose: Outputs the instrument's model number in ASCII.

Response: 7090A   CRLF

OO ;

OO
The Output Options Instruction
Page 10-3

Purpose: Outputs the instrument's capabilities.

Response: 0,1,0,0,0,0,0,0   CRLF

(1 —— indicates pen select capability is included)

OP ;

OP
The Output P1 and P2 Instruction
Page 6-4

Purpose: Outputs the plotter-unit coordinates of the scaling points P1 and P2.

Response: $P1_X$, $P1_Y$, $P2_X$, $P2_Y$ CRLF

OS ;

OS
The Output Plotter Status Instruction
Page 12-2

Purpose: Outputs the decimal value (0–255) of the plotter status byte in ASCII.

Response: plotter status   CRLF

OW ;

OW
The Output Window Instruction
Page 6-15

Purpose: Outputs the plotter unit coordinates of the lower-left and upper-right corners of the current window.

Response: $X_{lower\ left}$, $Y_{lower\ left}$, $X_{upper\ right}$, $Y_{upper\ right}$ CRLF

OY
The Locate Syntax Error Instruction
Page 12-6

OY ;

Purpose: Outputs the sixteen characters received immediately prior t an HP-RL or HP-GL error.

Response: ASCII character string

OZ
The Output Command String Instruction
Page 10-3

OZ ;

Purpose: Outputs OZ and the fourteen characters received immediately before OZ. Carriage return and line feed characters are not included.

Response: ASCII character string

PA
The Plot Absolute Instruction
Page 5-3

PA X_1 coordinate, Y_1 coordinate (, X_2 coordinate, Y_2 coordinate, . . . , X_n coordinate, Y_n coordinate);
or
PA ;

Purpose: Plots to the X,Y coordinates in the order listed using the current pen up/down state. PA; sets absolute plotting. Coordinate pairs represent plotter units if scaling is off and user units if scaling is on.

Parameter	Format	Range	Default
X,Y coordinates	integer	−32 768 to 32 767	none

PD
The Pen Down Instruction
Page 5-2

PD ;
or
PD X,Y(,. . .);

Purpose: Programmatically lowers the pen. Parameters may be included as in PA or PR.

Parameter	Format	Range	Default
X,Y	integer	−32 768 to 32 767	none

PR X_1 increment, Y_1 increment (, X_2 increment, Y_2 increment, . . . , X_n increment, Y_n increment);
or
PR ;

Purpose: Plots, in order, to the points indicated by the X,Y increments, relative to the previous position. PR; sets relative plotting for PU or PD with parameters.

PR
The Plot Relative Instruction
Page 5-6

Parameter	Format	Range	Default
X,Y increments	integer	−32 768 to 32 767	none

PS paper size;

Purpose: Used to toggle between A and B, or A3 and A4 paper sizes.

PS
The Paper Size Instruction
Page 2-4

Parameter	Format	Range	Default
paper size	integer	0 to 3 = B or A3 size 4 to 127 = A or A4 size	none

PU ;
or
PU X,Y (, . . .);

Purpose: Programmatically raises the pen. Parameters may be included as in PA or PR.

PU
The Pen Up Instruction
Page 5-2

Parameter	Format	Range	Default
X,Y	integer	−32 768 to 32 767	none

RO angle;
or
RO ;

Purpose: Rotates the coordinate systems 90 degrees. Omitting parameter defaults to 0 degrees (no rotation).

RO
The Rotate Coordinate System Instruction
Page 13-3

Parameter	Format	Range	Default
angle	integer	0 = normal 90 = rotated	0

SA The Select Alternate Set Instruction Page 8-8

SA ;

Purpose: Selects the alternate character set designated by the CA instruction as the character set to be used for subsequent labeling.

SC The Scale Instruction Page 6-4

SC X_{min}, X_{max}, Y_{min}, Y_{max};
or
SC ;

Purpose: Scales the plotting area into user units. Omitting parameters cancels scaling, and parameters of plot instructions are subsequently interpreted as plotter units.

Parameter	Format	Range	Default
X,Y	integer	−32 768 to 32 767	plotter units

SI The Absolute Character Size Instruction Page 8-13

SI width, height;
or
SI ;

Purpose: Sets character width and height for labels, in centimetres. Omitting parameters defaults character size based on the paper size.

Parameter	Format	Range	Default	
			A/A4	B/A3
width	real	−128.0000 to 127.9999	0.187	0.285
height	real	−128.0000 to 127.9999	0.269	0.375

SL The Character Slant Instruction Page 8-16

SL tan θ;
or
SL ;

Parameter	Format	Range	Default
tan θ	real	−128.0000 to 127.9999	0

SM
The Symbol Mode Instruction
Page 7-4

SM character;
or
SM ;

Purpose: Draws a single character which is centered at the end of each plotted vector. Omitting parameter cancels symbol mode.

Parameter	Format	Range	Default
character	character	all printing characters except semicolon	off

SP
The Select Pen Instruction
Page 5-2

SP pen number;
or
SP ;

Purpose: Selects or stores a pen. Omitting the parameter or a parameter of 0 stores the pen.

Parameter	Format	Range	Default
pen number	integer	0 to 6	0

SR
The Relative Character Size Instruction
Page 8-15

SR width, height;
or
SR ;

Purpose: Specifies the size of characters and symbols as a percentage of the distance between P1 and P2. Omitting parameters defaults to 0.75 for width and 1.5 for height.

Parameter	Format	Range	Default
width	real	−128.0000 to 127.9999	0.75
height	real	−128.0000 to 127.9999	1.5

SS
The Select Standard Set Instruction
Page 8-8

SS ;

Purpose: Selects the standard character set designated by the CS instruction as the character set used for subsequent labeling.

TL
The Tick Length Instruction
Page 7-2

TL tp (, tn) ;
or
TL ;

Purpose: Establishes the length of ticks drawn with the instructions XT and YT. Omitting parameters defaults tp and tn to 0.5% of ($P2_Y - P1_Y$) for XT and 0.5% of ($P2_X - P1_X$) for YT.

Parameter	Format	Range	Default
tp, tn	real	−128.0000 to 127.9999	tp and tn = 0.5% of ($P2_Y - P1_Y$) for XT and 0.5% of ($P2_X - P1_X$) for YT

VS
The Velocity Select Instruction
Page 5-8

VS pen velocity ;
or
VS ;

Purpose: Specifies the pen-up and pen-down speed for plotting and labeling operations. Omitting parameter defaults pen speed to 75 cm/s (30 in./s).

Parameter	Format	Range	Default
pen velocity	real	0 to 127.9999 cm/s	75 cm/s

XT
The X-Tick Instruction
Page 7-1

XT ;

Purpose: Draws a vertical X-tick at the current pen location.

YT
The Y-Tick Instruction
Page 7-1

YT ;

Purpose: Draws a horizontal Y-tick at the current pen location.

Appendix C
Reference Material

Specifications*

Inputs	
Number of channels	3
Type of input	floating, guarded
Sensitivity	5 mV to 100 V full scale
Zero offset	±2 full scale or ±100 V maximum
Input impedance (NOMINAL)	1 MΩ, shunted by 45 pf.
Maximum input voltage	200 V, dc or peak
Maximum source resistance	10 kΩ
Common mode rejection ratio	140 dB dc; 100 dB ac @ 60 Hz with 1 kΩ unbalance in LOW terminal on most sensitive range @ 25°C, 50% RH
Timebase	
Range:	
Buffer mode	30 milliseconds to 24 hours
Direct record mode	1 second to 24 hours
Accuracy	±0.1%
Dynamic Performance	
Slewing speed (NOMINAL):	
Direct record mode	127 cm/s (50 in./s)
Plotting mode	75 cm/s (30 in./s)
Acceleration (NOMINAL)	2 g constant (1960 cm/s^2)
Bandwidth (−3 dB)	3 kHz for all full-scale ranges ⩾ 20 mV 2.6 kHz for all full-scale ranges < 20 mV
Peak capture	250 μs at fastest timebase range
Memory Per Channel	
Size	1000 words
Resolution	12 bits

*Specifications describe the instrument's warranted performance. Supplemental characteristics are intended to provide information useful in applying the instrument by giving **typical** or **nominal**, but not warranted, performance parameters.

Trigger Characteristics

Internal trigger:
- Inside or outside window
- Above or below level, selectable over the full-scale range in 1.0% increments (NOMINAL)
- Source, channel 1

External trigger:
- BNC connector, TTL level or contact closure to ground

Manual trigger:
- Available from front-panel controls or I/O

Display:
- Up to 100% pre-trigger; up to 24 hour post-trigger delay after trigger before measurement start

Accuracy**

Electrical Accuracy @ 25 Degrees C

Range	Constant Inaccuracy	Percent Reading Inaccuracy
5 mV	±0.013 mV	±0.055%
10 mV	±0.021 mV	
20 mV	±0.036 mV	
50 mV	±0.082 mV	
100 mV	±0.158 mV	
200 mV	±0.306 mV	
500 mV	±0.760 mV	
1 V	±0.0015 V	
2 V	±0.0030 V	
5 V	±0.0076 V	
10 V	±0.0152 V	
20 V	±0.0304 V	
50 V	±0.0760 V	
100 V	±0.1520 V	

Electrical Accuracy Temperature Coefficient from 25 Degrees C

Range	Constant Inaccuracy Per Degree C	Percent Reading Inaccuracy Per Degree C
5 mV	±0.0022 mV	±0.01%
10 mV	±0.0028 mV	
20 mV	±0.0040 mV	
50 mV	±0.0076 mV	
100 mV	±0.0136 mV	
200 mV	±0.0256 mV	
500 mV	±0.0616 mV	
1 V	±0.0001 V	
2 V	±0.0002 V	
5 V	±0.0006 V	
10 V	±0.0012 V	
20 V	±0.0024 V	
50 V	±0.0060 V	
100 V	±0.0120 V	

**Accuracy specifications assume |offset| < full-scale range and a signal source resistance ⩽ 10 kΩ.

Range	Constant Inaccuracy	Percent Reading Inaccuracy
5 mV	±6	±0.055%
10 mV	±5	
20 mV	±4	
50 mV and up	±3	

Electrical Accuracy @ 25 Degrees C (A/D Converter Counts)

Range	Constant Inaccuracy	Percent Reading Inaccuracy
5 mV	±0.88	±0.01%
10 mV	±0.56	
20 mV	±0.40	
50 mV and up	±0.30	

Electrical Accuracy Temperature Coefficient from 25 Degrees C (A/D Converter Counts)

Scaling Without Using the SC Instruction

The plotter movements are in terms of plotter units where a plotter unit = 0.025 mm. While the plotter can be scaled into user units using the SC instruction, it may be convenient for you to write programs where numbers to be plotted are in some units other than plotter units. These "user units" can be converted into plotter units by the computer using the following equations:

$$X_{scaled} = \left[\frac{P2_X - P1_X}{U2_X - U1_X}\right] A_X + P1_X - U1_X \left[\frac{P2_X - P1_X}{U2_X - U1_X}\right]$$

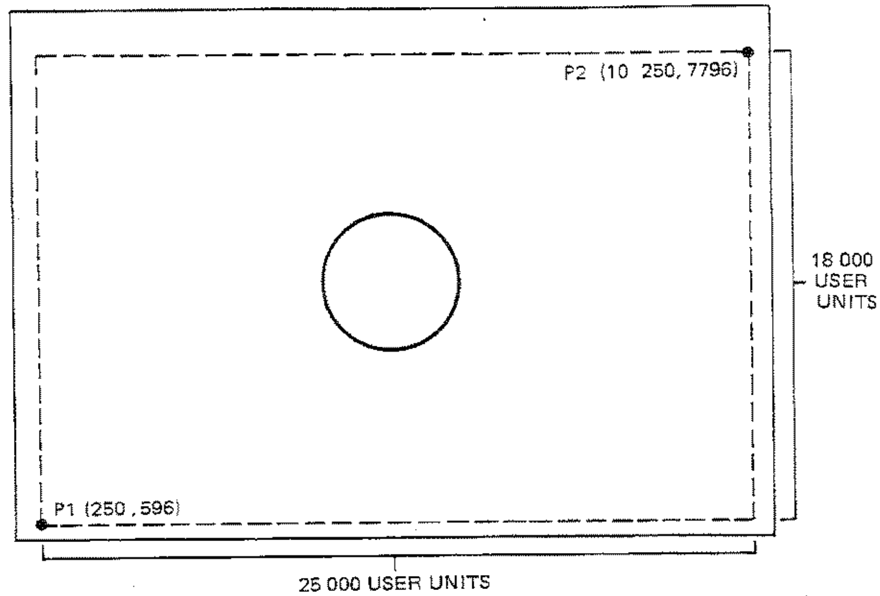
$$Y_{scaled} = \left[\frac{P2_Y - P1_Y}{U2_Y - U1_Y}\right] A_Y + P1_Y - U1_Y \left[\frac{P2_Y - P1_Y}{U2_Y - U1_Y}\right]$$

where: A_X is the X-coordinate of the desired point in user units,
A_Y is the Y-coordinate of the desired point in user units,
$P1_X$ is the X-coordinate of P1 in plotter units,
$P1_Y$ is the Y-coordinate of P1 in plotter units,
$P2_X$ is the X-coordinate of P2 in plotter units,
$P2_Y$ is the Y-coordinate of P2 in plotter units,
$U1_X$ is the X-coordinate of P1 in user units,
$U1_Y$ is the Y-coordinate of P1 in user units,
$U2_X$ is the X-coordinate of P2 in user units, and
$U2_Y$ is the Y-coordinate of P2 in user units.

The following example is included to demonstrate the use of scaling equations.

PROBLEM

Scale the platen area (P1 = 250 , 596 and P2 = 10 250 , 7796) into user units where P1 = 0 , 0 and P2 = 25 000 , 18 000. At the center point (X = 12 500, Y = 9000), draw a circle with radius 2500 as shown on the following page.



SOLUTION

1. Recall that the equations of a circle are:

   $X = R\cos T$
   $Y = R\sin T$
   where $0 \leqslant T \leqslant 2\pi$

2. Since we are to plot relative to a point that is not at the origin, an offset X_O, Y_O must be added to the circle equations. The offset in user units is:

   $X_O = 12\,500$
   $Y_O = 9000$

3. The desired circle equations are then:

   $A_X = 2500\cos T + 12\,500$
   $A_Y = 2500\sin T + 9000$

4. Determine the user scale:

   $X = 0$ to $25\,000$
   $Y = 0$ to $18\,000$

   therefore

   $U1_X = 0$
   $U1_Y = 0$
   $U2_X = 25\,000$
   $U2_Y = 18\,000$

5. Determine the values for P1 and P2 which were set using the IP instruction:

   $P1 = 250, 596$
   $P2 = 10\,250, 7796$

   therefore

   $P1_X = 250$
   $P1_Y = 596$
   $P2_X = 10\,250$
   $P2_Y = 7796$

6. Solving for X and Y:

$$X = \left[\frac{P2_X - P1_X}{U2_X - U1_X}\right] A_X + P1_X - U1_X \left[\frac{P2_X - P1_X}{U2_X - U1_X}\right]$$

$$= \left[\frac{10\,250 - 250}{25\,000 - 0}\right] (2500\cos T + 12\,500) + 250 - 0 \left[\frac{10\,250 - 250}{25\,000 - 0}\right]$$

$$= 0.4\,(2500\cos T + 12\,500) + 250 - 0$$

$$= 1000\cos T + 5250$$

$$Y = \left[\frac{P2_Y - P1_Y}{U2_Y - U1_Y}\right] A_Y + P1_Y - U1_Y \left[\frac{P2_Y - P1_Y}{U2_Y - U1_Y}\right]$$

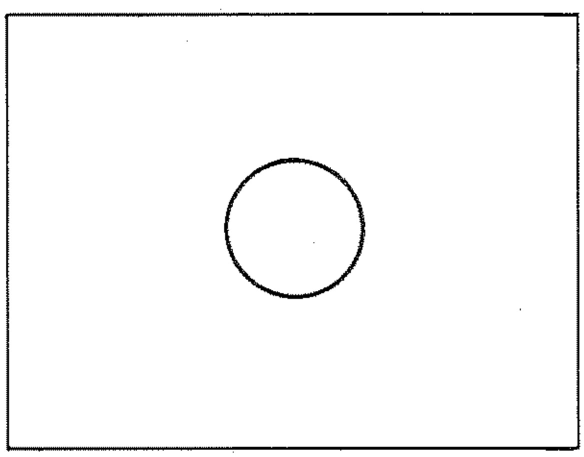
$$= \left[\frac{7796 - 596}{18\,000 - 0}\right] (2500\sin T + 9000) + 596 - 0 \left[\frac{7796 - 596}{18\,000 - 0}\right]$$

$$= 0.4\,(2500\sin T + 9000) + 596 - 0$$

$$= 1000\sin T + 4196$$

7. Sending the following program will plot the required circle.

```
10     OUTPUT 705;"IP250,596,10250,7796;SP1;"
20     FOR T=0 TO 2*PI STEP PI/20
30     X=1000*COS(T)+5250
40     Y=1000*SIN(T)+4196
50     OUTPUT 705;"PA";X;Y;"PD"
60     NEXT T
70     OUTPUT 705;"PU;SP0;"
80     END
```



HP-RL and HP-GL Error Messages

error 0	No error.
error 1	Instruction not recognized. The plotter has received an illegal character sequence.
error 2	Wrong number of parameters. Too many or too few parameters have been sent with an instruction.
error 3	Out-of-range parameters.
error 4	Not used.
error 5	Unknown character set. A character set out of the range 0–4 has been designated as either the standard or alternate character set.
error 6	Position overflow. An attempt to draw a character (LB) or perform a CP that is located outside the plotter's numeric limit of −32 768 to +32 767.
error 7	Not used.
error 8	Vector received while pinch wheels raised or servo inoperative.

The No-Operation Instructions

In order to maintain software compatibility with the HP 9872 plotter, the HP 7090 recognizes seven HP 9872-related instructions as no-operation (NOP) instructions. These seven NOP instructions are:

Automatic Pen Pickup AP	Advance Full Page AF
Adaptive Velocity VA	Advance Half Page AH
Normal Velocity VN	Enable Cutter EC
Page PG	

If these instructions are included in a program, they are recognized by the HP 7090 and implemented as a NOP (i.e., they are ignored).

ASCII Character Codes

Binary is often used as a code to represent not only numbers, but also alphanumeric characters such as "A" or "," or "x" or "2." One of the most common binary codes used is ASCII[1]. ASCII is an eight-bit code, containing seven data bits and one parity bit. The HP 7090 uses ASCII for most I/O operations. No parity bit is used. For example:

Character	ASCII Binary Code	ASCII Decimal Code
A	01000001	65
B	01000010	66
?	00111111	63

[1]American Standard Code for Information Interchange.

A complete list of ASCII characters and their decimal representation and the characters drawn by the plotter in each of the five character sets are shown on the following pages. The five character sets are:

Set No.	Description
Set 0	ANSI ASCII
Set 1	HP 9825 Character Set
Set 2	French/German
Set 3	Scandinavian
Set 4	Spanish/Latin American

HP 7090 ASCII Code Definitions

Decimal Value	ASCII Character	All Sets
0	NULL	No Operation (NOP)
1	SOH	NOP
2	STX	NOP
3	ETX	End Label Instruction
4	ETO	NOP
5	ENQ	NOP
6	ACK	NOP
7	BEL	NOP
8	BS	Backspace
9	HT	NOP
10	LF	Line Feed
11	VT	Inverse Line Feed
12	FF	NOP
13	CR	Carriage Return
14	SO	Select Alternate Character Set
15	SI	Select Standard Character Set
16	DLE	NOP
17	DC1	NOP
18	DC2	NOP
19	DC3	NOP
20	DC4	NOP
21	NAK	NOP
22	SYN	NOP
23	ETB	NOP
24	CAN	NOP
25	EM	NOP
26	SUB	NOP
27	ESC	NOP
28	FS	NOP
29	GS	NOP
30	RS	NOP
31	US	NOP
32	SP	Space

NOTE: Shaded characters have the automatic backspace feature. ■

Decimal Value	Set 0	Set 1	Set 2	Set 3	Set 4
33	!	!	!	!	!
34	"	"	"	"	"
35	#	#	£	£	¿
36	$	$	$	$	$
37	%	%	%	%	%
38	&	&	&	&	&
39	'	'	'	'	'
40	(	(	(	(	(
41	)	)	)	)	)
42	*	*	*	*	*
43	+	+	+	+	+
44	,	,	,	,	,
45	–	–	–	–	–
46	.	.	.	.	.
47	/	/	/	/	/
48	0	0	0	0	0
49	1	1	1	1	1
50	2	2	2	2	2
51	3	3	3	3	3
52	4	4	4	4	4
53	5	5	5	5	5
54	6	6	6	6	6
55	7	7	7	7	7
56	8	8	8	8	8
57	9	9	9	9	9
58	:	:	:	:	:
59	;	;	;	;	;
60	<	<	<	<	<
61	=	=	=	=	=
62	>	>	>	>	>
63	?	?	?	?	?
64	@	@	@	@	@
65	A	A	A	A	A
66	B	B	B	B	B
67	C	C	C	C	C
68	D	D	D	D	D
69	E	E	E	E	E
70	F	F	F	F	F
71	G	G	G	G	G
72	H	H	H	H	H
73	I	I	I	I	I
74	J	J	J	J	J
75	K	K	K	K	K
76	L	L	L	L	L
77	M	M	M	M	M
78	N	N	N	N	N
79	O	O	O	O	O
80	P	P	P	P	P

Decimal Value	Set 0	Set 1	Set 2	Set 3	Set 4
81	Q	Q	Q	Q	Q
82	R	R	R	R	R
83	S	S	S	S	S
84	T	T	T	T	T
85	U	U	U	U	U
86	V	V	V	V	V
87	W	W	W	W	W
88	X	X	X	X	X
89	Y	Y	Y	Y	Y
90	Z	Z	Z	Z	Z
91	[	[	[	Ø	[
92	\	ƒ	ç	Æ	¡
93	]	]	]	ø	]
94	^	↑	^	æ	^
95	_	_	_	_	_
96	`	`	`	`	`
97	a	a	a	a	a
98	b	b	b	b	b
99	c	c	c	c	c
100	d	d	d	d	d
101	e	e	e	e	e
102	f	f	f	f	f
103	g	g	g	g	g
104	h	h	h	h	h
105	i	i	i	i	i
106	j	j	j	j	j
107	k	k	k	k	k
108	l	l	l	l	l
109	m	m	m	m	m
110	n	n	n	n	n
111	o	o	o	o	o
112	p	p	p	p	p
113	q	q	q	q	q
114	r	r	r	r	r
115	s	s	s	s	s
116	t	t	t	t	t
117	u	u	u	u	u
118	v	v	v	v	v
119	w	w	w	w	w
120	x	x	x	x	x
121	y	y	y	y	y
122	z	z	z	z	z
123	{	π	¨	¨	~
124	¦		•	•	~
125	}	→	¨	¨	~
126	~	~	'	•	~
127					

Appendix D
Glossary of Terms

Absolute plotting	Plotting to a point whose location is specified relative to the reference coordinate system origin (0 , 0).
Aliasing	Reconstruction of a signal using less than 2 digitized samples from each cycle of the original signal.
Anisotropic scaling	Establishing user-units of unequal length in the X- and Y-axes.
Character space field	The space occupied by a single character, together with the space between it and the next character and the space above the character which separates it from the previous text line.
Clipping	Exclusion of data with coordinate values that exceed the currently defined recording or plotting area.
Hard-clip limits	The boundaries which are automatically established to prevent drawing off the paper.
HP-GL	Hewlett-Packard Graphics Language. The I/O instructions understood by the HP 7090 that control the plotter functions.
HP-IB	Hewlett-Packard Interface Bus. HP's implementation of the IEEE standard 488-1978 digital interface for programmable instrumentation.
HP-RL	Hewlett-Packard Recorder Language. The I/O instructions understood by the HP 7090 that control the recorder and data acquisition functions.
Isotropic scaling	Establishing user-units of equal length in the X- and Y-axes.
Latched bits	Bits which remain set to 1 until a specific action re-sets them to 0.
Plotting area	The area within the soft-clip limits, defined by the IW instruction, where labeling and plotter pen motion can occur.
Plotter unit	A plotter unit is 0.025 mm (0.000 98 in.) in length and is the smallest move that can be made.
Plotter unit equivalent	The plotter unit coordinates of a point that is specified in user units.
Polling	The process used by the computer to determine which device on the HP-IB has initiated a require service message.
Recording	The act of filling the buffers or drawing a real-time plot.
Recording area	The area where real-time and buffered data are plotted.

Reference coordinate system	The two-dimensional Cartesian coordinate system that is scaled into plotter units. The recording and plotting areas are mapped onto these coordinates via scaling points zero/full scale and P1/P2.
Relative plotting	Plotting to a point whose location is specified relative to the current pen position.
Scaling	Dividing the recording area into units of volts and/or time and the plotting area into user units.
Soft-clip limits	A rectangular window set by the IW instruction to restrict programmed pen motion. This window area has no effect on direct or buffered recording traces or front-panel initiated labels.
Trigger	The event which initiates data recording in the direct or buffered recording modes.
Windows	See soft-clip limits.

Subject Index

a

b

c

d

Subject Index (Continued)

d (Continued)

e

f

g

h

i

l

Subject Index (Continued)

m

n

o

p

p (Continued)

q

Subject Index (Continued)

q (Continued)

r

s

S (Continued)

t

t (Continued)

u

v

w

x

y

z