Agilent 4395A Network/Spectrum/Impedance Analyzer

Programming Manual

SERIAL NUMBERS

This manual applies directly to instruments with serial number prefix JP1KE and MY411. For additional important information about serial numbers, read in "Serial Number" in Chapter 12.

Agilent Technologies

Agilent Part No. 04395-90031
Printed in JAPAN December 2002

Fifth Edition

Notice

The information contained in this document is subject to change without notice.

Agilent Technologies Japan, Ltd.
Component Test PGU-Kobe
1-3-2, Murotani, Nishi-ku, Kobe-shi,
Hyogo, 651-2241 Japan

MS-DOS® is a U.S. registered trademark of Microsoft Corporation.

The customer shall have the personal, non-transferable rights to use, copy, or modify SAMPLE PROGRAMS in this manual for the Customer's internal operations. The customer shall use the SAMPLE PROGRAMS solely and exclusively for their own purpose and shall not license, lease, market, or distribute the SAMPLE PROGRAMS or modification of any part thereof.

Agilent Technologies shall not be liable for the quality, performance, or behavior of the SAMPLE PROGRAMS. Agilent Technologies especially disclaims that the operation of the SAMPLE PROGRAMS shall be uninterrupted or error free. The SAMPLE PROGRAMS are provided AS IS.

AGILENT TECHNOLOGIES DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Agilent Technologies shall not be liable for any infringement of any patent, trademark, copyright, or other proprietary rights by the SAMPLE PROGRAMS or their use. Agilent Technologies does not warrant that the SAMPLE PROGRAMS are free from infringements of such rights of third parties. However, Agilent Technologies will not knowingly infringe or deliver software that infringes the patent, trademark, copyright, or other proprietary right of a third party.

Manual Printing History

The manual's printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates that are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

September 1997 .... First Edition (part number: 04395-90001)

March 1998 .... Second Edition (part number: 04395-90011)

March 2000 .... Third Edition (part number: 04395-90011)

July 2001 .... Fourth Edition (part number: 04395-90021)

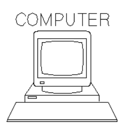
December 2002 .... Fifth Edition (part number: 04395-90031)

Typeface Conventions

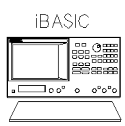
Bold	Boldface type is used when a term is defined. For example: **icons** are symbols.
Italics	Italic type is used for emphasis and for titles of manuals and other publications. Italic type is also used for keyboard entries when a name or a variable must be typed in place of the words in italics. For example: `copy` *filename* means to type the word `copy`, to type a space, and then to type the name of a file such as `file1`.
`Computer`	Computer font is used for on-screen prompts and messages.
(HARDKEYS)	Labeled keys on the instrument front panel are enclosed in ( ).
SOFTKEYS	Softkeys located to the right of the LCD are enclosed in .

Graphic Symbols

General definitions of other graphic symbols used in manuals.



COMPUTER denotes information for a programmer using an external computer as the system controller.



iBASIC denotes information for a programmer using an analyzer with HP Instrument BASIC as the system controller.

Documentation Map

The following manuals are available for the analyzer.

Operation Manual (Agilent Part Number 04395-900x0)

The Operation Manual describes all function accessed from the front panel keys and softkeys. It also provides information on options and accessories available, specifications, system performance, and some topics about the analyzer's features.

Programming Manual (Agilent Part Number 04395-900x1)

The Programming Manual shows how to write and use BASIC program to control the analyzer and describes how HP Instrument BASIC works with the analyzer..

HP Instrument BASIC Users Handbook (Agilent Part Number 04155-90151)

The HP Instrument BASIC User's Handbook introduces you to the HP Instrument BASIC programming language, provide some helpful hints on getting the most use from it, and provide a general programming reference. It is divided into three books, *HP Instrument BASIC Programming Techniques*, *HP Instrument BASIC Interface Techniques*, and *HP Instrument BASIC Language Reference*.

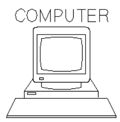
Service Manual (Option 0BW only), (Agilent Part Number 04395-901x0)

The Service Manual explains how to adjust, troubleshoot, and repair the instrument. This manual is option 0BW only.

The number indicated by "x" in the part number of each manual, is allocated for numbers increased by one each time a revision is made. The latest edition comes with the product.

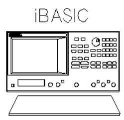
Sample Program Disks

Two sample program disks (Agilent Part Number 04395-180x0) are furnished with 4395A. The disks contain the sample programs listed in this manual. The number indicated by "x" in the part number of each manual, is allocated for numbers increased by one each time a revision is made. The latest edition comes with the product.



Sample program disk for external controller (Disk 1 of 2)

This disk contains the programs for the users who work mainly on the external controller.



Sample program disk for HP Instrument BASIC (Disk 2 of 2)

This disk contains the programs for the users who work mainly on the 4395A using HP Instrument BASIC.

Contents

C. Commands in Entry Block C (*C included)

D. Commands in Entry Block D

E. Commands in Entry Block E (∗E included)

X. Commands in Entry Block X

Z. Commands in Entry Block Z (Other commands included)

Messages

Index

Figures

Tables

1

Introduction

Document Concepts and Usage

This manual provides an introduction to writing BASIC programs for the 4395A Network/Spectrum/Impedance Analyzer (analyzer). To reduce the time required for you to learn how to write programs for the analyzer,the examples shown in this guide are supplied on sample disks. You can perform each example sequentially or you can select the examples that apply to your immediate needs and learn those techniques. Use the table of contents and the index to quickly locate these examples.

Also, depending upon your experience in writing BASIC programs using GPIB commands, you may want to do one of the following:

1. If you are an experienced programmer and have programmed GPIB systems before, you can scan the examples in this guide to find out how the analyzer can be used in your system.
2. If you are not experienced in programming for GPIB instruments, we recommend you to read this manucal from the beggining. Chapter 1 will help you greatly, providing programming basics.
3. Sample programs will give you a hint on how to use an GPIB command in your program.

   For detailed information on an GPIB command, see Chapter 11.
4. The 4395A provides the HP Instrument BASIC feature. If you use the HP Instrument BASIC for the first time, see Chapter 9 which describes the usage of the feature.

Overview of the GPIB Remote Control System

This chapter provides information on how to configure the GPIB remote-control system and the basic use of the GPIB commands. In the examples used in this manual, most of the commands are the simple GPIB commands. Note that no SCPI commands are available for the 4395A. Chapter 11 describes all the GPIB commands that are available for the 4395A.

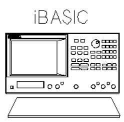
What is GPIB?

The General Purpose Interface Bus (GPIB) is used for remote control of the 4395A Network/Spectrum Analyzer (analyzer). GPIB is a standard for interfacing instruments to computers and peripherals. This standard supports worldwide standards IEEE 488.1, IEC-625, and IEEE 488.2. The GPIB interface allows the analyzer to be controlled by an external computer. The computer sends commands or instructions to and receives data from the instrument through the GPIB.

Required Equipment

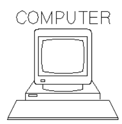
To perform the examples in this manual, you need the following equipment:

1. The analyzer and the accessories required to test a specific device under test (DUT).
2. For the GPIB system controller,



If the analyzer has the HP Instrument BASIC installed, it can be used as the system controller.

Or,



An HP Vectra PC with GPIB interface card (82341D etc.) or an HP 9000 Series 700 computer with GPIB interface card (E2071D etc.). For any computer, you need an GPIB control software, for example HP BASIC for windows. (You can use HP 9000 Series 200/300 computer, too).

3. Peripherals (printer, plotter, and so on) and any GPIB instruments that are required for your application.
4. 10833A/B/C/D GPIB cables to interconnect the computer, the analyzer, and any peripherals.

Controller

In the GPIB terminology, a *controller* is defined to be a device that can permit an GPIB device to *talk* (output data) or *listen* (receive incoming data).

When multiple controllers exist on an GPIB bus, only one of them can be active at a time and can control other devices on the bus. The active controller can issue a `PASS CONTROL` command to pass control to another controller in the same GPIB remote control system.

In a multiple-controller configuration, you can designate one of the controllers as the *system controller*. The system controller becomes active by default when the system power is turned ON. When another controller is serving as the active controller, the system controller can issue an `ABORT` *select code* to become the active controller at any time.

Device Selector

The active controller can control any of the connected GPIB devices. To select which GPIB device to put under its control, the active controller uses the device selector mapped to that target device. Then, the active controller can send various commands to control the behavior or activity of the target device.

1. Use GPIB cables to connect the 4395A with controllers (computers) and peripherals.

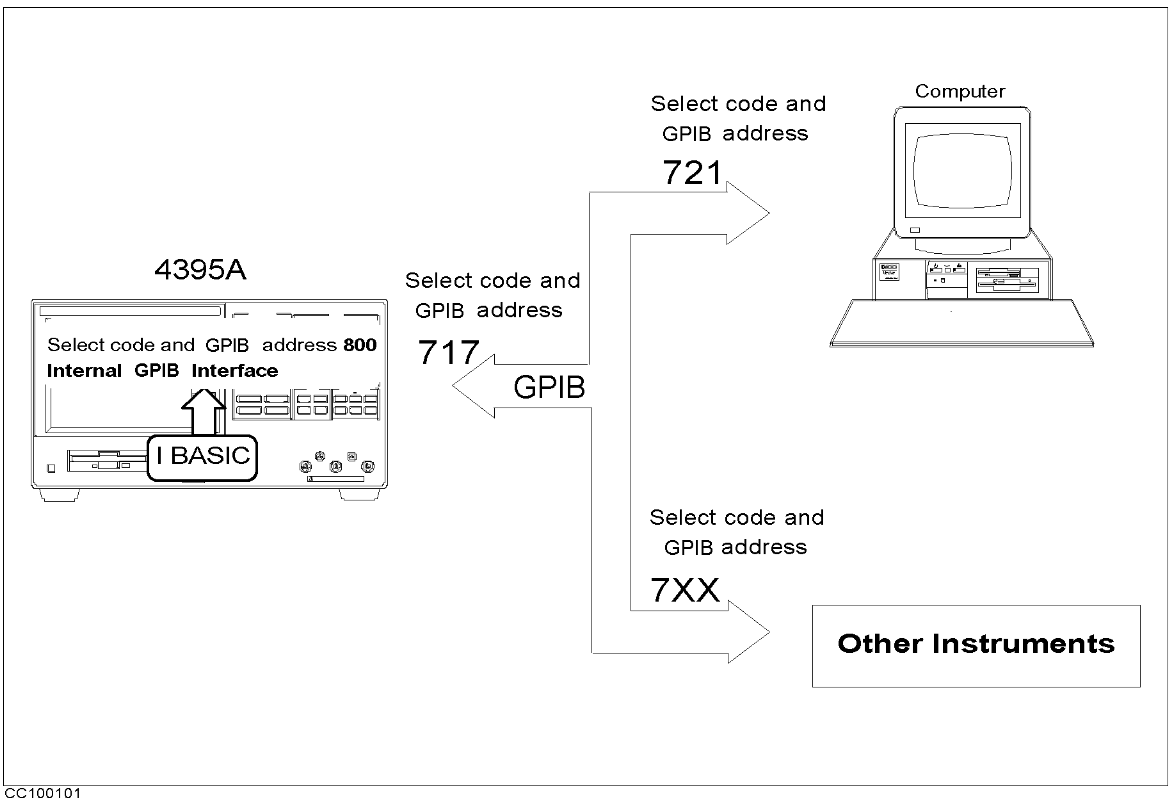


Figure 1-1. Configuration of an GPIB Remote Control System

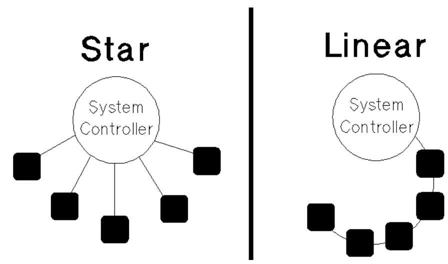
Figure 1-1 illustrates how GPIB addresses are mapped to device selectors. For example, a printer whose GPIB address is “1” is mapped to device selector “701”.

The HP Instrument BASIC feature is internally connected to the 4395A via the built-in interface. The interface select code for the built-in interface is defined as "8." Thus it is distinguished from the external select code "7."

From HP Instrument BASIC, any address ranging from "00" to "30" can be used to designate the analyzer, which is only the device that is connected to the built-in interface. Throughout this manual, the address "00" is always used for the analyzer so that its device selecter is "800."

How large a system can you configure?

- maximum of 15 devices can be connected on one bus system.
- The length of cable between one device and another must be less than or equal to four meters. The total length of cable in one bus system must be less than or equal to two meters timesthe number of devices connected on the bus (the GPIB controller counts as one device). The total length of cable must not exceed 20 meters
- Star, linear, and combinational cable configurations are allowed. There must be no loop.



- It is recommended that no more than four piggyback connectors be stacked together on one device. Otherwise, the resulting structure could exert enough force on the connector mounting to damage it.

Writing and Running Programs

Easy Program Writing

This section serves a simple programming example, which describes procedures required to write and run a program using HP Instrument BASIC. See Chapter 9 for general description of the HP Instrument BASIC and its usage. You can also type in the program without using the BASIC feature.

In this example, the 4395A is set to the condition shown below:

ACTIVE CHANNEL Block	Channel 1 (Default)
MEASUREMENT Block	Network Analyzer A/R LOG MAG Format (Default) Display Scale : Auto
SWEEP Block	Center Frequency : 70MHz Span Frequency : 100kHz

This example requires no keyborad operation;all the procedure can be done by pressing the keys on the front panel.

1. Turn ON the 4395A
2. Press the key and softkeys as shown below to display the softkeys for the network analyzer.

   (Meas) ANALYZER TYPE NETWORK ANALYZER

3. Press

   (System) IBASIC Edit

   The system goes to the edit mode. The cursor is located at the line 10.

```
10 _
```

4. Press

   ASSIGN Hp4395

   This brings the command below at the cursor.

```
10 ASSIGN Hp4395 TO 800_
```

5. Press

   (x1)

This confirms the entry of a command and the cursor moves to the next line.

```
10 ASSIGN Hp4395 TO 800
20 _
```

6. Press

   OUTPUT Hp4395

   You will see the following character strings on the screen:

```
10 ASSIGN Hp4395 TO 800
20 OUTPUT Hp4395""
```

7. Press the key shown below to enter the preset command.

   (Preset)

   At the cursor displayed is the GPIB command ";PRES" which presets an instrument.

```
10 ASSIGN Hp4395 TO 800
20 OUTPUT Hp4395;";PRES"
```

   Then press (x1).

Note

If you place more than one command in a OUTPUT statement, they should be delimited using ";". The delimiter is automatically inserted when you enter GPIB commands with the keys on the front panel.

8. Press the key and softkeys as shown below to specify the measurement parameter to A/R.

   OUTPUT Hp4395 (Meas) ANALYZER TYPE NETWORK ANALYZER RETURN A/R

   This generates the program code as follows:

```
10 ASSIGN Hp4395 TO 800
20 OUTPUT Hp4395;";PRES "
30 OUTPUT Hp4395;";NA;MEAS AR"
```

   Then press (x1).

9. Press the key and softkeys as shown below to specify the center and span frequencies.

   (System) IBASIC OUTPUT Hp4395 (Cent) (7) (0) (M/μ) (Span) (1) (0) (0) (k/m) (x1)

```
10 ASSIGN Hp4395 TO 800
20 OUTPUT Hp4395;";PRES"
30 OUTPUT Hp4395;";NA;MEAS AR"
40 OUTPUT Hp4395;";CENT 70E6;SPAN 100E3"
50 _
```

10. Press the key and softkeys as shown below to auto-scaling.

    System IBASIC OUTPUT Hp4395 Scale Ref AUTO SCALE x1

```
10 ASSIGN Hp4395 TO 800
20 OUTPUT Hp4395;";PRES"
30 OUTPUT Hp4395;";NA;MEAS AR"
40 OUTPUT Hp4395;";CENT 70E6;SPAN 100E3"
50 OUTPUT Hp4395;";AUTO"
60 _
```

11. Enter END command to end the program.

    System IBASIC END x1

```
10 ASSIGN Hp4395 TO 800
20 OUTPUT Hp4395;";PRES"
30 OUTPUT Hp4395;";NA;MEAS AR"
40 OUTPUT Hp4395;";CENT 70E6;SPAN 100E3"
50 OUTPUT Hp4395;";AUTO"
60 END
70 _
```

12. Press the key below to exit the edit mode.

    END EDIT

    The LCD switches back to the measurement results.

Note HP Instrument BASIC cannot be used to program certain operations, such as the procedures for calibration.

Running (Executing) Programs

Press the following key and softkeys from the front panel to execute the program:

(System) IBASIC Run

The system executes the program. You can execute the RUN statement from the keyboard. Type and press as follows:

RUN (Enter)

Saving Programs

Simple procedure for saving a program is shown here. See "Saving Programs (SAVE)" in Chapter 9 for details.

1. Connect the keyboard to the 4395A
2. Press the keys and softkeys as shown below to switch the screen area alloation.

   (Display) MORE DISPLAY ALLOCATION ALL BAISC

3. Select the storage device where you wish to save your program.

   (System) IBASIC MORE MSI[]

4. In the key sequence above, choose [INTERNAL] to save the program on a floppy disk or [MEMORY] on the memory disk.
5. Press the keys and softkeys as shown below.

   (System) IBASIC MORE SAVE

   This gives:

   ```
   SAVE ""
   ```

6. From the keyboard, type in a file name you want to use. Note that the SAVE command does not work if any file that has the same name already resides in the storage. In this case, use a different name or overwrite the file with RE-SAVE command.

Retrieving a Program You Saved

1. Press the keys and softkeys as shown below to switch the screen area alloation.

   (Display) MORE DISPLAY ALLOCATION ALL BAISC

2. Select the storage device which stores the file you wish to retrieve.

   (System) IBASIC MORE MSI[]

3. Press

   (System) IBASIC MORE GET

4. From the keyboard, type in the file name.
5. Use CAT command to list the names of files, if you are not certain an exact name of the file.

2

Programming Measurement Sequence

This chapter provides basic procedures required for programming a measurement sequence. The chaper covers:

- GPIB Commands Overview
- To Program a Basic Measurement

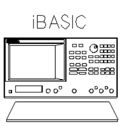
GPIB Commands Overview

All the analyzer's front-panel keys have a corresponding GPIB command. By executing an GPIB command, you can operate the analyzer as if you were pressing the corresponding key.

For example, Pressing (Preset) is the same as executing the GPIB command, `PRES`.

Sending a GPIB Command

Combine the BASIC `OUTPUT` statement with the GPIB select code, the device address, and finally the analyzer command. For example, to execute `PRES` command, type:



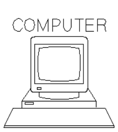
Using HP Instrument BASIC

```
OUTPUT 800;"PRES"
```

8 = Select code (internal GPIB interface); 00 = GPIB address *

* You can set any GPIB address up to 31

And press (Return). The analyzer goes to the preset state.



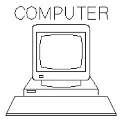
Using an External Controller

```
OUTPUT 717;"PRES"
```

7 = Select code; 17 = GPIB address (same number as you set in page 1-2.)

And press (Return). The analyzer is set to GPIB remote mode. Then the analyzer goes to the preset state.

What is GPIB remote mode?



Executing an `OUTPUT` statement that is addressed to the analyzer, sets it to the GPIB remote mode. In the remote mode, all the analyzer's front-panel keys are locked out, except (Local). Pressing (Local) puts the analyzer back in local mode. In local mode, all front-panel keys are enabled.

Note

In the above example, the OUTPUT statement can be written as follows if you use it with an ASSIGN statement:

```
ASSIGN @Hp4395 TO 800
OUTPUT @Hp4395;"PRES"
```

This style might make modification of a program easier, depending how your program is organized.

To Execute an GPIB Command with a Parameter

Some GPIB commands require a numeric parameter. For example:

```
OUTPUT @Hp4395;"CENT 25000000"   ! Set center frequency to 25 MHz.
```

(The space between the command and the numeric parameter is mandatory.)

You can program it to be entered each time the program is run. For example:

```
100 INPUT "Enter center frequency(Hz).";F_cent
110 OUTPUT @Hp4395;"CENT ";F_cent
```

Executing this gives you:

```
Enter center frequency (Hz).
25000000
```

The analyzer's center frequency is set to 25 MHz.

To Execute a Query

A query command is a command that inquires an instrument informations such as measurement data. In general, a query command is used in an OUTPUT statement with an ENTER statement associated. Note that executing a query command does not always result in a single numerical value; it may contain multiple values or charater strings. See Chapter 11 for details.

```
10 OUTPUT @Hp4395;"CNTS?"
20 ENTER @Hp4395;A
```

Any GPIB command that is used with a numeric parameter can also be used as query command. For example, the `CENT` *numeric_parameter* command used in the previous example, can be combined with a `?`, and used as a query command as follows,

```
10 OUTPUT @Hp4395;"CENT?"
20 ENTER @Hp4395;A
30 PRINT A
```

A query command is used mostly in an OUTPUT statement followed by an ENTER statement;the OUTPUT statement sends the query command to the 4395A and the ENTER statement receive a return value from the analyzer.

The `CENT?` command returns the current center frequency, which is put into `A`. Executing this program results in the following:

```
25000000
```

By interrogating the analyzer to determine the values of the start and stop frequencies, or the center frequency and frequency span, the computer can keep track of the actual frequencies.

To Program a Basic Measurement

This section describes how to organize the commands into a measurement sequence.

Figure 2-1 shows a typical program flow for a measurement.

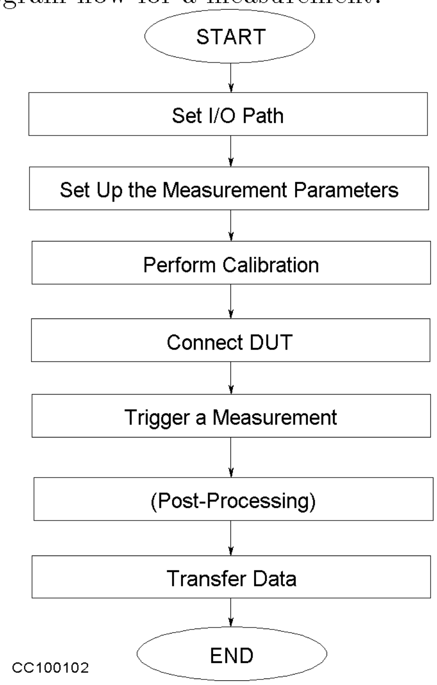
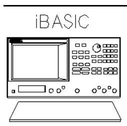


Figure 2-1. Program Flow

The following program performs the measurement flow controlling the analyzer using GPIB.



This guide shows program lists of sample programs for an external controller. To use the sample programs in this guide with HP Instrument BASIC, change the select code from 7 to 8 and change the GPIB address from 17 to 00 (that is, use 800 instead of 717).

```
10    !
20    !Fig.2-2  Basic Measurement
30    !
40    ASSIGN @Hp4395 TO 717 ! When iBASIC is used, change "717" to "800".
50    !
60    OUTPUT @Hp4395;"PRES" ! Preset 4395A
70    OUTPUT @Hp4395;"CHAN1;NA;MEAS S21;FMT LOGM"
80    INPUT "Enter center frequency (Hz).",F_cent
90    INPUT "Enter frequency span (Hz).",F_span
100   OUTPUT @Hp4395;"CENT ";F_cent
110   OUTPUT @Hp4395;"SPAN ";F_span
120   !
130   ! Frequency Response Calibration
140   OUTPUT @Hp4395;"CALK N50"       ! Select 50 ohm type-N Cal. kit
150   OUTPUT @Hp4395;"CALI RESP"      ! Select Response cal.
160   OUTPUT @Hp4395;"CLES"           ! Clear all status
170   INPUT "Connect THRU, then press [Enter].",Dum$
180   OUTPUT @Hp4395;"*SRE 4;ESNB 1" ! Set enable STB and ESB
```

Figure 2-2. Sample Program : Basic Measurement (1/2)

```
190   ON INTR 7 GOTO Cal_end      ! \ When iBASIC is used, change "7" to "8".
200   ENABLE INTR 7;2             ! /
210     OUTPUT @Hp4395;"STANC"         ! Measure THRU
220 Calibrating:    GOTO Calibrating
230 Cal_end:    !
240   OUTPUT @Hp4395;"RESPDONE"      ! Calculating cal coefficients
250   OUTPUT @Hp4395;"*OPC?"      ! \ Waiting calculation end
260   ENTER @Hp4395;Dum           ! /
270   DISP "Response cal completed."
280   !
290   ! Measurement
300   INPUT "Connect DUT, then press [Enter].",Dum$
310   OUTPUT @Hp4395;"CLES"    ! Clear all status registers
320   OUTPUT @Hp4395;"*SRE 4;ESNB 1"
330   ON INTR 7 GOTO Sweep_end     ! \ When iBASIC is used,
340   ENABLE INTR 7;2              ! / change "7" to "8"
350     OUTPUT @Hp4395;"SING"    ! Sweep mode is SINGLE
360 Measuring:    GOTO Measuring
370 Sweep_end:    !
380   OUTPUT @Hp4395;"MKR ON"        ! Marker 1 ON
390   OUTPUT @Hp4395;"SEAM MAX"      ! Search MAX
400   OUTPUT @Hp4395;"OUTPMKR?"      ! Output marker value
410   ENTER @Hp4395;Val1,Val2,Swp
420   PRINT "Max val:",Val1;"dB"
430   PRINT "Swp.Prmtr:",Swp;"Hz"
440   END
```

Figure 1-3. Sample Program : Basic Measurement (2/2)

Set I/O Path

```
40    ASSIGN @Hp4395 TO 717  ! When iBASIC is used, change "717" to "800".
```

This operation allows you to use @Hp4395, instead of 717(or 800), as the GPIB address in the program.

Set Up the Measurement Parameters

```
60    OUTPUT @Hp4395;"PRES"
70    OUTPUT @Hp4395;"CHAN1;NA;MEAS S21;FMT LOGM"
80    INPUT "Enter center frequency (Hz).",F_cent
90    INPUT "Enter frequency span (Hz).",F_span
100   OUTPUT @Hp4395;"CENT ";F_cent
110   OUTPUT @Hp4395;"SPAN ";F_span
```

You can execute GPIB commands in the same sequence as key operation. Lines 60 and 70 perform the same operation as pressing Preset Chan 1 Meas ANALYZER TYPE NETWORK ANALYZER S PARAMETERS Trans:FDW S21[B/R] Format LOG MAG.

In general, the procedure for setting up measurements on the analyzer via GPIB follows the same sequence as performing the procedure manually. There is no required order, as long as the desired frequency range, number of points, and power level are set before performing the calibration.

In line 70, several GPIB commands, separated by semicolon, are executed in a line. This is the same as:

```
70    OUTPUT @Hp4395;"CHAN1"
71    OUTPUT @Hp4395;"NA"
72    OUTPUT @Hp4395;"MEAS S21"
73    OUTPUT @Hp4395;"FMT LOGM"
```

In lines 80 to 110 (setting frequency), parameters are required with the GPIB command. To set parameters, see "To Execute an GPIB Command with a Parameter" later in this chapter.

Perform Calibration

```
130   ! Frequency Response Calibration
140   OUTPUT @Hp4395;"CALK N50"       ! Select 50 ohm type-N Cal. kit
150   OUTPUT @Hp4395;"CALI RESP"      ! Select Response cal.
160   OUTPUT @Hp4395;"CLES"           ! Clear all status
170   INPUT "Connect THRU, then press [Return].",Dum$
180   OUTPUT @Hp4395;"*SRE 4;ESNB 1" ! Set enable STB and ESB
190   ON INTR 7 GOTO Cal_end     ! \ When iBASIC is used, change "7" to "8".
200   ENABLE INTR 7;2            ! /
210     OUTPUT @Hp4395;"STANC"         ! Measure THRU
220 Calibrating:    GOTO Calibrating
230 Cal_end:    !
240   OUTPUT @Hp4395;"RESPDONE"      ! Calculating cal coefficients
250   OUTPUT @Hp4395;"*OPC?"      ! \ Waiting calculation end
260   ENTER @Hp4395;Dum           ! /
270   DISP "Response cal completed."
```

In lines 140 to 240, the GPIB program follows the key strokes required to calibrate from the front panel. This program performs a response calibration.

Line 170 requests the operator to connect a THRU calibration standard.

Lines 180 through 220 use the status bytes to detect the completion of the THRU calibration. See "To Wait for Sweep End" in Chapter 5.

All the setting and calibration procedures are completed. Now you can start measuring your DUT.

Connect DUT

```
300   INPUT "Connect DUT, then press [Enter].",Dum$
```

Line 300 requests the operator to connect a DUT to the analyzer.

Trigger a Measurement

```
310   OUTPUT @Hp4395;"CLES"   ! Clear all status registers
320   OUTPUT @Hp4395;"*SRE 4;ESNB 1"
330   ON INTR 7 GOTO Sweep_end    ! \ When iBASIC is used,
340   ENABLE INTR 7;2             ! / change "7" to "8"
350     OUTPUT @Hp4395;"SING"   ! Sweep mode is SINGLE
360 Measuring:    GOTO Measuring
370 Sweep_end:   !
```

Lines 310 to 370 enable SRQ interruption for sweep end detection. For details, see "To Wait for Sweep End" in Chapter 5.

In line 350, the analyzer executed a single trigger. For more advanced trigger control, see Chapter 6.

Post-Processing

```
380   OUTPUT @Hp4395;"MKR ON"      ! Marker 1 ON
390   OUTPUT @Hp4395;"SEAM MAX"     ! Search MAX
```

Line 380 activates the marker and line 390 moves the marker to the maximum value on the trace. For details on using the marker, see Chapter 3.

Transfer Data

```
400   OUTPUT @Hp4395;"OUTPMKR?"  ! Output marker value
410   ENTER @Hp4395;Val1,Val2,Swp
```

The measured data is transferred to the controller. For details about data transfer, see Chapter 3.

3

Processing and Transferring Data

This chapter illustrates how to transfer and receive data stored in the 4395A between the controller via GPIB.

Measurement data can be read out of the analyzer in the following ways:

1. The entire trace (or data for a specified number of points) can be read out in the following ways:

   - Data arrays — In regard to the data processing flow, the following data arrays are available.

     RAW DATA ARRAYS
     CALIBRATION COEFFICIENT ARRAYS
     DATA ARRAYS
     MEMORY ARRAYS
     DATA TRACE ARRAYS
     MEMORY TRACE ARRAYS

Note

When you transfer these data to 4395A, set 4395A's configuration the same way as when you received data. Without this, you will be unable to measure correct data. Be sure to set up CALIBRATION COEFFICIENT ARRAYS.

   - Data format — The analyzer provides four data transfer formats.

     FORM2 IEEE 32 bit floating point format
     FORM3 IEEE 64 bit floating point format
     FORM4 ASCII format
     FORM5 MS-DOS® personal computer format

     Depending on the format, the data transfer speed and the number of digits are changed. Generally, binary data transfer (FORM2, FORM3, or FORM5) is faster than ASCII (FORM4).

2. Data can be read off the trace selectively using the markers.

   The present value of the marker (real-imaginary data and sweep parameter) is retrieved. For additional information on the marker functions, see Chapter 8.

Data Processing Flow

This section gives an overview of the data processing flow in the 4395A and describes commands to write data in the data arrays.

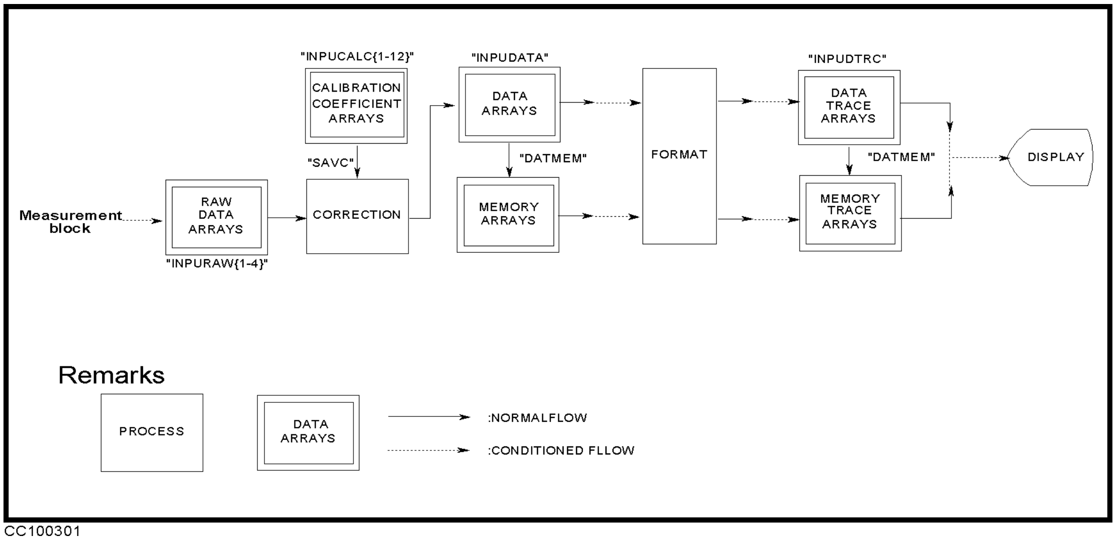


Figure 3-1. Data Arrays, Data Processing, and GPIB Command

- Reset command

`*RST` or `PRES` command clears all arrays.

- Data array writing command

`INPURAW{1-4}`, `INPUDATA` and `INPUDTRC` commands write the corresponding arrays. These commands immediately reshape the data trace on the analyzer's display.

`INPUCALC{1-12}` commands write the CALIBRATION COEFFICIENT ARRAYS.

- DATA to MEMORY command

`DATMEM` command restores the contents in DATA ARRAYS into MEMORY ARRAYS, and the contents in DATA TRACE ARRAYS into MEMORY TRACE ARRAYS.

- Data processing command

`SAVC` command executes the data processing CORRECTION with the current RAW ARRAYS and CALIBRATION COEFFICIENT ARRAYS.

The following examples show how to modify the DATA ARRAYS and DATA TRACE ARRAYS. See "Data Levels" for further information on the data arrays in Figure 3-1.

To Modify Calibration Data

```
10     !
20     !Fig.3-2  To Modify Calibration Data
30     !
40     ASSIGN @Hp4395 TO 717  ! When iBASIC is used, change "717" to "800"
50     !
60     OUTPUT @Hp4395;"PRES"
70     OUTPUT @Hp4395;"NA"
80     INPUT "Enter center frequency(Hz).",F_cent
90     INPUT "Enter frequency span(Hz).",F_span
100    OUTPUT @Hp4395;"CENT ";F_cent
110    OUTPUT @Hp4395;"SPAN ";F_span
120    OUTPUT @Hp4395;"HOLD"
130    !
140    ! Calibration
150    OUTPUT @Hp4395;"CLES"
160    OUTPUT @Hp4395;"*SRE 4;ESNB 1"    ! Set enable STB and ESB
170    INPUT "Connect THRU and press [RETURN] to do CAL.",Dum$
180    OUTPUT @Hp4395;"CALI RESP"
190    ON INTR 7 GOTO Cal_end    ! \ When iBASIC is used,
200    ENABLE INTR 7;2           ! / change "7" to "8"
210      OUTPUT @Hp4395;"STANC"          ! Measure THRU
220 Calibrating:GOTO Calibrating
230 Cal_end:     !
240    OUTPUT @Hp4395;"RESPDONE"         ! Calculating cal coefficient
250    OUTPUT @Hp4395;"*OPC?"         ! \ Wait until calculating ends
260    ENTER @Hp4395;Dum              ! /
270    DISP "Calibration Complete"
280    !
290    !  Read Calibration Data
300    DIM Dat(1:801,1:2)   ! When iBASIC is used, change "801" to "802"
310    OUTPUT @Hp4395;"POIN?"  ! \
320    ENTER @Hp4395;Nop       ! | When iBASIC is used,delete these lines
330    REDIM Dat(1:Nop,1:2)    ! /
340    ASSIGN @Dt TO 717;FORMAT OFF     ! When iBASIC is used,
350    OUTPUT @Hp4395;"FORM3"           ! change "717" to "800"
360    OUTPUT @Hp4395;"OUTPCALC1?"
370    ENTER @Dt USING "%,8A";Head$
380    ENTER @Dt;Dat(*)
390    ENTER @Dt USING "%,1A";Dum$  ! When iBASIC is used, delete this line
400 !
410 !   Modify Calibration Data
420 !
430    !   Restore Calibration  Data
440    OUTPUT @Hp4395;"INPUCALC1 ";
450    OUTPUT @Dt USING "#,8A";Head$
460    OUTPUT @Dt;Dat(*),END
470    ASSIGN @Dt TO *
480    OUTPUT @Hp4395;"SAVC" ! Redraw Trace
490    END
```

Figure 3-2. Sample Program : To Modify Calibration Data

This program measures calibration standards, reads the obtained calibration data, and restores the data in the analyzer.

Read Error-Corrected Data

```
290     !  Read Calibration Data
300     DIM Dat(1:801,1:2)   ! When iBASIC is used, change "801" to "802"
310     OUTPUT @Hp4395;"POIN?"
320     ENTER @Hp4395;Nop
330     REDIM Dat(1:Nop,1:2)        ! | When iBASIC is used, delete these lines
340     ASSIGN @Dt TO 717;FORMAT OFF
350     OUTPUT @Hp4395;"FORM3"
360     OUTPUT @Hp4395;"OUTPCALC1?"
370     ENTER @Dt USING "%,8A";Head$
380     ENTER @Dt;Dat(*)
390     ENTER @Dt USING "%,1A";Dum$  ! | When iBASIC is used, delete these lines
```

The controller can read out the error coefficients using the GPIB commands `OUTPCALC{1-12}`. Each point is a real/imaginary pair, and the number of points in the array is the same as the number of points in the sweep. For details on data transfer, see Chapter 3.

Each calibration type uses only as many arrays as needed, starting with array 1, and each array stores a specific error coefficient. Therefore, it is necessary to know the type of calibration about to be read out: attempting to read an array not being used in the current calibration causes the "REQUESTED DATA NOT CURRENTLY AVAILABLE" warning to be displayed. For assignment of data arrays, see "Calibration Types and Standard Classes, and Calibration Arrays" in this chapter.

Modify Calibration Data

```
400 !
410 !   Modify Calibration Data
420 !
```

In this portion of program, you modify the CALIBRATION COEFFICIENT ARRAY, which is contained in `Dat(1:801,1:2)`.

Restore Modified Calibration Data

```
430     !   Restore Calibration  Data
440     OUTPUT @Hp4395;"INPUCALC1 ";
450     OUTPUT @Dt USING "#,8A";Head$
460     OUTPUT @Dt;Dat(*),END
```

Line 440 opens the CALIBRATION COEFFICIENT ARRAY 1 in the analyzer. This array is used to restore the data.

Lines 450 and 460 send the file header (`Head$`), calibration data (`Dat(*)`) and the terminator (`END`). The file header is an input in line 370.

This example sets the trigger to HOLD at line 120. The analyzer does not redraw the trace with the new CALIBRATION COEFFICIENT ARRAYS when the trigger is set to HOLD. You can redraw the trace by issuing the GPIB command `SAVC`.

To Modify Error-Corrected Data

```
10    !
20    !Fig.3-3  To Modify Error-Corrected Data
30    !
40    ASSIGN @Hp4395 TO 717  ! When iBASIC is used, change "717" to "800"
50    !
60    OUTPUT @Hp4395;"PRES"
70    OUTPUT @Hp4395;"NA"
80    INPUT "Enter center frequency(Hz).",F_cent
90    INPUT "Enter frequency span(Hz).",F_span
100   OUTPUT @Hp4395;"CENT ";F_cent
110   OUTPUT @Hp4395;"SPAN ";F_span
120    !
130   INPUT "Connect DUT and press [RETURN].",Dum$
140   OUTPUT @Hp4395;"CLES"
150   OUTPUT @Hp4395;"*SRE 4;ESNB 1"    ! Set enable STB and ESB
160   ON INTR 7 GOTO Sweep_end  ! \ When iBASIC is used,
170   ENABLE INTR 7;2           ! / change "7" to "8"
180     OUTPUT @Hp4395;"SING"
190 Measuring:     GOTO Measuring
200 Sweep_end:     !
210   DISP "Measurement Complete"
220   !
230   !  Read Error-Corrected Data
240   DIM Dat(1:801,1:2)   ! When iBASIC is used, change "801" to "802"
250   OUTPUT @Hp4395;"POIN?"  ! \
260   ENTER @Hp4395;Nop       ! | When iBASIC is used, delete these lines
270   REDIM Dat(1:Nop,1:2)    ! /
280   ASSIGN @Dt TO 717;FORMAT OFF   ! When iBASIC is used,
290   OUTPUT @Hp4395;"FORM3"         ! change "717" to "800"
300   OUTPUT @Hp4395;"OUTPDATA?"
310   ENTER @Dt USING "%,8A";Head$
320   ENTER @Dt;Dat(*)
330   ENTER @Dt USING "%,1A";Dum$    ! When iBASIC is used, delete this line
340 !
350 !   Modify Error-Corrected Data
360 !
370   !   Restore Error-Corrected Data
380   OUTPUT @Hp4395;"INPUDATA ";
390   OUTPUT @Dt USING "#,8A";Head$
400   OUTPUT @Dt;Dat(*),END
410   ASSIGN @Dt TO *
420   END
```

Figure 3-3. Sample Program : To Modify Error-Corrected Data

This program measures the DUT, reads the obtained data, and restores the data in the analyzer.

Read Error-Corrected Data

```
230   !  Read Error-Corrected Data
240   DIM Dat(1:801,1:2)   ! When iBASIC is used, change "801" to "802"
250   OUTPUT @Hp4395;"POIN?"  ! \
260   ENTER @Hp4395;Nop       ! | When iBASIC is used, delete these lines
270   REDIM Dat(1:Nop,1:2)    ! /
280   ASSIGN @Dt TO 717;FORMAT OFF   ! When iBASIC is used,
290   OUTPUT @Hp4395;"FORM3"         ! change "717" to "800"
300   OUTPUT @Hp4395;"OUTPDATA?"
310   ENTER @Dt USING "%,8A";Head$
320   ENTER @Dt;Dat(*)
330   ENTER @Dt USING "%,1A";Dum$    ! When iBASIC is used, delete this line
```

`OUTPDATA?` command retrieves DATA ARRAYS in the analyzer.

Restore Modified Error-Corrected Data

```
370   !   Restore Error-Corrected Data
380   OUTPUT @Hp4395;"INPUDATA ";
390   OUTPUT @Dt USING "#,8A";Head$
400   OUTPUT @Dt;Dat(*),END
```

Line 380 opens the DATA ARRAYS in the analyzer to restore the data.

Lines 390 to 400 transfer data in FORM3 (a similar procedure is used in the “To Modify Calibration Data” example).

To Modify Trace Data

```
10    !
20    !Fig.3-4  To Modify Trace Data
30    !
40    ASSIGN @Hp4395 TO 717  ! When iBASIC is used, change "717" to "800"
50    !
60    OUTPUT @Hp4395;"PRES"
70    OUTPUT @Hp4395;"NA"
80    INPUT "Enter center frequency(Hz).",F_cent
90    INPUT "Enter frequency span(Hz).",F_span
100   OUTPUT @Hp4395;"CENT ";F_cent
110   OUTPUT @Hp4395;"SPAN ";F_span
120    !
130   INPUT "Connect DUT and press [RETURN].",Dum$
140   OUTPUT @Hp4395;"CLES"
150   OUTPUT @Hp4395;"*SRE 4;ESNB 1"    ! Set enable STB and ESB
160   ON INTR 7 GOTO Sweep_end    ! \ When iBASIC is used,
170   ENABLE INTR 7;2             ! / change "7" to "8"
180     OUTPUT @Hp4395;"SING"
190 Measuring:     GOTO Measuring
200 Sweep_end:     !
210   DISP "Measurement Complete"
220   !
230   !  Read Trace Data
240   DIM Dat(1:801,1:2)  ! When iBASIC is used, change "801" to "802"
250   OUTPUT @Hp4395;"POIN?"  ! \
260   ENTER @Hp4395;Nop       ! | When iBASIC is used, delete these lines
270   REDIM Dat(1:Nop,1:2)    ! /
280   ASSIGN @Dt TO 717;FORMAT OFF   ! When iBASIC is used,
290   OUTPUT @Hp4395;"FORM3"         ! change "717" to "800"
300   OUTPUT @Hp4395;"OUTPDTRC?"
310   ENTER @Dt USING "%,8A";Head$
320   ENTER @Dt;Dat(*)
330   ENTER @Dt USING "%,1A";Dum$  ! When iBASIC is used, delete this line
340 !
350 !   Modify Trace Data
360 !
370   !   Restore Trace Data
380   OUTPUT @Hp4395;"INPUDTRC ";
390   OUTPUT @Dt USING "#,8A";Head$
400   OUTPUT @Dt;Dat(*),END
410   ASSIGN @Dt TO *
420   END
```

Figure 3-4. Sample Program : To Modify Trace Data

This program measures the DUT, reads the obtained data, and restores the data into the analyzer. For details on how to read the data array, see Chapter 3.

For details on how to modify the trace on the display, see the “To Modify Calibration Data” example.

Read Trace Data

```
230   !  Read Trace Data
240   DIM Dat(1:801,1:2)  ! When iBASIC is used, change "801" to "802"
250   OUTPUT @Hp4395;"POIN?"  ! \
260   ENTER @Hp4395;Nop      ! | When iBASIC is used, delete these lines
270   REDIM Dat(1:Nop,1:2)    ! /
280   ASSIGN @Dt TO 717;FORMAT OFF   ! When iBASIC is used,
290   OUTPUT @Hp4395;"FORM3"         ! change "717" to "800"
300   OUTPUT @Hp4395;"OUTPDTRC?"
310   ENTER @Dt USING "%,8A";Head$
320   ENTER @Dt;Dat(*)
330   ENTER @Dt USING "%,1A";Dum$  ! When iBASIC is used, delete this line
```

The `OUTPDTRC?` command (line 300) retrieves trace data in the analyzer. For details on data transfer, see "To Get Measurement Data Using ASCII Format" and Figure 3-6.

Restore Modified Trace Data

```
370   !   Restore Trace Data
380   OUTPUT @Hp4395;"INPUDTRC ";
390   OUTPUT @Dt USING "#,8A";Head$
400   OUTPUT @Dt;Dat(*),END
```

Line 380 opens the DATA TRACE ARRAYS in the analyzer to restore the data.

Lines 390 and 400 transfer data in FORM3 (a similar procedure is used in the "To Modify Calibration Data" example).

To Get Measurement Data Using ASCII Format

This section provides procedures required for getting data using ASCII format, taking a measurement trace as an example.

```
10    !
20    !Fig.3-5  To Get Measurement Trace Using ASCII Format
30    !
40    ASSIGN @Hp4395 TO 717  ! When iBASIC is used, change "717" to "800"
50    !
60    INPUT "ENTER CENTER FREQUENCY (Hz)",F_cent
70    INPUT "ENTER FREQUENCY SPAN (Hz)",F_span
80    OUTPUT @Hp4395;"CENT";F_cent
90    OUTPUT @Hp4395;"SPAN";F_span
100   !
110   OUTPUT @Hp4395;"CLES"
120   OUTPUT @Hp4395;"*SRE 4;ESNB 1"
130   ON INTR 7 GOTO Sweep_end  ! \ When iBASIC is used, change "7" to "8"
140   ENABLE INTR 7;2           ! /
150     OUTPUT @Hp4395;"SING"       ! Trigger a Measurement
160 Measuring:    GOTO Measuring    ! Measuring
170 Sweep_end:   !
180     DIM Dat(1:801,1:2),Swp(1:801)  ! For spectrum measurement, change
190     OUTPUT @Hp4395;"FORM4"         ! "Dat(1:801,1:2)" to "Dat(1:801)"
200     OUTPUT @Hp4395;"OUTPDTRC?"
210     ENTER @Hp4395 USING "%,K";Dat(*)
220     OUTPUT @Hp4395;"OUTPSWPRM?"
230     ENTER @Hp4395 USING "%,K";Swp(*)
240     !
250     OUTPUT @Hp4395;"POIN?"
260     ENTER @Hp4395;Nop
270     FOR I=1 TO Nop
280       PRINT Swp(I);"Hz",Dat(I,1);"dB" ! For spectrum measurement, change
290     NEXT I                            ! "Dat(I,1)" to "Dat(I)"
300     END
```

Figure 3-5. Sample Program : To Get Measurement Trace Using ASCII Format

Set the Receive Array

```
180      DIM Dat(1:801,1:2),Swp(1:801)
```

Line 180 sets the array size to the analyzer's maximum number of measurement points (801).

In this example, it is assumed that the analyzer is in the network analyzer mode of operation, in which each point has complex data. If you use the analyzer in the spectrum analyzer mode, each measurement point has only real data, so you must set the data array `Dat` as follows:

```
180      DIM Dat(1:801),Swp(1:801)

280      PRINT Swp(I);"Hz",Dat(I);"dB"
```

If the number of measurement points changes, then so does the number of data. You must control the number of entered measurement data (see lines 210 and 230).

Set Data Transfer Format

```
190     OUTPUT @Hp4395;"FORM4"
```

Line 190 tells the analyzer to use the ASCII transfer format.

Read Data

```
200     OUTPUT @Hp4395;"OUTPDTRC?"
210     ENTER @Hp4395 USING "%,K";Dat(*)
220     OUTPUT @Hp4395;"OUTPSWPRM?"
230     ENTER @Hp4395 USING "%,K";Swp(*)
```

`OUTPDTRC?` retrieves DATA TRACE ARRAYS, and `OUTPSWPRM?` retrieves sweep parameters.

In line 210 and 230, you must choose `%,K` to allow for an insufficient number of data points to fill the array (which is 801 as declared in line 180).

What are other data arrays?

You can retrieve the following data arrays, exchanging GPIB command `OUTPDTRC?` in line 200. For details on each command, see Chapter 11.

- RAW DATA ARRAYS — `OUTPRAW{1-4}?`
- DATA ARRAYS — `OUTPDATA?`
- MEMORY ARRAYS — `OUTPMEMO?`
- MEMORY TRACE ARRAYS — `OUTPMTRC?`
- CALIBRATION COEFFICIENT ARRAYS — `OUTPCALC{1-12}?`

To Get Measurement Trace Using Binary Format

This section provides procedures required for getting data using binary format, taking a measurement trace as an example.

Before running the program in Figure 3-6, you must modify the dimension of the data arrays to match to the analyzer type (network or spectrum). (See the "Set the Receive Array" example.)

```
10  !
20  !Fig.3-6  To Get Measurement Trace Using
30  !             IEEE 64-bit Floating point Format (For External Controller)
40  !
50    ASSIGN @Hp4395 TO 717
60    !
70    INPUT "ENTER CENTER FREQUENCY (Hz)",F_cent
80    INPUT "ENTER FREQUENCY SPAN (Hz)",F_span
90    OUTPUT @Hp4395;"CENT";F_cent
100   OUTPUT @Hp4395;"SPAN";F_span
110   !
120   OUTPUT @Hp4395;"CLES"
130   OUTPUT @Hp4395;"*SRE 4;ESNB 1"
140   ON INTR 7 GOTO Sweep_end   !
150   ENABLE INTR 7;2            !
160     OUTPUT @Hp4395;"SING"
170 Measuring:GOTO Measuring
180 Sweep_end:   !
190   DIM Dat(1:801,1:2),Swp(1:801)   ! For spectrum measurement, change
200   OUTPUT @Hp4395;"POIN?"          ! "Dat(1:801,1:2)" to "Dat(1:801)"
210   ENTER @Hp4395;Nop
220   REDIM Dat(1:Nop,1:2),Swp(1:Nop)
230   OUTPUT @Hp4395;"FORM3"
240   ASSIGN @Dt TO 717;FORMAT OFF
250   OUTPUT @Hp4395;"OUTPDTRC?"
260   ENTER @Dt USING "%,8A";A$
270   ENTER @Dt;Dat(*)
280   ENTER @Dt USING "%,1A";B$
290   OUTPUT @Hp4395;"OUTPSWPRM?"
300   ENTER @Dt USING "%,8A";A$
310   ENTER @Dt;Swp(*)
320   ENTER @Dt USING "%,1A";B$
330   ASSIGN @Dt TO *
340   !
350   FOR I=1 TO Nop
360     PRINT Swp(I);"Hz",Dat(I,1);"dB"   ! For spectrum measurement, change
370   NEXT I                              ! "Dat(I,1)" to "Dat(I)"
380   END
```

Figure 3-6.
Sample Program : To Get Measurement Trace Using IEEE 64-bit Floating Point Format (For External Controller)

This program is similar to the ASCII transfer program. However, you must set the data transfer format OFF when using the binary data transfer format.

Set the Receive Array

```
190    DIM Dat(1:801,1:2),Swp(1:801)
200    OUTPUT @Hp4395;"POIN?"
210    ENTER @Hp4395;Nop
220    REDIM Dat(1:Nop,1:2),Swp(1:Nop)
```

Line 190 sets the array size to the analyzer's maximum number of measurement points (801).

In this example, it is assumed that the analyzer is in the network analyzer mode of operation, in which each point has complex data. If you use the analyzer in the spectrum analyzer mode, each measurement point has only real data, so you must set the data array `Dat` as follows:

```
190      DIM Dat(1:801),Swp(1:801)

220      REDIM Dat(1:Nop),Swp(1:Nop)

360      PRINT Swp(I);"Hz",Dat(I);"dB"
```

Lines 200 and 210 interrogate the analyzer to determine the number of measurement points. Line 220 resizes the receive array to match the data.

Set Data Transfer Format

```
200    OUTPUT @Hp4395;"FORM3"
210    ASSIGN @Dt TO 717;FORMAT OFF
```

To use FORM3 the computer must be instructed to stop formatting the incoming data with the `ENTER` statement. This is done by defining an I/O path with ASCII formatting OFF. The I/O path points to the analyzer. This path can be used to read or write data to the analyzer, as long as that data is in binary rather than ASCII format.

What are other binary data formats?

You can use the following data transfer formats, by changing the GPIB command `FORM3` in line 200.

- IEEE 32 bit floating point format `FORM2`
- MS-DOS® personal computer format `FORM5`

Read Data

```
250      OUTPUT @Hp4395;"OUTPDTRC?"
260      ENTER @Dt USING "%,8A";A$
270      ENTER @Dt;Dat(*)
280      ENTER @Dt USING "%,1A";B$
290      OUTPUT @Hp4395;"OUTPSWPRM?"
300      ENTER @Dt USING "%,8A";A$
310      ENTER @Dt;Swp(*)
320      ENTER @Dt USING "%,1A";B$
```

FORM3 has an eight-byte header to deal with. The first two bytes are the ASCII characters #6. This indicates that a fixed length block transfer follows and that the next 6 bytes form an integer specifying the number of bytes in the block to follow. The header must be read in so that data order is maintained (lines 260 and 300).

At the data end, the terminator "LF^EOI" is sent(lines 280 and 320).

Data Formats in Binary Transferring

The analyzer can transmit data over GPIB in four different formats. The type of format affects what kind of data array is declared (real or integer), because the format determines what type of data is transferred.

- Form 2

  IEEE 32-bit floating point format. Figure 3-7 shows the data transfer format of Form 2. In this mode, each number takes 4 bytes.

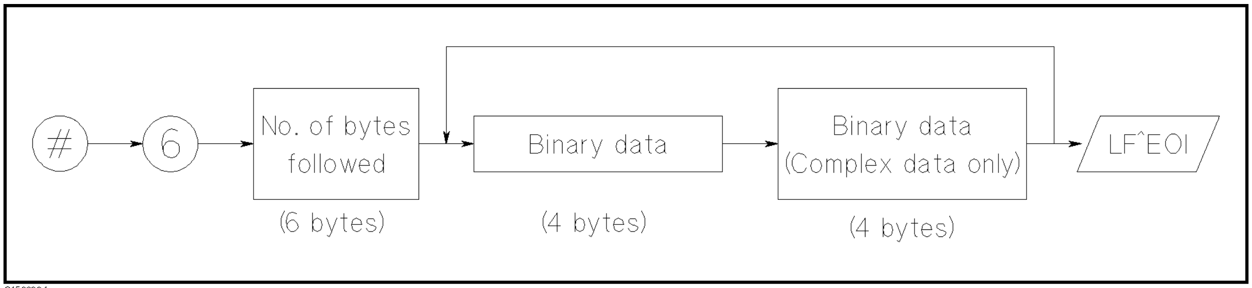


Figure 3-7. Form 2 Data Transfer Format

- Form 3

  IEEE 64-bit floating point format. Figure 3-8 shows the data transfer format of Form 3. Data is stored internally in the 200/300 series computer with the IEEE 64-bit floating point format, eliminating the need for any reformatting by the computer. In this mode, each number takes 8 bytes.

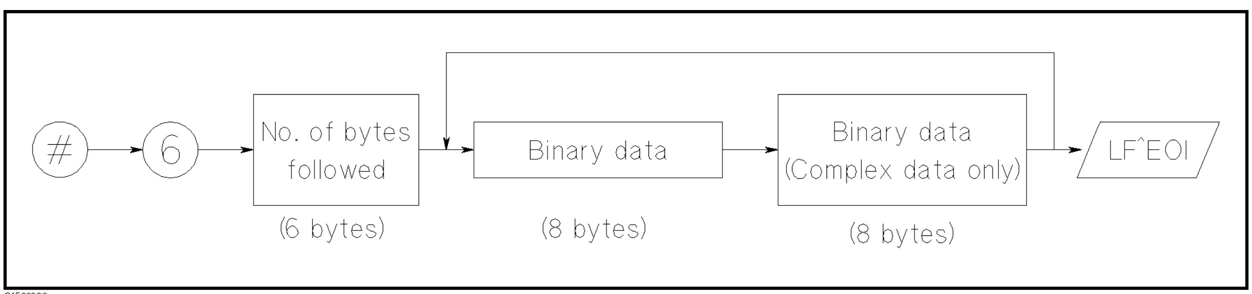


Figure 3-8. Form 3 Data Transfer Format

- Form 4

  ASCII data transfer format. In this mode, each number is sent as a 24-character string, each character being a digit, sign, or decimal point.

- Form 5

  MS-DOS® personal computer format. This mode is a modification of IEEE 32-bit floating point format with the byte order reversed. Form 5 also has a four-byte header that must be read in so that data order is maintained. In this mode, an MS-DOS® PC can store data internally without reformatting it.

File Headers

When using the binary data transfer format, the transferred data must be accompanied by the file header that represents the data length. In this example, the data transfer format is FORM3 and the transferred data is configured as follows:

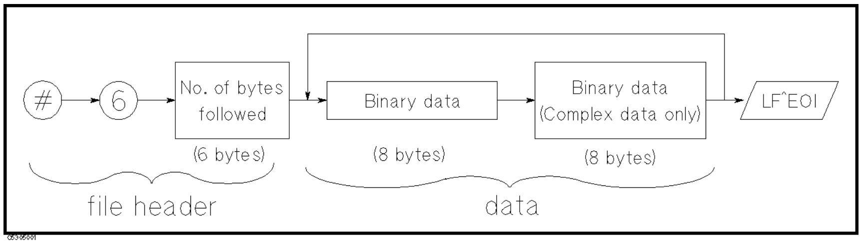


Figure 3-9. FORM3 Data Transfer Format

If you are not reading the header, you can create it using the number of data points. Change the program lines 440 to 460 as follows:

```
440   OUTPUT @Hp4395;"POIN?"
441   ENTER @Hp4395;Nop
442   V$=VAL$(Nop*2*8)
443   Numv=LEN(V$)
444   Head$="000000"
445   FOR I=1 TO Numv
446     Head$[7-I,7-I]=V$[Numv-I+1,Numv-I+1]
447   NEXT I
448   !
449   OUTPUT @Hp4395;"INPUCALC1 ";
450   OUTPUT @Dt USING "#,8A";"#6"&Head$
460   OUTPUT @Dt;Dat(*),END
```

Lines 440 to 442 calculate the number of bytes transferred (8 byte for real part, 8 byte for imaginary part), and represents it in the string format.

Line 443 counts the number of characters in the string that contains the number of bytes transferred.

Line 444 enters 0 as the initial value in all header arrays.

Lines 445 to 447 place the number of bytes transferred to the header array digit by digit from the sixth array to the first array of the header.

For example, if the number of points is 201, the value of `Head$` is `003216`.

Saving a Data File

This section explains how to save data from the 4395A to a storage device. All examples contained herein assume that you intend to output measurement data from the 4395A upon completion of a measurement and then save the data on a floppy disk.

If you wish to use the following sample program with Instrument BASIC, specify in advance a memory disk or floppy disk drive as the target storage device.

```
10  DIM Dat(1:802,1:2)
20  ASSIGN @Hp4395 TO 717
40  OUTPUT @Hp4395;"POIN?"
50  ENTER @Hp4395;Nop
60  REDIM Dat(1:Nop,1:2)
70  OUTPUT @Hp4395;"FORM4"
80  OUTPUT @Hp4395;"OUTPDTRC?"
90  ENTER @Hp4395;Dat(*)
100 !
110 !
120     Fname$="data01"
130     PURGE Fname$
140     CREATE Fname$,1
150     ASSIGN @F TO Fname$;FORMAT ON
160     OUTPUT @F;Dat(*)
170     ASSIGN @F TO *
180     !
190  END
```

Figure 3-10. Saving Data on a Floppy Disk

Creating a File to Contain the Data

```
120     Fname$="data01"
130     PURGE Fname$
140     CREATE Fname$,1
```

The above code creates a file named `data01`. Notice that line 120 deletes any existing file named `data01`.

Opening the File and Transferring the Data

```
150     ASSIGN @F TO Fname$;FORMAT ON
160     OUTPUT @F;Dat(*)
170     ASSIGN @F TO *
```

The above code opens the file in the ASCII format, and transfers the trace data to the file. Line 170 closes the file; this statement is required if you are dealing with multiple files.

Data Levels

The analyzer has the following data arrays in internal memory:

- Raw data

  These arrays store the results of all the preceding data processing operations. Note that the numbers here are still complex pairs.

  When the Network analyzer mode and the full 2-port error correction are on, the raw data arrays contain all four S-parameter measurements required for accuracy enhancement.

- Error corrected data

  The results of error correction are stored in the data arrays as complex number pairs.

- Formatted data

  This is the array of data being displayed. It reflects all post-processing functions such as electrical delay, and the units of the array read out depends on the current display format.

- Calibration coefficients (Network and impedance analyzer only)

  The results of a calibration are stored arrays of calibration coefficients that are used by the error correction routines. Each array corresponds to a specific error term in the error model. The calibration coefficients are read out with `OUTPCALC{1-12}?`.

- fixture compensation coefficients (Impedance analyzer only)

  The results of a fixture compensation are stored arrays of fixture compensation coefficients that are used by the error correction routines. Each array corresponds to a specific error term in the error model. The fixture compensation coefficients are read out with `OUTPCOMC{1-3}?`.

Formatted data is generally the most useful, because it is the same information as that seen on the display. However, if post-processing is not necessary, as may be the case with smoothing, error corrected data is more desirable. Error corrected data also gives you the opportunity to load the data into the instrument and apply post-processing at a later time.

Calibration Types and Standard Classes, and Calibration Arrays

Table 3-1 lists which standard classes are required for each calibration type. This table shows that, for example, S_{11}1 port calibration requires three calibrations;S11A(OPEN), S11B(SHORT), and S11C(LOAD).

Table 3-1. Calibration Types and Standard Classes

Class	Response	Response and Isolation	S_{11} 1-port	S_{22} 1-port	One-path 2-port	Full 2-port	ZA calibration
Response:	•						
Response and isolation:							
Response		•					
Isolation		•					
Reflection:[1]					•	•	
S11A (opens)			•		•	•	
S11B (shorts)			•		•	•	
S11C (loads)			•		•	•	
S22A (opens)				•		•	
S22B (shorts)				•		•	
S22C (loads)				•		•	
Transmission:[1]					•	•	
Forward match					•	•	
Forward thru					•	•	
Reverse match						•	
Reverse thru						•	
Isolation:[1]					•	•	
Forward					•	•	
Reverse						•	
Impedance analyzer cal							
IMPA (OPEN)							•
IMPB (SHORT)							•
IMPC (LOAD)							•

1 These subheadings must be called when doing 2-port calibrations.

Table 3-2 specifies where the calibration coefficients are stored for different calibration types.

Table 3-2. Calibration Array

Array	Response[1]	Response and Isolation[1]	1-port[1] ZA cal	2-port[1 2]
1	E_R or E_T	E_X (E_D)[3]	E_D	E_{DF}
2		E_T (E_R)	E_S	E_{SF}
3			E_R	E_{RF}
4				E_{XF}
5				E_{LF}
6				E_{TF}
7				E_{DR}
8				E_{SR}
9				E_{RR}
10				E_{XR}
11				E_{LR}
12				E_{TR}

1 Meaning of first subscript: D=directivity; S=source match; X=crosstalk; L=load match; T=transmission tracking. Meaning of second subscript: F=forward; R=reverse.

2 One path, 2-port cal duplicates arrays 1 to 6 in arrays 7 to 12.

3 Response and isolation corrects for crosstalk and transmission tracking in transmission measurements, and for directivity and reflection tracking in reflection measurements.

The table shows that, for example, for 1 port calibration (or calibration for impedance analyzer mode), E_D (directive error-correction coefficients) is stored in the first element of an array, E_S (source match error-correction coefficients) in the second, and E_R (reverse error-correction coefficients) in the third.

4

Synchronizing the 4395A with a Controller

You can use a program running on a controller to remotely instruct an GPIB compatible device to calibrate itself, output measurement data, perform calculations or other tasks. Implementing such a remote control system, however, requires you to keep the 4395A synchronized with the program execution.
For example, when obtaining measurement data, the controller must wait until the 4395A completes the current measurement process. For calibration, the controller must wait until the 4395A finishes processing the data.

You can synchronize the 4395A with the controller in several ways:

- Use the `*OPC?` command to suspend the controller until the preceding command is completed.
- Use the `WAIT` command of BASIC.
- Use an EXECUTE `SING` statement of Instrument BASIC to suspend the program until the completion of the sweep process.
- Use an SRQ to suspend the external controller until the completion of the sweep process.

Also, you can use a `*WAI` command to suspend the execution of an GPIB command although this technique does not provide synchronization with a controller.

Using the *OPC Command

```
10 !
20 !Fig.4-1  To Wait for the Preceding Operation Complete
30 !
40   ASSIGN @Hp4395 TO 717     ! When iBASIC is used, change "717" to "800"
50   !
60   !  OUTPUT statement to send GPIB command
70   !
80   OUTPUT @Hp4395;"*OPC?"
90   ENTER @Hp4395;A
100  !
110  !  Next operation
120  !
130  END
```

Figure 4-1. Sample program: Wait until the preceding command is completed.

Suspending a Program Running on a Controller (*OPC)

```
80   OUTPUT @Hp4395;"*OPC?"
90   ENTER @Hp4395;A
```

The above code causes the controller to suspend the program execution until all preceding commands are successfully processed and `*OPC?` returns `1`.

In Figure 2-2 (Chapter 1), for example, the `*OPC?` command is used as follows:

```
 :
240   OUTPUT @Hp4395;"RESPDONE"  ! Compute calibration coefficients
250   OUTPUT @Hp4395;"*OPC?"     ! Wait until completed
260   ENTER @Hp4395;Dum
270   DISP "Response cal completed."
 :
```

Using the WAIT Command of BASIC

Using the `WAIT` command of BASIC, you can have the controller wait for a particular period of time. This is a very simple solution, but requires you to accurately measure the time required for the 4395A to actually process a certain command(s).

```
10 OUTPUT 717;"SA"
20 WAIT 5
30 OUTPUT 717;"NA"
40 WAIT 5
50 END
```

Since this technique causes the controller to wait without communicating with the 4395A, an improper wait time setting can result in an unpredictable error. However, using the `WAIT` command can effectively speed up the execution of your program as long as the setting is accurate.

Using the EXECUTE Statement to Synchronize with the Completion of Sweep

```
10 !
20 !Fig.4-2  To Wait for the Preceding Operation Complete
30 !
40   ASSIGN @Hp4395 TO 800     ! Only iBASIC is available
50   !
60   !  OUTPUT statement to send GPIB command
70   !
80   EXECUTE "SING"
90   OUTPUT @Hp4395;"MKR ON"
100  OUTPUT @Hp4395;"SEAM MAX"
110  !  Next operation
120  !
130  END
```

Figure 4-2. Sample program: Wait until the preceding command is completed.

Here is a tip: You can just issue a single EXECUTE "SING" statement to synchronize with the completion of sweep. This is useful, for example, when you want to use the marker after a single cycle of measurement.

```
80   EXECUTE "SING"
```

This causes the controller to wait until one cycle of sweep is completed.

```
90   OUTPUT @Hp4395;"MKR ON"
100  OUTPUT @Hp4395;"SEAM MAX"
```

Because the measurement is already completed, you can effectively use the marker.

Note that the `EXECUTE` command is supported for Instrument BASIC only.

Using SRQ

You can also use an SRQ to implement synchronization with the completion of sweep. This technique is recommended when you are using an external controller. For how to synchronize an SRQ with the completion of a sweep process that uses an SRQ, refer to "To Wait for Sweep End" in Chapter 5.

Using the *WAI Command

Note

If the active controller is an external controller, using the *WAI command cannot perfectly synchronize the controller with the 4395A.

The *WAI command prevents the 4395A from processing any newly received commands until it finishes processing all preceding commands. If the 4395A receives a new GPIB command during the wait, it stores the command in a buffer.

Example)

```
10 OUTPUT 717;"SING"
20 OUTPUT 717;"*WAI"
30 OUTPUT 717;"AUTO"
40 PRINT "Program end"
50 END
```

The above program inhibits the 4395A from processing the AUTO command until the completion of the SING command. However, the controller executes line 40 of the program, whether or not the SING command has been successfully processed. On the other hand, the 4395A does not execute the AUTO command until the completion of the SING command, regardless of the program execution on the controller.

5

Status Reporting System and Processing Generated Interruptions

The analyzer has a status reporting ssytem to report the condition of the analyzer. This chapter gives you an overview of the status reporting system.

This chapter provides the following information:

- General status register model
- Status bit definitions of the Status Byte
- Status bit definitions of the Standard Event Status Register
- Status bit Definitions of the Operation Status Register
- OSPT, OSNT
- Using the Service Request (SRQ)
- Reporting command error occurrence (with sample program)
- Waiting for sweep end (with sample program)

What is an SRQ?

An SRQ (Service Request) is an interrupt generated by the analyzer. The analyzer can be setup to sent an SRQ when it needs the attention of the controller. The controller can ignore the SRQ or it can be setup to interrupt the program using the ON INTR commands. The Status Byte can be used to define the specific event that generates an SRQ (for example, the end of sweep complete).

General Status Register Model

The analyzer has a status reporting system to report the condition of the analyzer.

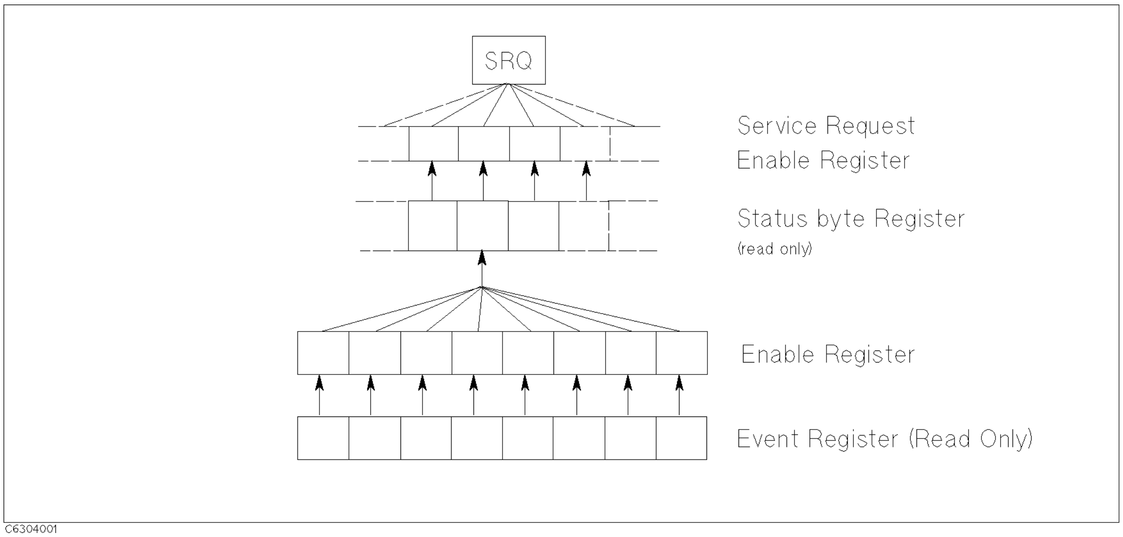


Figure 5-1. General Status Register Model

The status reporting system has a hierarchical structure as shown in Figure 5-1. When the analyzer condition satisfies the particular condition, the corresponding bit of the event register is set "1". Therefore, you can check the analyzer condition by reading the event register.

When the event register bit is set to "1", and corresponding enable register bit is also "1", the summary bit of the status byte register is set to "1". You can read the status byte register by using the serial poll.

If the corresponding bit of the service request enable register is "1", the service request (SRQ) is generated with the positive transition of the status byte register bit. By generating the SRQ, you can notify the controller that the analyzer is requesting service.

Event Register

Reflects the correspondent analyzer condition as a bit status. These bits monitor the changing analyzer state continuously and change the bit status as required.

You cannot change bit status by GPIB command.

The analyzer has the following event registers:

- Standard Event Status Register (ESB) Contains the summary of general processing which includes completion of sweep. See Table 5-2 for details.
- Event Status Register (ESR) Contains the information on occurrence of an operation failure such as a command error. See Table 5-3 for details.
- Operation Status Event Register (OSR) Contains only the information that a program is runnig and/or data is being transfered to the printer. See Table 5-4 for details.

Enable Register

The enable register selects which event register bits can set the bit in the summary bit of the status byte register that is connected to SRQ generation. The register bits work like mask bits. When you want to set a bit in the status byte register by a specific register condition, set the corresponding enable register to 1. This sets a 1 bit in the status byte register with a corresponding event register bit.

Status Byte Register

If enabled event register is set to 1, the corresponding bit of the status byte register is set to 1. This register also indicates the output queue and SRQ status.

The value of the status byte register can be read by using the **SPOLL** statement or the ***STB?** query from the controller. Reading the status byte register by either command does not affect the contents of the status byte register. Table 5-1 lists the contents of the status byte register.

A serial poll initiated by using the **SPOLL** command reads bit 6 of the status byte register as the RQS bit. The ***STB?** command reads bit 6 as the MSS bit.

SRQ (Service Request) can be generated by the status byte register by setting the service request enable register. For more information about SRQ, see Figure 5-6 in this chapter.

Transition Filter and Condition Register

The transition filter allows you to select which transitions of the analyzer condition will set a bit in the event register.

When the status register has a transition filter, there is a lower register called a condition register under the event register. The transition filter is between the event register and the condition register. The transition filter enables you to select a positive and/or negative transition of the condition register bit to set a bit in the corresponding event register. For example, if you set the negative transition filter, a 1 is set in the event register by changing from 1 to 0 in the event register.

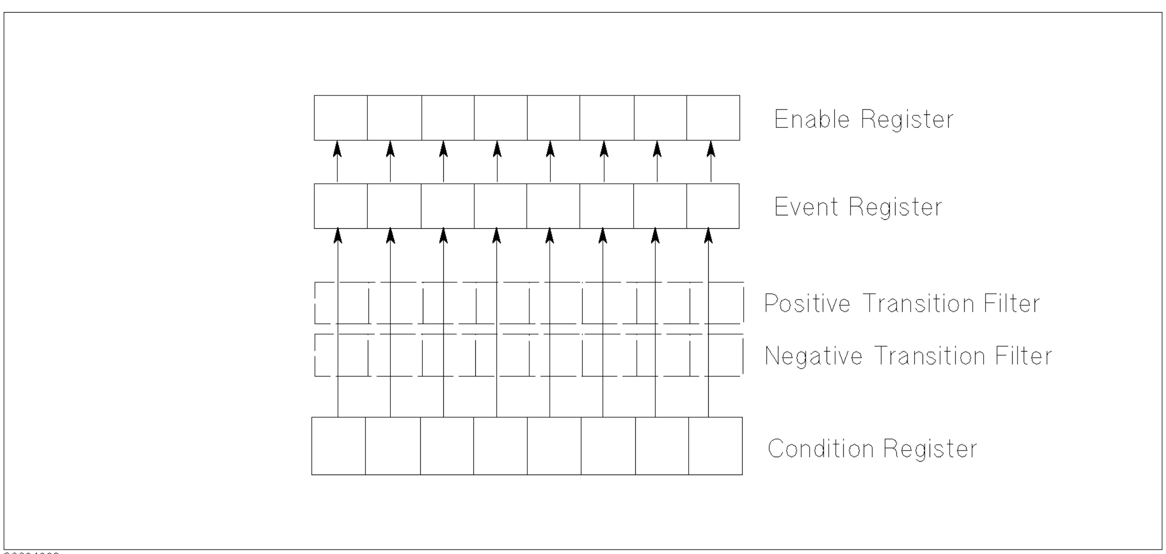


Figure 5-2. Transition Filter and Condition Register

For the 4395A, only the “Program Running” bit of the operation status register has a transition filter. By using the transition filter, you can generate an SRQ either at the start or the end of the program execution.

Status Register Structure

Figure 5-3 shows the status reporting structure of the 4395A.

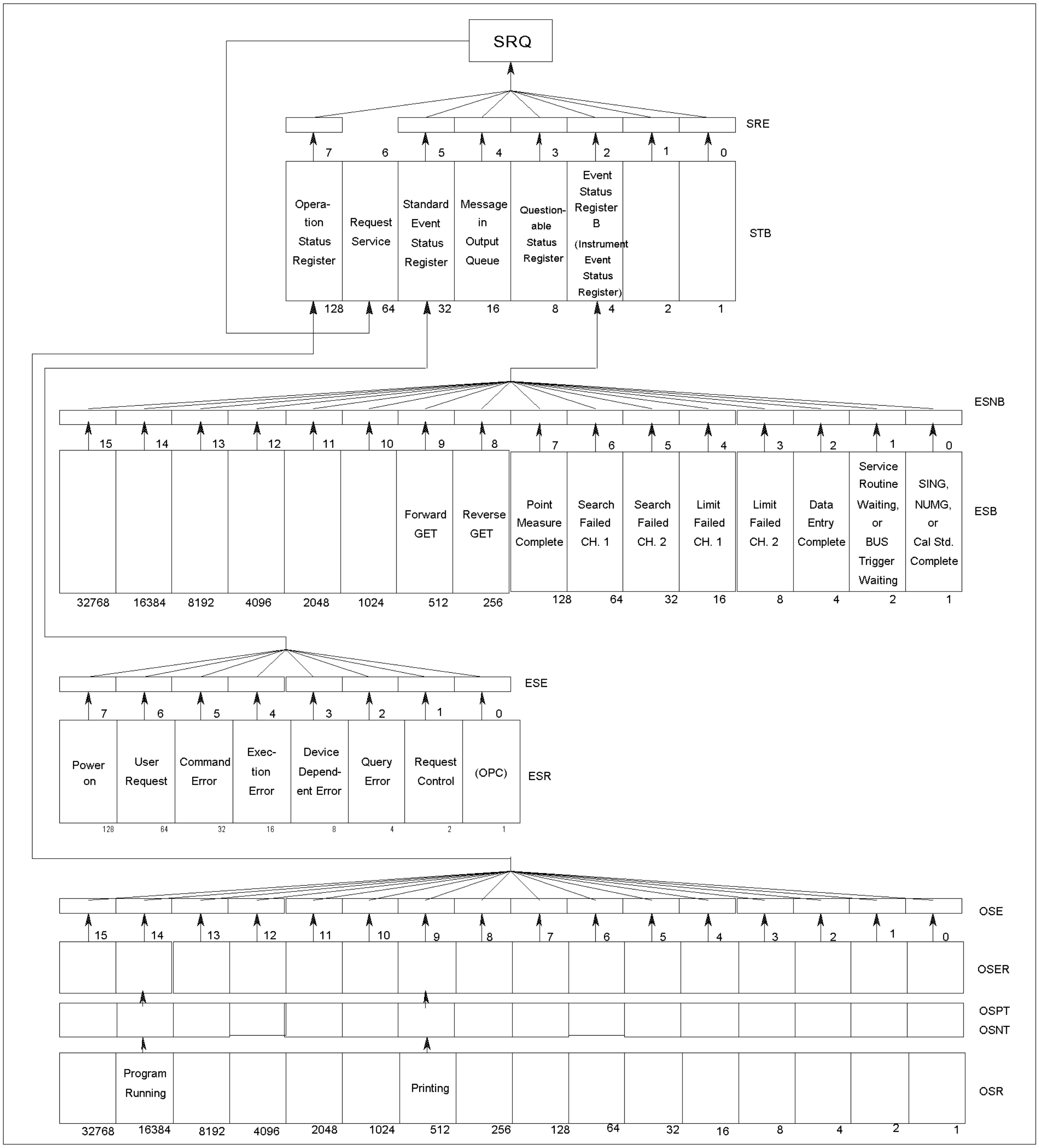


Figure 5-3. Status Reporting Structure

Status Bit Definitions of the Status Byte (STB)

The status bytes consist of 8-bit registers, with each bit representing a specific analyzer condition. The value of the Status Byte can be read by using SPOLL(717) statement from an external controller. This command reads a value directly from the analyzer without being set to remote. So, you can operate front panel keys while a controller is reading the Status Byte. Contents of the Status Byte can also be read by using the *STB? command. Reading the Status Byte does not affect the contents of the Status Byte. Table 5-1 shows contents of Status Byte.

Table 5-1. Status Bit Definitions of the Status Byte (STB)

Bit	Name	Description
2	Event Status Register B Summary Bit	One of the enabled bits in Event Status Register B (Instrument Event Status Register) has been set.
3	Questionable Status Register Summary Bit	The analyzer has no operation to report the event to the Questionable Status Register group. This register is available to keep the consistency with other SCPI compatible instruments.
4	Message in Output Queue	A command has prepared information to be output, but it has not been read yet.
5	Standard Event Status Register Summary Bit	One of the enabled bits in the Standard Event Status Register has been set.
6	Request Service	One of the enabled Status Byte bits is causing an SRQ.
7	Operation Status Register Summary Bit	One of the enabled bits in the Operation Status Register has been set.

For example, to read the contents of Message in the output queue,

```
10  Stat=SPOLL(717)
20  Stb4=BIT(Stat,4)
30  PRINT Stb4
40  END
```

Figure 5-4. Example of Reading Status Byte (1)

or,

```
10  ASSIGN Hp4395 TO 717
20  OUTPUT Hp4395;"*STB?"
30  ENTER Hp4395;Stat
40  Stb4=BIT(Stat,4)
50  PRINT Stb4
60  END
```

Figure 5-5. Example of Reading Status Byte (2)

Status Bit Definitions of ESB, ESR, and OSR

The Standard Event Status Register (ESR), Event Status Register B (ESB; Instrument Event Status Register), and Operation Status Register (OSR) are subordinate to the Status Byte. Each register can set a bit with a condition that is watched by status bit. A status bit is cleared when it is read by query or the `CLES` or `*CLS` command is executed.

Table 5-2. Status Bit Definitions of the Standard Event Status Register (ESR)

Bit	Name	Description
0	Operation Complete	A command for which OPC has been enabled, and completed an operation.
1	Request Control	The analyzer has been commanded to perform an operation that requires control of a peripheral, and needs control of GPIB.
2	Query Error	1. The analyzer has been addressed to talk, but there is nothing in the output queue to transmit. 2. Data in the Output Queue has been lost.
3	Device Dependent Error	An error, other than a command error, a query error, and an execution error has occurred.
4	Execution Error	1. A program data element following a header exceeded its input range, or is inconsistent with the analyzer's capabilities. 2. A valid program message could not be properly executed due to some analyzer condition.
5	Command Error	1. An IEEE 488.2 syntax error has occurred. Possible violations include, a data element violated the analyzer listening formats or a data element type is unacceptable to the analyzer. 2. A semantic error that indicates an unrecognized header was received has occurred. Unrecognized headers include incorrect device-specific headers and incorrect or unimplemented IEEE 488.2 common commands. 3. A Group Execute Trigger (`GET`) was entered into the Input Buffer of a program message.
6	User Request	The operator pressed a front panel key or an optional keyboard key or turned the rotary knob.
7	Power ON	This bit is set when a power-on sequence occurs.

Table 5-3. Status Bit Definitions of the Event Status Register B (ESB)

Bit	Name	Description
0	SING, NUMG, or Cal Std. Complete	A single, group sweep, calibration, or compensation has been completed since the last read of the register. Operates in conjunction with SING or NUMG.
1	Service Routine Waiting or Bus Trigger Waiting	1. An internal service routine has completed an operation, or is waiting for an operator response. 2. The analyzer has set the manual trigger to the point mode and is waiting for a manual trigger.
2	Data Entry Complete	A terminator key has been pressed.
3	Limit Failed, Ch 2	Limit test failed on channel 2.
4	Limit Failed, Ch 1	Limit test failed on channel 1.
5	Search Failed, Ch 2	A marker search was executed on channel 2, but the target value was not found.
6	Search Failed, Ch 1	A marker search was executed on channel 1, but the target value was not found.
7	Point Measurement Complete[1]	One measurement point of a sweep has been completed.
8	Reverse GET	A one-path 2-port calibration is active, and the analyzer has stopped, waiting for the operator to connect the device for a reverse measurement.
9	Forward GET	A one-path 2-port calibration is active, and the analyzer has stopped, waiting for the operator to connect the device for a forward measurement.

1 This bit is set only when the related bits of both SRE and ESNB are enabled.

In the case of the manual trigger on point mode, the analyzer accepts the next trigger while the current measurement is in progress (up to the number of points). Use bit 1 and bit 7 correctly to synchronize the measurement and external triggering. For example, 1) wait until bit 1 is set, 2) trigger, and 3) wait until bit 7 is set.

Table 5-4. Status Bit Definitions of the Operation Status Register (OSR)

Bit	Name	Description
9	Printing	Data is being transfered to the printer.
14	Program running	An HP Instrument BASIC program is running.

Each status register has a register that enables generating a Service Request (SRQ) with a condition of a status bit. For instance, to generate an SRQ when the analyzer completes the specified number of sweeps, enable ESNB bit 1. Bit 1 of ESNB is the mask register for ESB 0 ("SING, NUMG, or Cal Std. Complete") which shows sweep completion and SRE bit 2. This enables a path from ESB bit 0 to generate an SRQ. Figure 5-6 shows a program listing that can be used to generate an SRQ.

```
10  ASSIGN Hp4395 TO 717
20  !
30  OUTPUT Hp4395;"CLES"   ! Clears status registers
40  OUTPUT Hp4395;"ESNB 1" ! Enables mask register of "SING. NUMG. or
50  !                      ! Cal Std. Complete" of ESB
60  OUTPUT Hp4395;"*SRE 4" ! Enables mask register of "Event Status
70  !                      ! Register B" of STB
80  !
90  ON INTR 7 GOTO End     ! Declare SRQ interrupt
100 ENABLE INTR 7;2
110 OUTPUT Hp4395;"SING"   ! Execute single sweep
120 GOTO 120               ! Endless loop
130 !
140 End:                   ! Exit from loop when sweep is completed
150 END
```

Figure 5-6. Example of Generating a Service Request (SRQ)

OSPT, OSNT

OSPT (Operation Status Positive Transition Filter)

Sets the positive transition filter. Setting a bit in OSPT will cause a 0 to 1 transition in the corresponding bit of the associated Operation Status Register (OSR) to cause a 1 to be written in the associated bit of corresponding Operation Status Event Register (OSER).

Bit 14 of the analyzer's OSR is used to show program status. When bit 14 of OSPT is set to 1, starting a program causes a 1 to be written in bit 14 of OSER. (This sets bit 7 of STB to 1.)

OSNT (Operation Status Negative Transition Filter)

Sets the negative transition filter. Setting a bit in the negative transition filter will cause a 1 to 0 transition in the corresponding bit of the associated Operation Status Register to cause a 1 to be written in the associated bit of corresponding Operation Status Event Register.

Bit 14 of the analyzer's OSR is used to show program status. When bit 14 of OSNT is set to 1, stopping a program causes a 1 to be written in bit 14 of OSER. (This sets bit 7 of STB to 1.)

How to Use the Status Registers in a Program

You can use the status registers to determine the specific analyzer status in the program. To determine the contents of the status register, the following methods are used:

- Read an event register directly.
- Use the Service Request (SRQ).

Reading an Event Register Directly

You can read the contents of the event register directly to determine the specific analyzer condition. Use this method if you do not need to know the timing of the event register changes. The following procedure reads the register directly:

1. Query the event register or the condition register contents.
2. Retrieve a return value.

The list shown below is an HP-BASIC sample program using the BIT function.

```
OUTPUT @Hp4395"ESB?"                  Queries instrument event status register contents.
ENTER @Hp4395Esb                      Retrieve return value.
IF BIT(Esb,4) THEN                    If bit 4 of the instrument event status register is set to 1, the limit test
  DISP "LIMIT TEST FAILED AT Ch 1."   failed on channel 1.
END IF
```

Module 4-. Reading an Event Register

- Related GPIB Commands. The following query commands can be used to read the contents of an event register directly.

*STB?	Returns Status Byte Register contents.
*ESR?	Returns Event Status Register contents.
ESB?	Returns Instrument Event Status Register contents.
OSR?	Returns Operation Status Register contents.

To Report Command Error Occurrence

This section provides an example which describes how to report command error occurrence using SRQ.

```
10   !
20   !Fig.5-7  To Report Command Error Occurrence
30   !
40   ASSIGN @Hp4395 TO 717  ! When iBASIC is used, change "717" to "800"
50   !
60   DIM Err$[30]
70   OUTPUT @Hp4395;"CLES"
80   OUTPUT @Hp4395;"*SRE 32 ;*ESE 32"
90   ON INTR 7 GOSUB Err_report   ! \ When iBASIC is used,
100  ENABLE INTR 7;2              ! / change "7" to "8"
110     !
120     !  OUTPUT statement to send GPIB command
130     !
140     !
150     !
160   GOTO Prog_end
170 Err_report:    !
180     OUTPUT @Hp4395;"OUTPERRO?"
190     ENTER @Hp4395;Err,Err$
200     PRINT "COMMAND ERROR DETECTED"
210     PRINT Err,Err$
220     !
230     A=SPOLL(@Hp4395)
240     OUTPUT @Hp4395;"*ESR?"
250     ENTER @Hp4395;Estat
260     ENABLE INTR 7    ! When iBASIC is used, change "7" to "8"
270   RETURN
280 Prog_end:   !
290   END
```

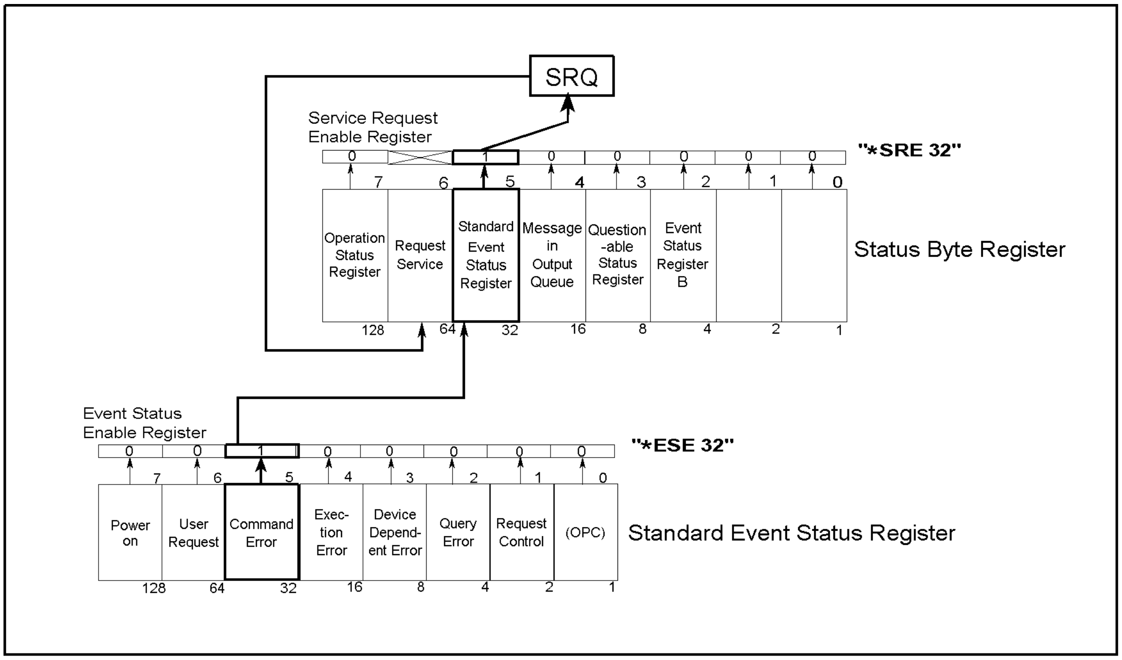
Figure 5-7. Sample Program : To Report Command Error Occurrence

Enable Error Bit

```
70   OUTPUT @Hp4395;"CLES"
80   OUTPUT @Hp4395;"*SRE 32 ;*ESE 32"
```

Line 70 clears all bits of the Status Registers and Enable Registers.

In line 80, the command `*SRE 32` sets the Service Request Enable Register to 00100000 (this enables bit 5 of the Status Byte Register). The command `*ESE 32` sets the Standard Event Status Enable Register to 00100000 (this enables bit 5 of the Standard Event Status Register (see Figure 5-8).



CC100501

Figure 5-8. Command-Error Bit Enabling

Report Command Error

```
90   ON INTR 7 GOSUB Err_report   ! \ When iBASIC is used,
100  ENABLE INTR 7;2              ! / change "7" to "8"
110     !
120     !  OUTPUT statement to send GPIB command
130     !
140     !
150     !
160   GOTO Prog_end
170 Err_report:    !
```

If an GPIB command (executed between lines 100 and 160) causes an GPIB command error, the analyzer generates an SRQ and the controller branches to `Err_report`. For example, the `OUTPUT` statement:

`120 OUTPUT @Hp4395;"CENT "` *Setting center frequency, but no parameter*

Output Error

```
180     OUTPUT @Hp4395;"OUTPERRO?"
190     ENTER @Hp4395;Err,Err$
200     PRINT "COMMAND ERROR DETECTED"
210     PRINT Err,Err$
```

These commands retrieve the error number and description.

In the error shown in the line 120 example, the controller displays the following:

```
COMMAND ERROR DETECTED
-109   "Missing parameter"
```

Many different kinds are defined for GPIB errors. Refer to the error code table when an error occurs.

Return to Execute GPIB command

```
230      A=SPOLL(@Hp4395)
240      OUTPUT @Hp4395;"*ESR?"
250      ENTER @Hp4395;Estat
260      ENABLE INTR 7    ! When iBASIC is used, change "7" to "8"
270    RETURN
```

Lines 230 to 270 clear SRQ before returning to the main routine.

Line 230 reads the analyzer's status byte. The `A=SPOLL(@Hp4395)` statement reads the Status Byte Register of the address `@Hp4395`(analyzer), and enters the value into `A`. The command error causes the SRQ and with bit 5 and bit 6 of the Status Byte Register set, the value of `A` is 96. Reading the Status Byte Register by using the `SPOLL` command clears SRQ (status byte bit 6).

In line 240 and line 250, the command `*ESR?` reads the contents of the Standard Event Status Register. With Bit 5 of Standard Event Status Register set, the value of `Estat` is 32. Reading the Standard Event Status Register by using the `*ESR?` command clears the register.

A branch to `Err_report` disables the interrupt. Therefore, the return from `Err_report` must reenable the interrupt.

To Wait for Sweep End

```
10 !
20 !Fig.5-9  To Wait for Sweep End
30 !
40  ASSIGN @Hp4395 TO 717  ! When iBASIC is used, change "717" to "800"
50  OUTPUT @Hp4395;"TRGS INT"
60  OUTPUT @Hp4395;"CLES"
70  OUTPUT @Hp4395;"*SRE 4;ESNB 1"
80  ON INTR 7 GOTO Sweep_end  ! \ When iBASIC is used, change "7" to "8"
90  ENABLE INTR 7;2           ! /
100   OUTPUT @Hp4395;"SING"
110 Measuring:GOTO Measuring
120 Sweep_end:  !
130  DISP "MEASUREMENT COMPLETE"
140  END
```

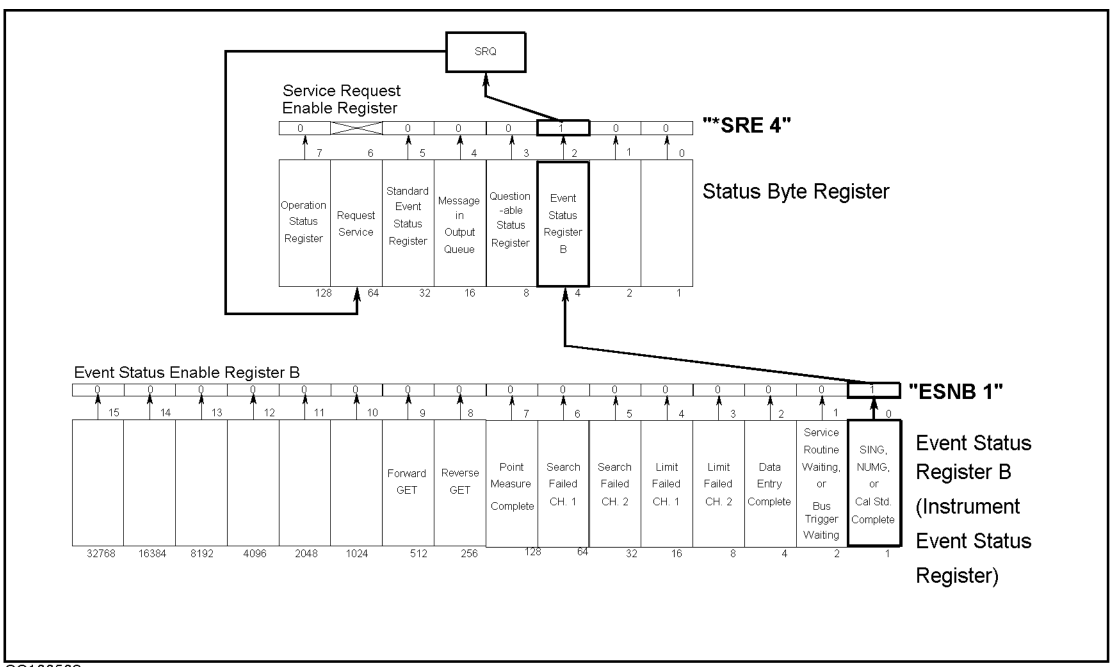
Figure 5-9. Sample Program : To Wait for Sweep End

Enable Sweep-End Bit

```
60   OUTPUT @Hp4395;"CLES"
70   OUTPUT @Hp4395;"*SRE 4;ESNB 1"
```

Line 60 clears all bits of the Status Registers and the Enable Registers.

In line 70, the command `*SRE 4` sets the Service Request Enable Register to 00000100 (this enables bit 2 of the Status Byte Register). The command `ESNB 1` sets the Event Status Enable Register B to 0000000000000001 (this enables bit 0 of the Event Status Register B. See Figure 5-10).



CC100502

Figure 5-10. Sweep-End Bit Enabling

Enable Registers select which events in the analyzer can cause a service request (SRQ). By setting bit 0 of the Event Status Enable Register B to 1, the occurrence of the corresponding event (sweep-end) sets bit 0 of the Event Status Register B. When this bit is set (and is enabled), itis used to set a summary bit in the Status Byte Register (bit 2). Also, because bit 2 of Service Request Enable Register is set, setting the corresponding bit (Event Status Register B summary bit) generates an SRQ. The SRQ sets bit 6 of the Status Byte Register.

Enable SRQ Interrupt

```
80   ON INTR 7 GOTO Sweep_end  ! \ When iBASIC is used, change "7" to "8"
90   ENABLE INTR 7;2           ! /
⋮
120 Sweep_end:  !
```

Line 80 defines a branch. When the SRQ interrupt is generated from the GPIB interface (whose select code is 7), the controller goes to `Sweep_end` (Line 120).

Line 90 enables an interrupt from interface 7 (GPIB) when bit 1 (SRQ bit) of the interrupt register (of the controller) is set by a value of 2.

Wait Until Measurement Is Done

```
100    OUTPUT @Hp4395;"SING"
110 Measuring:     GOTO Measuring
```

The controller loops back in line 110 until an SRQ interrupt occurs.

Generate SRQ

On a single sweep end, bit 0 of the ESB is set (which sets bit 2 of the Status Byte Register) and an SRQ is generated.

```
120 Measuring:     GOTO Measuring    Loop until SRQ interrupt
130 Sweep_end:  !                    At SRQ interrupt, jump to here
```

Once an SRQ is generated, the SRQ interrupt is disabled.

6

Using the Trigger System in 4395A

This chapter describes how to control the trigger system of the analyzer.

To trigger a measurement from a controller, the following steps are commonly used:

1. Set the trigger source to:

   Bus, or Internal (free run)

   (In External, Video, Manual or Gate trigger, you cannot trigger from the controller, so these sources are not mentioned in this guide.)

2. Set the number of measurements and the analyzer is initiated. You can set the number of measurements as:

   (Hold)—Single—Number of Group—Continuous

3. Generate the trigger event and the analyzer starts a measurement.

The analyzer trigger system has three states: Idle, Waiting for Trigger, and Measurement.

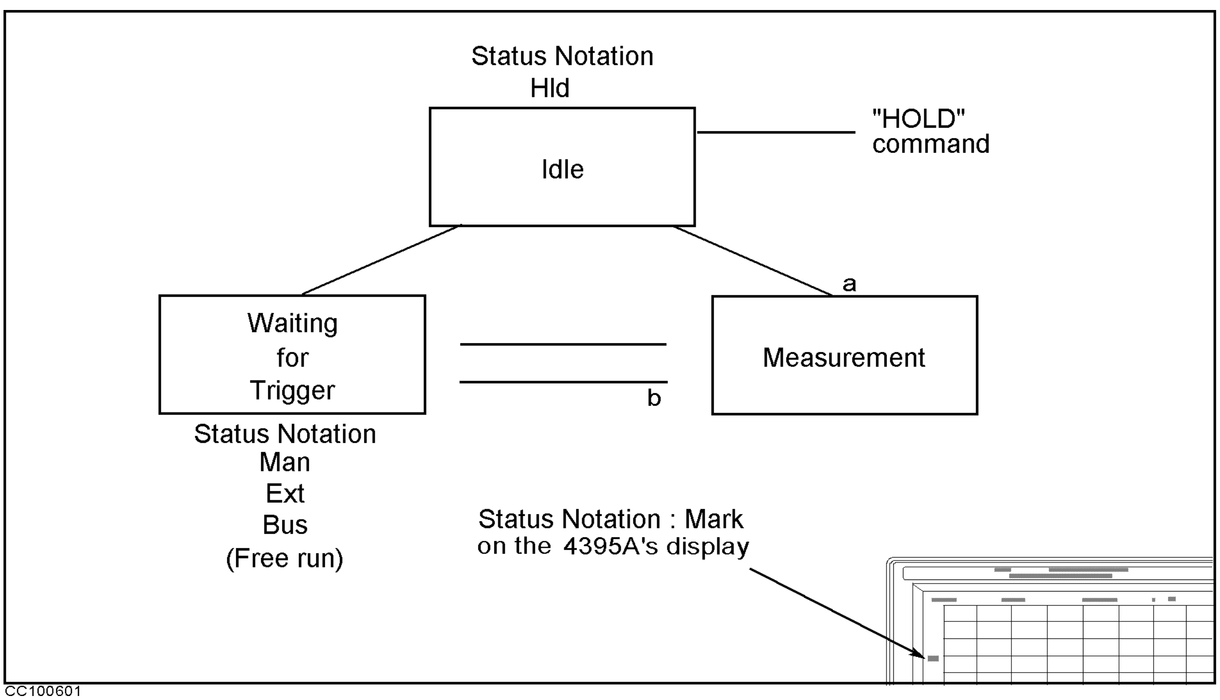


Figure 6-1. Trigger System

In Figure 6-1,

1. After a HOLD GPIB command execution, the analyzer returns to the “Idle” state.
2. By setting the number of measurements, the analyzer changes from the “Idle” state to the “Waiting for Trigger” state.
3. At the “Waiting for Trigger” state, a trigger input (corresponding to the trigger source) starts a measurement.

Bus	GPIB command *TRG or BASIC command TRIGGER triggers measurements.
Internal (free run)	There is no need for a trigger input. The analyzer starts the measurements immediately.

4. After the measurement is complete, the next state depends on the number of measurements.

Single	goes to the “Idle” state(4-a).
Number of Groups	Goes to the “Waiting for Trigger” state until the number of groups not measured yet equals zero(4-b). After all measurements are completed, goes to “Idle” state(4-a).
Continuous	goes to the “Waiting for Trigger” state(4-b).

To Measure Continuously

```
10 !
20 !Fig.6-2  To Trigger Measurement Continuously
30 !
40 ASSIGN @Hp4395 TO 717  ! When iBASIC is used, change "717" to "800"
50 OUTPUT @Hp4395;"TRGS INT"
60 OUTPUT @Hp4395;"CONT"
70 END
```

Figure 6-2. Sample Program : To Trigger Measurements Continuously

Set Trigger Source

```
50 OUTPUT @Hp4395;"TRGS INT"
```

Set the trigger source to internal.

Start Continuous Measurement Sweep

```
60 OUTPUT @Hp4395;"CONT"
```

The analyzer changes to the “Waiting for Trigger” state. In this program, the internal trigger source is selected and the analyzer immediately starts continuous measurements.

What can you do to abort a measurement?

Send the command:

```
OUTPUT @Hp4395;"HOLD"
```

Note

The 4395A will fail in proper measurement data transfer when it is triggerd using the internal trigger source. If you want to transfer measurement data to the controller, you must use either the `SING` or the `NUMG` *parameter* command to synchronize the controller and the analyzer. To use these commands, see "Using the EXECUTE Statement to Synchronize with the Completion of Sweep" in Chapter 4 and "To Wait for Sweep End" in Chapter 5.

To Trigger a Measurement From the Controller

Two methods of triggering a measurement from the controller are shown in Figure 6-3 and Figure 6-4.

```
10 !
20 !Fig.6-3  To Trigger Measurement From Controller(1)
30 !
40 ASSIGN @Hp4395 TO 717 ! When iBASIC is used, change "717" to "800"
50 OUTPUT @Hp4395;"TRGS INT"
60 OUTPUT @Hp4395;"SING"
70 END
10 !
```

Figure 6-3. Sample Program : To Trigger a Measurement from Controller (1)

Set Trigger Source

```
50 OUTPUT @Hp4395;"TRGS INT"
```

Set the trigger source to internal.

Trigger a Measurement

```
60 OUTPUT @Hp4395;"SING"
```

The analyzer changes to the "Waiting for Trigger" state. In this program, the internal source is selected and the analyzer immediately starts a measurement. After the measurement, the analyzer goes to the "Idle" state.

How can you perform averaging?

When you set the averaging on, you must also set the number of measurements to the same value as the averaging factor. For example, if the averaging factor is 10, replace line 60 as follows:

```
60 OUTPUT @Hp4395;"NUMG 10"
```

How can you wait for a measurement to be completed?

When you want to return the measurement data to the controller, you must wait for the measurement to be completed. For details, see Chapter 4.

```
10 !
20 !Fig.6-4  To Trigger Measurement From Controller(2)
30 !
40 ASSIGN @Hp4395 TO 717 ! When iBASIC is used, change "717" to "800"
50 OUTPUT @Hp4395;"TRGS BUS"
60 OUTPUT @Hp4395;"CONT"
70 OUTPUT @Hp4395;"*TRG"
80 END
```

Figure 6-4. Sample Program : To Trigger a Measurement from Controller (2)

Set Trigger Source

```
50 OUTPUT @Hp4395;"TRGS BUS"
```

Set the trigger source to bus.

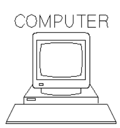
Trigger a Measurement

```
70 OUTPUT @Hp4395;"*TRG"
```

Triggers the analyzer. When the trigger source is set to bus, you can use the group execution trigger as follows:

```
70 TRIGGER 7
```

What is Group Execution Trigger (GET)?



The HP BASIC command `TRIGGER` can be used instead of the `*TRG` command. The HP BASIC command is used to trigger all triggerable instruments on a BUS at the same time. Therefore, to trigger all triggerable instruments on select code 7(GPIB bus) execute the command:

```
TRIGGER 7
```

Setting the Gate Trigger

The gate trigger can be set via the controller, though it cannot be controlled directly. You should define Gate Length and Gate Delay as below.

- Gate Delay = $\tau/2 + \mathrm{SD}$
- Gate Length = $\tau/4$

Where τ means Signal Length and SD means Signal. And you should measure these parameters with an oscilloscope. See *Operation Manual* for details in the gate trigger.

```
10 !
20 !Fig.6-5  Gate Trigger configure
30 !
40 ASSIGN @Hp4395 TO 717 ! When iBASIC is used, change "717" to "800"
50 OUTPUT @Hp4395;"TRGS GAT"
60 OUTPUT @Hp4395;"GATCTL LEV"
70 OUTPUT @Hp4395;"GATDLY 10E-6"
80 OUTPUT @Hp4395;"GATLEN 100E-6"
90 END
```

Figure 6-5. Sample Program : Setting the Gate Trigger

Setting the Gate Delay

```
70 OUTPUT @Hp4395;"GATDLY 10E-6"
```

Set the gate delay to 10 μsec.

Setting the Gate Length

```
80 OUTPUT @Hp4395;"GATLEN 100E-6"
```

Set the gate length to 100μsec.

7

Using the I/O Port

This chapter describes how to use the I/O port of the analyzer with the GPIB. For general operation of the I/O port, see the *Operation Manual*.

The I/O port on the analyzer's rear panel communicates with external devices (for example, a handler on a production line). 4395Ahas 8 bit I/O port and 24 bit I/O port.

Overview

This section gives you an overview of the usage of the I/O port, taking the 8-bit I/O port as an example.

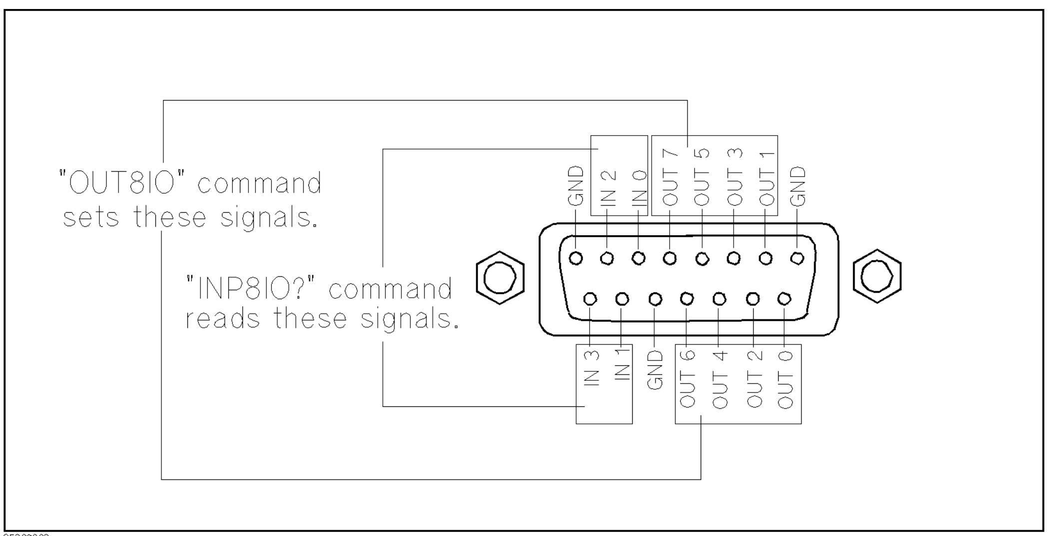


Figure 7-1. 8-bit I/O Port

The I/O port consists of the following 15 TTL compatible signals:

8-bit output
4-bit input
3 grounds

The signals IN 0 to IN 3 and OUT 0 to OUT 7 can be read and set by GPIB commands.

To Synchronize External Handler with Analyzer

```
10 !
20 !Fig.7-2  Synchronization of an External Handler
30 !            with the Analyzer
40 !
50  ASSIGN @Hp4395 TO 717 ! When iBASIC is used, change "717" to "800"
60  !
70  !
80  OUTPUT @Hp4395;"OUT8IO 8"
90  !
100 REPEAT                       !
110   OUTPUT @Hp4395;"INP8IO?"    !
120   ENTER @Hp4395;Inpio         ! Waiting Handler Response
130   A=BIT(Inpio,3)              !
140 UNTIL A=1                    !
150 !
160 !
170 END
```

Figure 7-2. Sample Program : Synchronization of an External Handler with the Analyzer

Send Signal to the External Handler

```
80 OUTPUT @Hp4395;"OUT8IO 8"
```

The `OUT8IO` *parameter* command sets the 8-bit data value of the OUT 0-7 lines. The `OUT8IO 8` command sets the OUT 3 line to TRUE (1).

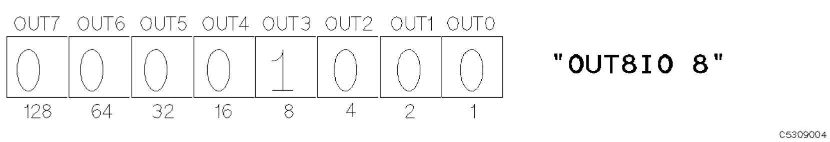


Figure 7-3. 8-Bit Data of OUT0-7

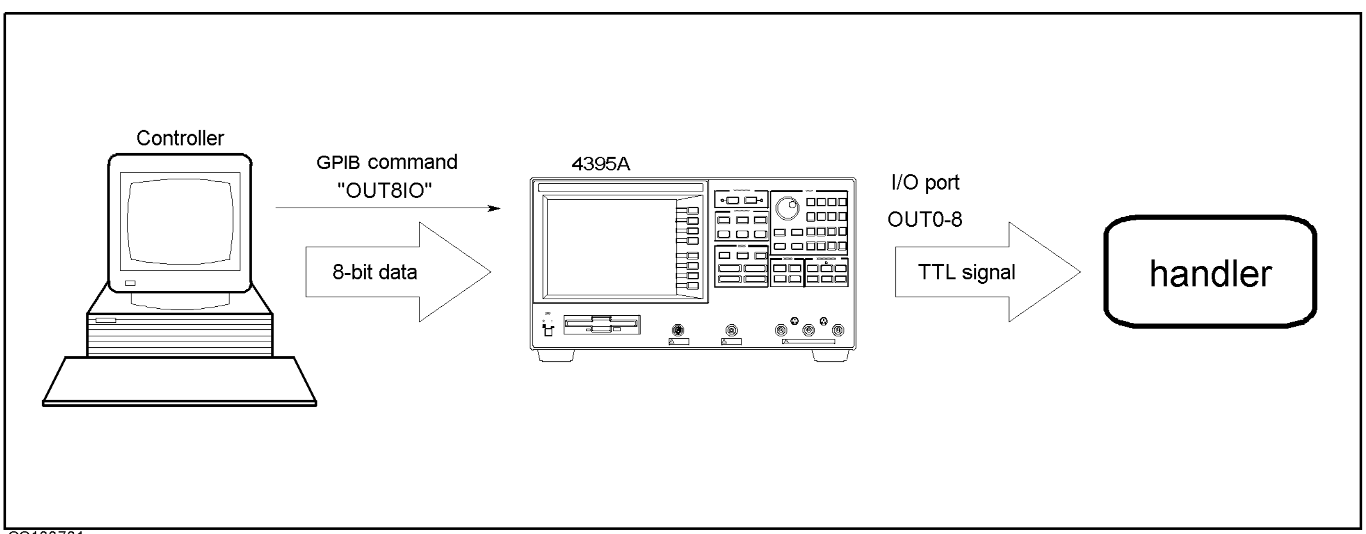


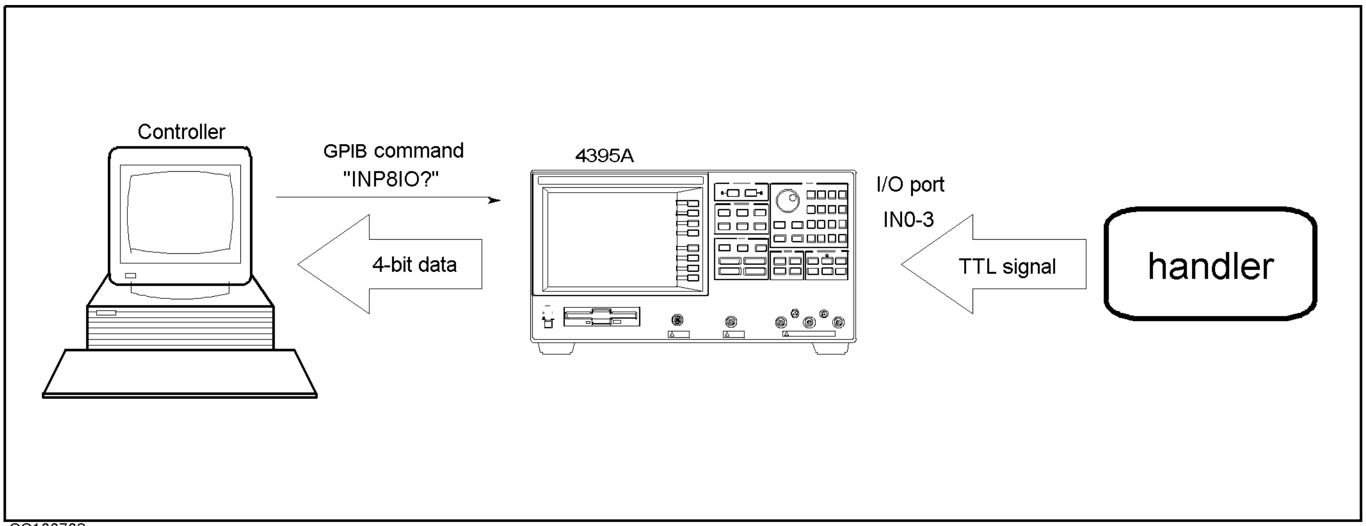
Figure 7-4. Sending Signal to an the External Handler

Read Signal from the External Handler

```
100 REPEAT
110   OUTPUT @Hp4395;"INP8IO?"
120   ENTER @Hp4395;Inpio
130   A=BIT(Inpio,3)
140 UNTIL A=1
```

The `INP8IO?` command returns the 4-bit data value of the IN 0-3 lines.

Lines 100 to 140 wait for the external handler to set signal on line IN 3 to TRUE (1).



CC100702

Figure 7-5. Reading Signal from the External Handler

8-bit I/O port

The 4395A's 8-bit I/O port consists of 15 TTL level signals, which fall into 8-bit output signals, 4-bit input signals, and grounds.

I/O Pins

Figure 7-1 shows I/O pins.

The signals assigned to each pin are described below:

OUT 0 through 7	Signal lines used to output signals to an external device. They are controlled by the command, `WRITEIO` or `OUT8IO` as described below. Once one of these commands is executed, the signal is latched until one of them is executed again.
IN 0 through 3	Signal lines used to read an input signal from an external device. They are controlled by the command `READIO` or `INP8IO` as described below.

IBASIC Commands for the 8-bit I/O Port Control

IBASIC commands related to the 8-bit I/O port are defined as follows:

- `WRITEIO 15,0;` Outputs 8-bit data through lines OUT 0 to OUT 7. The OUT 0 signal is the LSB (least significant bit) and the OUT 7 signal is the MSB (most significant bit).
- `READIO(15,0)` Inputs 4-bit data through lines from IN 0 to IN 3 to the 4395A's memory and returns the data to an IBASIC program. The IN 0 signal is the LSB and the IN 3 signal is the MSB.

GPIB Commands for the 8-bit I/O Port Control

The GPIB commands related to the parallel I/O ports are described below:

- `OUT8IO` Outputs 8-bit data through lines OUT 0 to OUT 7. The OUT 0 signal is the LSB (least significant bit), and the OUT 7 signal is the MSB (most significant bit).
- `INP8IO?` Inputs 4-bit data through lines from IN 0 to IN 3 to the 4395As memory and returns the data to a control device such as an external controller IBASIC program. The IN 0 signal is LSB (least significant bit) and the IN 3 signal is MSB (most significant bit).

The 24-bit I/O Port

The 4395A's 24-bit I/O port has four independent parallel ports for data input or output, and several control signal and power lines. All signals are TTL level. The data I/O port consists of 2 pairs of 8-bit output ports and 2 pairs of 4-bit two-ways ports. If you use these ports together, you can use them as a 24-bit output port or as an 8-bit input port at maximum. The I/O signal is initialized to use negative logic, but it can be set to use positive logic using an GPIB command. The control signal lines consist of measurement completion output, PASS/FAIL output of limit testing results, control signal outputs for handshaking, and so on.

Note A 36-pin cable (part number: 04278-61650) is available if you cannot connect the device directly to the connectors of the 24-bit I/O port interface on the rear panel. This cable enables a 1m cable extension of this interface.

I/O Port

The 4395A's 24-bit I/O port has following 2 pairs of output ports and 2 pairs of input/output ports.

- Output only ports
  - A port: 8-bit
  - B port: 8-bit

  The signal is TTL level and is a latched output.
- Two ways ports
  - C port: 4-bit
  - D port: 4-bit

  Both ports C and D are set as input ports when the 4395A is turned on. These ports can be used as output ports by using the GPIB commands `COUT` or `DOUT`. The signal is TTL level and is a latched output. (Related GPIB commands: `OUTCIO`, `OUTDIO`, `OUTPINPCIO?`, `OUTPINPDIO?`)

  Using GPIB commands, you can combine these ports for use as the following ports (in addition to the above 4 ports).
- The input/output port
  - E port: 8-bit (C port + D port)
- The output only ports
  - F port: 16-bit (A port + B port)
  - G port: 20-bit (A port + B port + C port)
  - H port: 24-bit (A port + B port + C port + D port)

Control Signal Lines

The I/O ports include 10 types of output signal lines and one input signal line. These control signals are TTL-compatible (excluding the power supply line). Each of them are described below.

Port C or Port D Status Output Signal

These signals are used to report the direction setting (input or output) of port C or D to external devices. Each of these signals is set to LOW respectively when port C or D is assigned as an input port. It is set to HIGH respectively when port C or D is defined as an output port. (Related GPIB commands: `CIN`, `COUNT`, `DIN`, and `DOUT`.)

WRITE STROBE Output Signal

When data is output to any output port, a negative pulse is output to the WRITE STROBE output. This negative output pulse notifies an external device of data output to the parallel I/O port. The pulse width is 10 μs (typical). Figure 7-6 shows the timing chart for the WRITE STROBE output and data output.

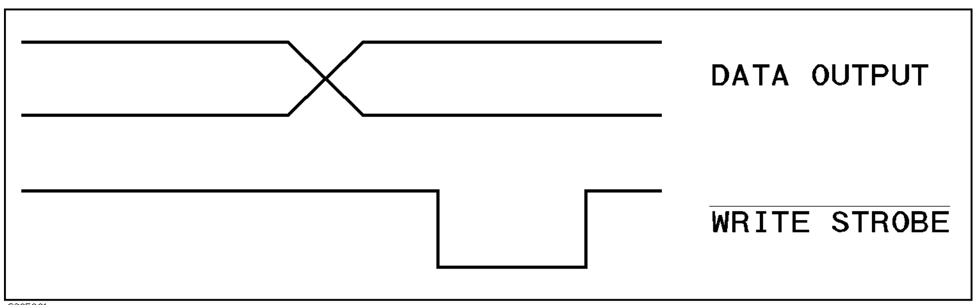


Figure 7-6. Write Strobe Signal Timing Chart

INPUT1 Input Signal

When a negative pulse is input to INPUT1, OUTPUT1 and OUTPUT2 are set to LOW or HIGH. The time interval is 200 ns (typical). An GPIB command is used to determine whether LOW or HIGH is to be set. The pulse width of the signal input to INPUT1 must be 1 μs or more. (Related GPIB commands: `OUT1ENVH`, `OUT1ENVL`, `OUT2ENVH`, `OUT2ENVL`, and `TRGOUT`.)

OUTPUT1 or OUTPUT2 Output Signal

This signal (a latch output signal) can be set to LOW or HIGH by inputting a negative pulse to INPUT1 or using an GPIB command. (Related GPIB commands: `OUT1H`, `OUT1L`, `OUT2H` and `OUT2L`.)

Note

You can change the logic level of an OUTPUT signal by synchronizing it with a measurement trigger, when you turn ON the trigger detection output function using the GPIB command `TRGOUT ON`. This function is used only in frequency transient measurements to send a load signal to a device immediately after a triggering measurement. The time interval between the measurement trigger and the logic level change is 85 μs (typical).

PASS/FAIL Output

Outputs a HIGH (positive logic) or LOW (negative logic) signal when the limit test passed, and a LOW (positive logic) or HIGH (negative logic) when the limit test failed. This is valid when the limit test function is set to ON.

WRITE STROBE Output for the PASS/FAIL Output

Outputs a negative pulse when a limit test result has been output through the PASS/FAIL output line. The output signal informs an external device of the limit test result being output through the PASS/FAIL output. The pulse width is 10 μs (typical).

SWEEP END Output

When the 4395A completes a measurement in the Tester mode or a sweep in the Analyzer mode, a negative pulse is output. When measurements are repeated with a continuous trigger, the pulse is output at every measurement or sweep end. The pulse width is 20 μs (typical).

+5V Output

A +5V output can be provided to an external device. The maximum current supplied is 100 mA. This line does not have a fuse. When excessive current flows, the 4395A's protection circuit automatically shuts down its internal power supply circuits. If you remove the cause of the excessive current, the 4395A's power will be turned on but the 4395A's setups are reset to the default settings.

Figure 7-7 shows the overview of I/O ports and control signal lines.

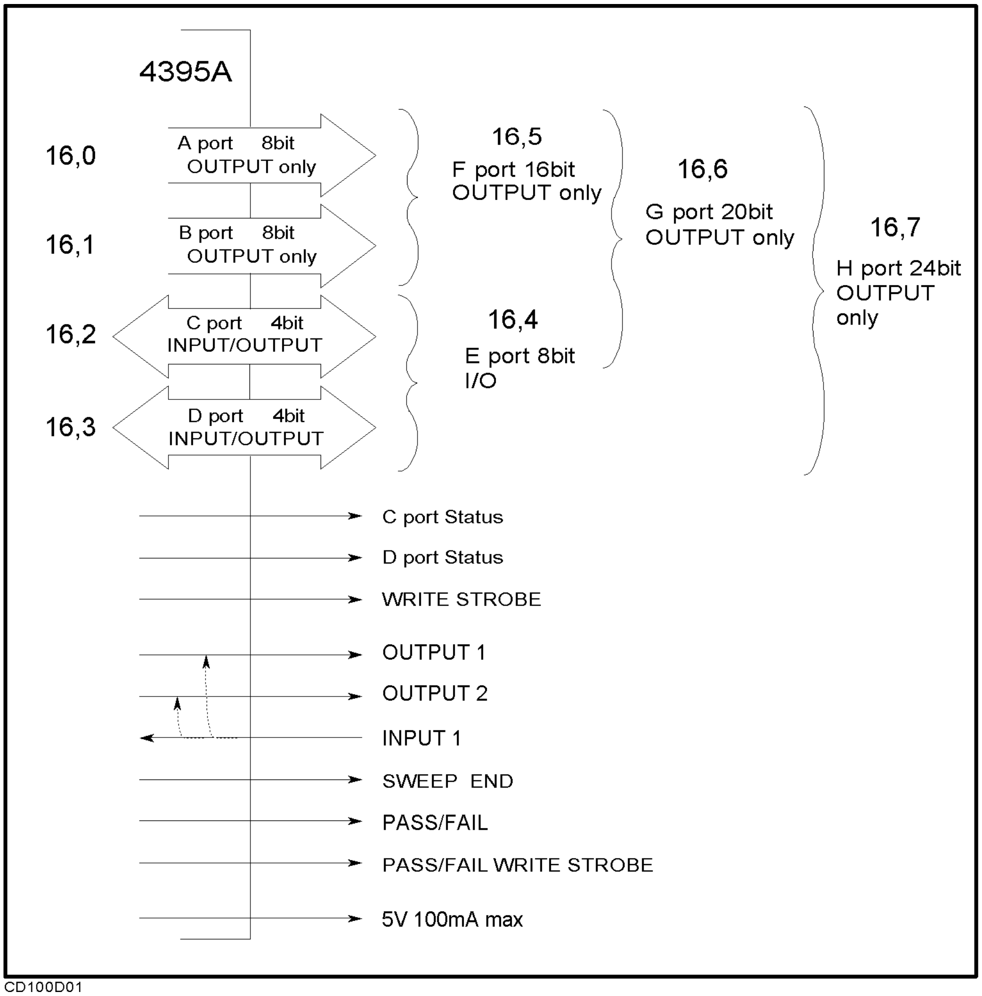


Figure 7-7. The Overview of 24-bit I/O Ports

Pin Assignment

Figure 7-8 shows the pin numbers. Table 7-1 shows the signal lines assigned to the pin numbers.

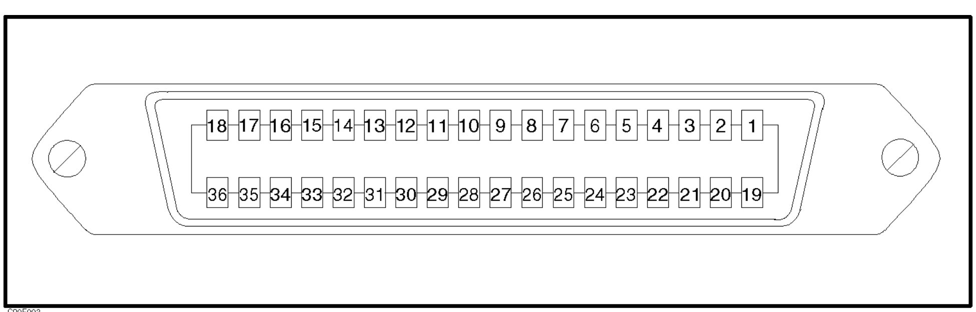


Figure 7-8. 24-bit I/O port Connector Pin Number

Table 7-1. Assignment of Signals to Pins (Standard)

Pin No.	Signal Name	Signal Standard
1	GND	0 V
2	INPUT1	TTL level, Pulse input (Width: $\geq$1 μs)
3	OUTPUT1	TTL level, Latch output
4	OUTPUT2	TTL level, Latch output
5	Output port A0	TTL level, Latch output
6	Output port A1	TTL level, Latch output
7	Output port A2	TTL level, Latch output
8	Output port A3	TTL level, Latch output
9	Output port A4	TTL level, Latch output
10	Output port A5	TTL level, Latch output
11	Output port A6	TTL level, Latch output
12	Output port A7	TTL level, Latch output
13	Output port B0	TTL level, Latch output
14	Output port B1	TTL level, Latch output
15	Output port B2	TTL level, Latch output
16	Output port B3	TTL level, Latch output
17	Output port B4	TTL level, Latch output
18	Output port B5	TTL level, Latch output
19	Output port B6	TTL level, Latch output
20	Output port B7	TTL level, Latch output
21	Input/output port C0	TTL level, Latch output
22	Input/output port C1	TTL level, Latch output
23	Input/output port C2	TTL level, Latch output
24	Input/output port C3	TTL level, Latch output
25	Input/output port D0	TTL level, Latch output
26	Input/output port D1	TTL level, Latch output
27	Input/output port D2	TTL level, Latch output
28	Input/output port D3	TTL level, Latch output
29	Port C status	TTL level, Input mode: Low, Output mode: High
30	Port D status	TTL level, Input mode: Low, Output mode: High
31	WRITE STROBE signal	TTL level, Negative logic, Pulse output (Width: $\geq$10 μs Typical)
32	+5 V pull-up	
33	SWEEP END signal	TTL level, Negative logic, Pulse output (Width: $\geq$20μs Typical)
34	+5 V	+5 V, 100 mA max.
35	PASS/FAIL signal	TTL level, PASS: HIGH, FAIL: LOW, Latch output
36	PASS/FAIL WRITE STROBE signal	TTL level, Negative Logic, Pulse output (Width: 10 μs; Typical)

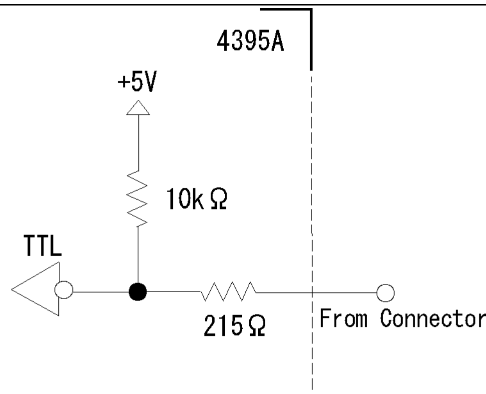
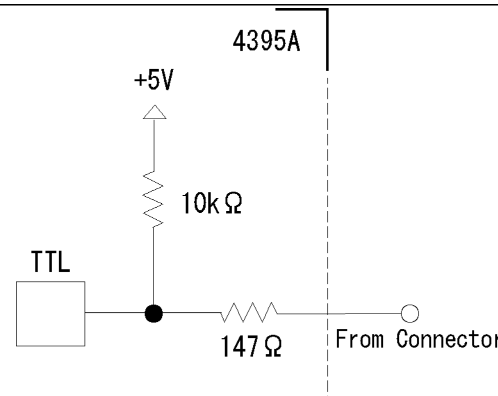
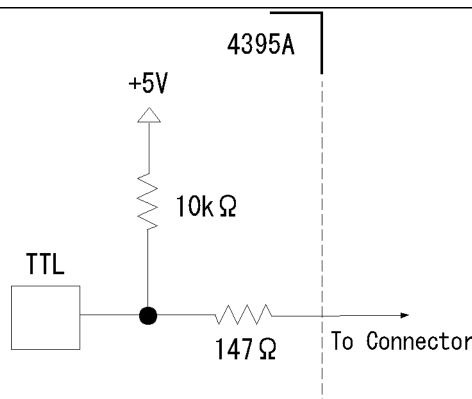
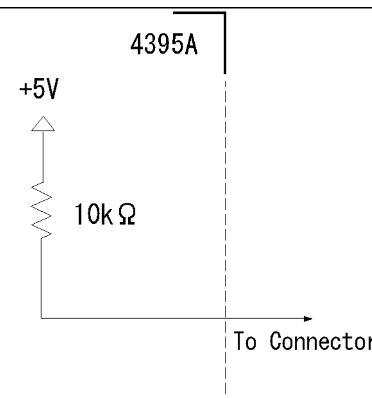
Power-ON Default

The 24-bit I/O port is set to the following defaults at power-on. (These settings do not change when (Preset) is pressed.)

Logic type	Negative logic
WRITE STROBE signal	HIGH
SWEEP END signal	HIGH
Port A	Negative 0 ⟶ HIGH
Port B	Negative 0 ⟶ HIGH
Port C	Input
Port D	Input
OUTPUT1	HIGH, pulled HIGH by the falling edge of INPUT1 (`OUT1ENVH`)
OUTPUT2	HIGH, pulled HIGH by the falling edge of INPUT1 (`OUT2ENVH`)
PASS/FAIL signal	(Negative) ⟶ HIGH

Basic I/O circuit

Table 7-2. 24-bit I/O Port, Basic I/O Circuit

Basic circuit	Input port	
		
I/O pin	INPUT1	Port C, D (input)[1]
Basic circuit	Output port	Others
		
I/O pins	OUTPUT1, 2 port A[1], B[1], C(output)[1], D(output)[1] Write strobe signal SWEEP END signal Port C status, Port D status	+5V pullup

1 Common to all bits

IBASIC Commands for 24-bit I/O Port Control

IBASIC commands related to 24-bit I/O port are described in the following paragraphs.

Data Output

The following commands output data to the corresponding ports (A to H). If you use C, D, E, F, G, or H port as the output port, you must use the GPIB command `COUT` and/or `DOUT` to set the C and/or D port as an output port.

- `WRITEIO 16,0;` Output 8-bit data to port A.
- `WRITEIO 16,1;` Output 8-bit data to port B.
- `WRITEIO 16,2;` Output 4-bit data to port C.

- WRITEIO 16,3; Output 4-bit data to port D.
- WRITEIO 16,4; Output 8-bit data to port E.
- WRITEIO 16,5; Output 16-bit data to port F.
- WRITEIO 16,6; Output 20-bit data to port G.
- WRITEIO 16,7; Output 24-bit data to port H.

Data Input

The following commands read data sent from an external device to an input port (C to E) and return the value to an HP IBASIC program. If you use the port as an input port, the port must be defined as an input port using the GPIB commands `CIN` and/or `DIN`.

- READIO(16,2) Reads 4-bit data from port C and returns the value.
- READIO(16,3) Reads 4-bit data from port D and returns the value.
- READIO(16,4) Reads 4-bit data from port E and returns the value.

GPIB commands for 24-bit I/O port control

The GPIB commands related to the parallel I/O ports are summarized below.

Data Output

The following commands output data to ports (A to H). If you use C, D, E, F, G or H port as the output port, you must use the GPIB command `COUT` and/or `DOUT` to set the C, D port to output port.

- OUTAIO Outputs 8-bit data to port A.
- OUTBIO Outputs 8-bit data to port B.
- OUTCIO Outputs 4-bit data to port C.
- OUTDIO Outputs 4-bit data to port D.
- OUTEIO Outputs 8-bit data to port E.
- OUTFIO Outputs 16-bit data to port F.
- OUTGIO Outputs 20-bit data to port G.
- OUTHIO Outputs 24-bit data to port H.

Data Input

The following commands read data sent from an external device to an input port (C to E) and return the value to the GPIB. If you use the port as an input port, the port must be defined as an input port using the GPIB command `CIN` and/or `DIN`.

- OUTPINPCIO? Reads 4-bit data from port C and returns its value to the GPIB.
- OUTPINPDIO? Reads 4-bit data from port D and returns its value to the GPIB.
- OUTPINPEIO? Reads 8-bit data from port E and returns its value to the GPIB.

Setting Input/Output Directions of Ports C and D

The following commands set the input/output directions of ports C and D. When the power is turned ON, ports C and D are defined as input ports. [Preset] does not affect the setup. This setting is saved to an instrument state file using the Save function.

- CIN Defines port C as an input port.
- COUT Defines port C as an output port.
- DIN Defines port D as an input port.
- DOUT Defines port D as an output port.

Positive or Negative Logic Setting

You can set the logic level of the following ports and signal to negative or positive using the following GPIB command `NEGL` or `POSL`. When the power is turned ON, negative logic is set. (Preset) does not affect this setup. This setup is saved to an instrument state file using the Save function.

- Output ports A to H
- Input ports C and D
- PASS/FAIL signal

OUTPUT1 and OUTPUT2 Level Setting Commands

The following commands set the logic level of OUTPUT1 and OUTPUT2.

- `OUT1H` Sets OUTPUT1 to HIGH.
- `OUT1L` Sets OUTPUT1 to LOW.
- `OUT2H` Sets OUTPUT2 to HIGH.
- `OUT2L` Sets OUTPUT2 to LOW.

Checking Input to INPUT1

This command checks whether a pulse has been input to INPUT1. Send this command after a pulse has been input to INPUT1 and the return value will be "1". If no pulse has been input, it will be "0". Once "1" is read by this command, it will be reset to "0" until the next pulse is input.

8

Application Programming

This chapter provides the application programs for 4395A. According to the analyzer mode, the programs falls into the groups shown below:

- To Read Data Using the Marker Search Function
- To Perform Limit Test
- To Set List Sweep
- To Print Analyzer Display
- Programs for the network analyzer mode:

  To Perform 1 Pass 2 Port Calibration
  To Analyze a Filter
  To Analyze a Crystal Filter
  To Measure Gain Compression
- Programs for the spectrum analyzer mode:

  To Obtain Total Harmonic Distortion (THD)
  To Obtain an Integral of a Power
  To Obtain Adjacent Channel Power
  To Obtain Occupied Power Bandwidth
  To Calculate an S/N Ratio
- Programs for the impedance analyzer mode:

  To Perform Calibration
  To Measure Capacitance and the factor D
  To Measure a Varactor Diode Using DC Bias Sweep (With Option 010)
- File transfer function

  File transfer from the 4395A to the external controller
  File transfer from the external controller to the 4395A
  Listing of the files in the current directory of the 4395A

The end of this chapter lists the hints and notes for programming.

To Read Data Using the Marker Search Function

```
10    !
20    !Fig.8-1  To Read Data Using Marker Search Function
30    !
40    ASSIGN @Hp4395 TO 717   ! When iBASIC is used, change "717" to "800"
50    INPUT "ENTER CENTER FREQUENCY (Hz)",F_cent   !  Setting 4395A
60    INPUT "ENTER FREQUENCY SPAN (Hz)",F_span     !
70    OUTPUT @Hp4395;"CENT ";F_cent                !
80    OUTPUT @Hp4395;"SPAN ";F_span                !
90    OUTPUT @Hp4395;"*OPC?"
100   ENTER @Hp4395;Dum
110  !
120   OUTPUT @Hp4395;"CLES"
130   OUTPUT @Hp4395;"*SRE 4;ESNB 1"
140   ON INTR 7 GOTO Sweep_end   ! \ When iBASIC is used, change "7" to "8"
150   ENABLE INTR 7;2            ! /
160     OUTPUT @Hp4395;"SING"        ! Trigger a Measurement
170 Measuring:    GOTO Measuring     ! Measuring
180 Sweep_end:   !
190     OUTPUT @Hp4395;"MKR ON"
200     OUTPUT @Hp4395;"SEAM MAX"
210     OUTPUT @Hp4395;"OUTPMKR?"
220     ENTER @Hp4395;Val1,Val2,Swp
230     PRINT "Max Val:",Val1;"dB"
240     PRINT "Swp.Prmtr:",Swp,"Hz"
250   END
```

Figure 8-1. Sample Program : Reading Data Using Marker Search Function

Searching Maximum Value

```
190      OUTPUT @Hp4395;"MKR ON"
200      OUTPUT @Hp4395;"SEAM MAX"
```

Line 190 activates the marker and line 200 moves the marker to the maximum value on the trace.

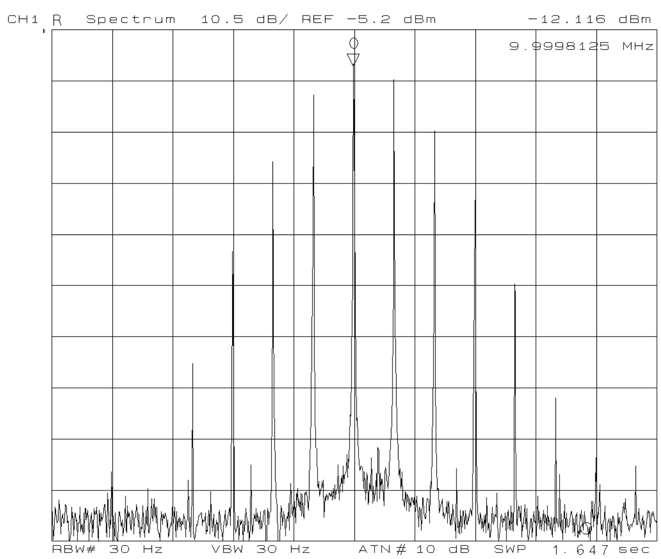


Figure 8-2. Marker on Trace

What are the other marker commands?

You can activate sub-markers and the Δmarker using the following commands:

`SMKR{1-7} ON`, `DMKR {ON|FIX|TRAC}`

You can move the marker using the following commands:[1]

• specified sweep parameter	`MKRPRM` *parameter*
• specified measurement point	`MKRP` *parameter*

You can move sub-markers using the following commands:[1]

• specified sweep parameter	`SMKRPRM{1-7}` *parameter*
• specified measurement point	`SMKRP{1-7}` *parameter*

You can move the Δmarker using the following commands:[1]

• specified sweep parameter	`DMKRPRM` *parameter*
• specified primary part of marker value	`DMKRVAL` *parameter*
• specified secondary part of marker value	`DMKRAUV` *parameter*

1 Before executing these commands, you must turn on the markers to be moved.

Reading Data

```
210      OUTPUT @Hp4395;"OUTPMKR?"
220      ENTER @Hp4395;Val1,Val2,Swp
```

The `OUTPMKR?` command returns the marker value in the following order: primary part of data, secondary part of data, and sweep parameter. See "Marker Readout" for details.

What are other marker value commands?

You can get the marker value using the following commands:

• get primary part of marker value	`MKRVAL?`
• get secondary part of marker value	`MKRAUV?`
• get sweep parameter	`MKRPRM?`
• get data point number	`MKRP?`

You can get the sub-marker value using the following commands:

• get primary part of sub-marker value	`SMKRVAL{1-7}?`
• get secondary part of sub-marker value	`SMKRAUV{1-7}?`
• get sweep parameter	`SMKRPRM{1-7}?`
• get data point number	`SMKRP{1-7}?`

You can get the Δmarker value using the following commands:

• get primary part of Δmarker value	`DMKRVAL?`
• get secondary part of Δmarker value	`DMKRAUV?`
• get sweep parameter	`DMKRPRM?`

Marker Readout

The values specified by the marker, sub-marker, or delta-mareker can be read using the following commands. See the following table which lists the amplitude value (value 1) and the auxiliary amplitude value (value 2) for each display format.

OUTPMKR?, OUTPSMKR?, OUTPDMKR?	Amplitude value (Value 1), Auxiliary amplitude value (Value 2), Sweep Parameter
MKRVAL?, SMKRVAL{1-7}?	Amplitude value (Value 1)
MKRAUV?, SMKRAUV{1-7}?	Auxiliary amplitude value (Value 2)

Table 8-1. Marker Readout

Analyzer Type	Display Format	Parameter of CIRF Command	Amplitude Value (Value 1)	Auxiliary Amplitude Value (Value 2)
Network Analyzer	Log Magnitude	—	Log Magnitude (dB)	0
	Phase Expanded Phase	—	Phase (degrees)	0
	Delay	—	Delay (seconds)	0
	Linear Magnitude	—	Linear Magnitude	0
	SWR	—	SWR	0
	Real	—	Real	0
	Imaginary	—	Imaginary	0
Network/ impedance Analyzer[1]	Smith Chart Polar Admittance	RI LIN LOG RX GB SWR	Real Linear Magnitude Log Magnitude (dB) Resistance (Ω) Conductance (S) SWR	Imaginary Phase (degrees) Phase (°) Reactance (Ω) Susceptance (S) Phase (°)
Spectrum Analyzer	Spectrum Measurement	—	Magnitude (dBm, dBV, dBμV, W, or V)[2]	0
	Noise Level Measurement	—	Magnitude (dBm, dBV, dBμV, W, or V)[2]	0

1 For the other format than listed above in the impedance analyzer mode, the marker readout has the unit of the selected parameter by Meas key.

2 Unit is specified by the SAUNIT command. (default: dBm)

To Perform Limit Test

```
10   !
20   !Fig.8-3  Limit Test
30   !
40    ASSIGN @Hp4395 TO 717  ! When iBASIC is used, change "717" to "800"
50   !
60    CLEAR SCREEN
70    PRINT USING "10A,15A,15A,15A";"Segment","Swp.Prmtr(Hz)","Upper","Lower"
80   !
90    DIM Table(1:18,1:3)
100   INPUT "Enter number of segments (<=18)",Numb
110   FOR I=1 TO Numb
120     GOSUB Loadlimit
130   NEXT I
140   !
150   LOOP
160      INPUT "Do you want to edit? (Y/N)",An$
170   EXIT IF An$="N" OR An$="n"
180      INPUT "Enter segment number(<=18)",I
190   IF Numb<I THEN Numb=I
200      GOSUB Loadlimit
210   END LOOP
220   !
230   OUTPUT @Hp4395;"EDITLIML"
240   OUTPUT @Hp4395;"LIMCLEL"
250   FOR K=1 TO Numb
260     OUTPUT @Hp4395;"LIMSADD"
270     OUTPUT @Hp4395;";LIMPRM ";Table(K,1)
280     OUTPUT @Hp4395;";LIMU ";Table(K,2)
290     OUTPUT @Hp4395;"LIML ";Table(K,3)
300     OUTPUT @Hp4395;"LIMSDON"
310   NEXT K
320   OUTPUT @Hp4395;"LIMEDONE"
330   OUTPUT @Hp4395;"LIMILINE ON"
340 !
350   INPUT "Connect DUT, and press Enter.",Dum$
360   OUTPUT @Hp4395;"CLES"
370   OUTPUT @Hp4395;"*SRE 4;ESNB 1"
380   ON INTR 7 GOTO Sweep_end  ! \ When iBASIC is used, change "7" to "8"
390   ENABLE INTR 7;2           ! /
400     OUTPUT @Hp4395;"SING"
410 Measuring:GOTO Measuring
420 Sweep_end:   !
430   OUTPUT @Hp4395;"LIMITEST ON"
440   DIM Dt(1:801,1:4)
450   OUTPUT @Hp4395;"OUTPLIMF?"         ! \ Output test results.
460   ENTER @Hp4395 USING "%,K";Dt(*)    ! /
470   OUTPUT @Hp4395;"OUTPFAIP?"
480   ENTER @Hp4395;Failp
490   IF Failp=0 THEN Passed
500   PRINT "           FAIL POINTS             "
510   FOR I=1 TO Failp
520     PRINT
530     PRINT "Swp. prmtr : ";Dt(I,1)
540     PRINT "   Results       Upper         Lower    "
550     PRINT TAB(5);Dt(I,2);TAB(17);Dt(I,3);TAB(32);Dt(I,4)
560   NEXT I
570 Passed: !
580   DISP "Program End"
```

Figure 8-3. Sample Program : Limit Test (1/2)

```
590   STOP
600   !
610 Loadlimit: !
620   INPUT "ENTER SWEEP PARAMETER (Hz)",Table(I,1)
630   INPUT "ENTER UPPER LIMIT VALUE",Table(I,2)
640   INPUT "ENTER LOWER LIMIT VALUE",Table(I,3)
650   PRINT I;TAB(11);Table(I,1);TAB(27);Table(I,2);TAB(42);Table(I,3)
660   RETURN
670 END
```

Figure 8-1. Sample Program : Limit Test (2/2)

Lines 60 and 70 print the limit table heads on the BASIC SCREEN.

Line 90 defines the table (array `Table(1:18,1:3)`) used to hold the limit values. It contains the sweep parameter, the upper limit value, and the lower limit value as follows:

Segment	Sweep Parameter	Upper Limit	Lower Limit
1	`Table(1,1)`	`Table(1,2)`	`Table(1,3)`
2	`Table(2,1)`	`Table(2,2)`	`Table(2,3)`
⋮	⋮	⋮	⋮

Lines 110 to 130 call the subroutine `Loadlimit` (line 610) to edit and print as many segments as you defined in line 100 (the analyzer can retain up to 18 segments).

```
Segment   Swp.Prmtr(Hz)  Upper          Lower
 1         2.E+6           0             -10
 2         3.E+6           10            -20
 3         4.E+6           10            -10
```

The loop, lines 150 to 210, determines if you want to edit the table and confirms that the segment is in the table.

Lines 230 to 330 transfers the limit table (edited using BASIC) to the analyzer (more specifically, lines 260 to 300 define the segmentation and the other lines establish the limit line).

The `OUTPLIMF?` command in line 450 returns the limit test result for failed points. The test results are in the following order: sweep parameter, result (0 for fail, -1 for no test), upper limit, and lower limit.

The `OUTPFAIP?` command in line 470 returns the number of failed points. (When the limit test result is PASS, it returns 0 and the program goes to `Passed`.) Then the array `Dt` is printed with as many lines as the transferred data.

Lines 510 to 560 print the limit test result as follows:

```
Swp.Prmtr(Hz) : 1.1925E+7
   Result          Upper           Lower
     0              20              -40

Swp.Prmtr(Hz) : 1.2125E+7
   Result          Upper           Lower
     0              20              -40
```

What are other commands used to retrieve the test results?

Instead of reading the limit test results for failed points by using the `OUTPLIMF?` command, you can read out the test result using the following commands:

- At all measurement points: `OUTPLIML?`
- At marker position: `OUTPLIMM?`

Both commands return the sweep parameter, result (1 for pass, 0 for fail, −1 for no test), upper limit, and lower limit.

```
 10   !
 20   !Fig.8-4  List Sweep
 30   !
 40   ASSIGN @Hp4395 TO 717  ! When iBASIC is used, change "717" to "800"
 50   !
 60   OUTPUT @Hp4395;"SA"
 70   CLEAR SCREEN
 80   PRINT "Segment";TAB(9);"Center(Hz)";TAB(20);"Span(Hz)";TAB(30);"Points";
 90   PRINT TAB(39);"Power(dBm)";TAB(50);"RBW(Hz)"
100   !
110   DIM Table(1:31,1:5)
120   INPUT "Enter number of segments (<=15)",Numb
130   FOR I=1 TO Numb
140     GOSUB Loadlist
150   NEXT I
160     !
170   LOOP
180     INPUT "Do you want to edit? (Y/N)",An$
190   EXIT IF An$="N" OR An$="n"
200     INPUT "Enter segment number(<=15)",I
210     IF Numb<I THEN Numb=I
220     GOSUB Loadlist
230   END LOOP
240    !
250   OUTPUT @Hp4395;"EDITLIST"
260   OUTPUT @Hp4395;"CLEL"
270   FOR K=1 TO Numb
280     OUTPUT @Hp4395;"SADD"
290     OUTPUT @Hp4395;"CENT ";Table(K,1)
300     OUTPUT @Hp4395;"SPAN ";Table(K,2)
310     OUTPUT @Hp4395;"POIN ";Table(K,3)
320     OUTPUT @Hp4395;"POWE ";Table(K,4)
330     OUTPUT @Hp4395;"BW ";Table(K,5)
340     OUTPUT @Hp4395;"SDON"
350   NEXT K
360   OUTPUT @Hp4395;"EDITDONE"
370   OUTPUT @Hp4395;"SWPT LIST"
380 !
390   INPUT "Connect DUT, and press Enter.",Dum$
400   OUTPUT @Hp4395;"CLES"
410   OUTPUT @Hp4395;"*SRE 4;ESNB 1"
420   ON INTR 7 GOTO Sweep_end  ! \ When iBASIC is used, change "7" to "8"
430   ENABLE INTR 7;2           ! /
440     OUTPUT @Hp4395;"SING"
450 Measuring:GOTO Measuring
460 Sweep_end:   !
470 DISP "Program End"
480   STOP
490   !
500 Loadlist: !
510   INPUT "Enter center frequency(Hz)",Table(I,1)
520   INPUT "Enter frequency span(Hz)",Table(I,2)
530   INPUT "Enter number of points",Table(I,3)
540   INPUT "Enter power level(dBm)",Table(I,4)
550   INPUT "Enter resolution band width(Hz)",Table(I,5)
560   PRINT I;TAB(11);Table(I,1);TAB(20);Table(I,2);TAB(30);Table(I,3);
570   PRINT TAB(40);Table(I,4);TAB(50);Table(I,5)
580   RETURN
590   END
```

Figure 8-4. Sample Program : List Sweep

Lines 70 to 90 print the list table heads on the BASIC screen.

Line 110 defines the table (array `Table(1:31,1:5)`) used to hold the list values. It contains the center frequency, frequency span, number of points, power level, and resolution band width of each segment as follows:

Segment	Center Frequency	Frequency Span	Number of Points	Power Level	Resolution Band Width
1	`Table(1,1)`	`Table(1,2)`	`Table(1,3)`	`Table(1,4)`	`Table(1,5)`
2	`Table(2,1)`	`Table(2,2)`	`Table(2,3)`	`Table(2,4)`	`Table(2,5)`
⋮	⋮	⋮	⋮	⋮	⋮

Lines 130 to 150 call the subroutine `Loadlist` (line 500) to edit and print as many segments as you defined in line 120 (The analyzer can retain up to 15 segments in the spectrum analyzer mode, up to 51 segments in the other analyzer modes).

```
Segment  Center(Hz)  Stop(Hz)  Points  Power(dBm)  RBW(Hz)
 1         100         20       100       0          100
 2         10000       1000     300       0          300
 3         1000000     1000     400       0          100
```

The loop, lines 170 to 230, determines if you want to edit the table and confirms that the segment is in the table.

Lines 250 to 370 transfers the list table (edited using BASIC) is to the analyzer (more specifically, lines 280 to 340 define the segmentation and the other lines establish the list table).

What are other commands are used to set the list values?

- When setting segment frequencies, instead of setting the center/span definition by using the `CENT` *parameter* / `SPAN` *parameter* commands, you can define start/stop frequency by using:

  1. `STAR` *parameter* / `STOP` *parameter* commands
  2. `MKRSTAR` / `MKRSTOP` commands (Maker to start/stop)

- When setting the IF band width (with the analyzer in network analyzer mode), use the `BW` *parameter* command.

To Print Analyzer Display

This section describes how to print the information on the analyzer display using GPIB commands.

Printer Preparation

1. Connect a printer using a parallel cable.
2. Turn the printer on.

Execute Print

To print the screen, execute the following command.

```
OUTPUT 717;"PRINALL"
```

Set the GPIB address when you execute from an external controller.

To Observe Printing

The HP Basic program shown below gives an example to detect printing end by using an SRQ interrupt. The interrupt is generated upon the printing bit of the operation status event register is enabled.

```
10   !
20   !Fig.8-5  To Observe Printing
30   !
40   ASSIGN @Hp4395 TO 717 ! When iBASIC is used, change "717" to "800".
50   !
60   OUTPUT @Hp4395;"CLES"
70   OUTPUT @Hp4395;"OSNT 512" !Catch High to Low Transition
80   OUTPUT @Hp4395;"OSPT 0"   !Disable Low to High Transitions
90   OUTPUT @Hp4395;"OSE 512"  !Enable OS Event Reg.
100   OUTPUT @Hp4395;"*SRE 128" !Enable OSR bit
120   ON INTR 7 GOTO La1
130   ENABLE INTR 7;2
140   OUTPUT @Hp4395;"PRINALL"
150   La1:!
160   GOTO La1
170   DISP "PRINT COMPLETE"
180   !
190   END
```

Figure 8-5. Sample Program : To Observe Printing

Programs for the Network Analyzer Mode

To Perform 1 Pass 2 Port Calibration

The sample program in this section performs 1 pass 2 port calibration, one of the variety of calibrations available for the 4395A in the network analyzer mode.

```
10  !
20  !Fig.8-6  1PASS 2PORT CALIBRATION
30  !
40  ASSIGN @Hp4395 TO 717 ! When iBASIC is used, change "717" to "800".
50  !
60  OUTPUT @Hp4395;"PRES"
70  OUTPUT @Hp4395;"NA"
100 !
110 Cal_1p2p: !
120 CLEAR SCREEN
130    BEEP
140    PRINT "One Path 2 Port CAL"
150    PRINT "    1:Reflect'N"
160    DISP "Open cal :press continue"
170    PAUSE
180     OUTPUT @Hp4395;"CALI ONE2;REFL;CLASS11A"
190      OUTPUT @Hp4395;"*OPC?"
200      ENTER @Hp4395;Dummy
210    DISP "Short cal :press continue"
220    PAUSE
230     OUTPUT @Hp4395;"CLASS11B"
240      OUTPUT @Hp4395;"*OPC?"
250      ENTER @Hp4395;Dummy
260    DISP "Load cal  :press continue"
270    PAUSE
280     OUTPUT @Hp4395;"CLASS11C"
290      OUTPUT @Hp4395;"*OPC?"
300      ENTER @Hp4395;Dummy
310     OUTPUT @Hp4395;"REFD"     ! SAVE Reflect'N Result
320      OUTPUT @Hp4395;"*OPC?"
330      ENTER @Hp4395;Dummy
340 !
350    BEEP
360    PRINT "    2:Transmission"
370    DISP "CONNET THRU and press continue"
380    PAUSE
390     OUTPUT @Hp4395;"TRAN;FWDT"
400      OUTPUT @Hp4395;"*OPC?"
410      ENTER @Hp4395;Dummy
420     OUTPUT @Hp4395;"FWDM"
430      OUTPUT @Hp4395;"*WAI"
440     OUTPUT @Hp4395;"TRAD"     ! SAVE Transmission
450      OUTPUT @Hp4395;"*OPC?"
460      ENTER @Hp4395;Dummy
470 !
480 ! ISOLATION
490    Num=1
500    BEEP
```

Figure 8-6. Sample Program : 1 Pass 2 Port Calibration (1/2)

```
510    PRINT "    3:Isolation"
520     INPUT " Select number, 1:OMIT or 2:FWD ISOL'N",Num
530   SELECT Num
540     CASE 1
550      OUTPUT @Hp4395;"ISOL;OMII"   ! Omits Isolation
560     CASE 2
570      DISP "Connect STD, then press continue"
580      PAUSE
590      OUTPUT @Hp4395;"ISOL;FWDI"   ! Measure Isolation Standard
600     CASE ELSE
610      BEEP
620      GOTO 520
630   END SELECT
640     OUTPUT @Hp4395;"*WAI"
650     OUTPUT @Hp4395;"ISOD"     ! SAVE Isolation
660      OUTPUT @Hp4395;"*WAI"
670     OUTPUT @Hp4395;"SAV2"     ! SAVE CAL DATA
680      OUTPUT @Hp4395;"*OPC?"
690      ENTER @Hp4395;Dummy
700 !
710  DISP "PROGRAM END"
720  END
```

Sample Program : 1 Pass 2 Port Calibration (2/2)

The measurement and calibration should be performed with the controller

When performing calibration, each calibration factor should be measured and stored in synchronization with the controller. See Chapter 4 for the detailed procedure required to synchronize with the controller.

For detailed information on the calibration, see *Operation Manual*.

To Analyze a Filter

The sample program in this section performs the waveform analysis of a bandpass filter. The filter is assumed to have a frequency characteristic shown below.

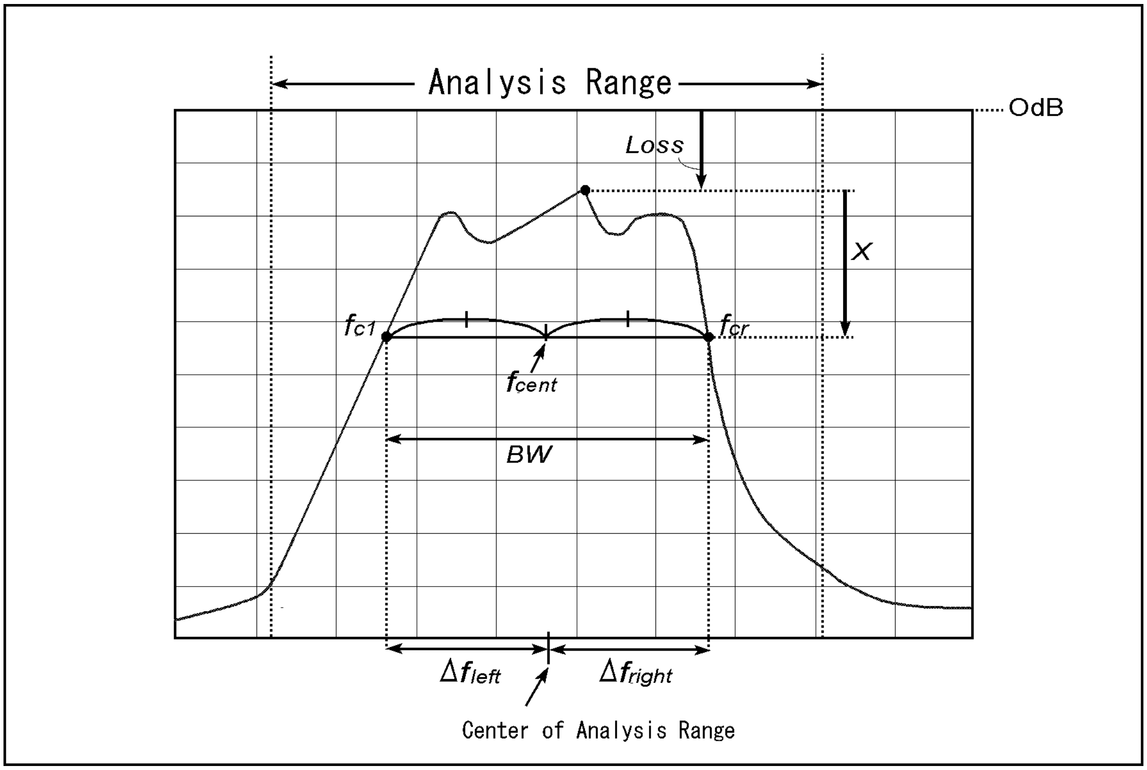


Figure 8-7. Conceptual View of a Bandpass-filtered Waveform

The table below lists the results obtained from the program.

Parameter	Description
$Loss$	Insertion loss
BW	xdB down bandwidth
f_{cent}	Center frequency
Q	Q (Quality factor)
Δf_{left}	Frequency difference between the left cutoff point and the middle of the analysis range.
Δf_{right}	Frequency difference between the right cutoff point and the middle of the analysis range.

Q is calculated using the following equation:

$$Q = \frac{\sqrt{f_{cl} \times f_{cr}}}{BW}$$

```
10  !Fig.8-8  To Analyze A Filter
20  ASSIGN @Agt4395 TO 717  ! When iBASIC is used, change "717" to "800"
30  !
40  Center=7.E+7
50  Spanv=200000.
60  !
70  OUTPUT @Agt4395;"PRES"
80  OUTPUT @Agt4395;"DISA HIHB"
90  GOSUB Setup
100  GOSUB Calibration
110  GOSUB Measurement
120  DISP "PROGRAM END"
130  STOP
140 !
150 Setup:!
160    OUTPUT @Agt4395;"NA"
170    OUTPUT @Agt4395;"MEAS BR"
180    OUTPUT @Agt4395;"CENT ";Center
190    OUTPUT @Agt4395;"SPAN ";Spanv
200    OUTPUT @Agt4395;"*OPC?"
210    ENTER @Agt4395;Dummy
220 RETURN
230 !
240 Calibration: !
250 CLEAR SCREEN
260    BEEP
270    PRINT "Calibration"
280    OUTPUT @Agt4395;"CALI RESP"
290    DISP "Thru cal :press continue"
300    PAUSE
310     OUTPUT @Agt4395;"STANC"
320      OUTPUT @Agt4395;"*WAI"
330     OUTPUT @Agt4395;"RESPDONE"     ! SAVE CAL DATA
340      OUTPUT @Agt4395;"*OPC?"
350      ENTER @Agt4395;Dummy
360 RETURN
370 !
380 !
390 Measurement: !
400 CLEAR SCREEN
410   BEEP
420   DISP "CONNECT DEVICE (FILTER) and press continue"
430   PAUSE
440   OUTPUT @Agt4395;"SING"
450   OUTPUT @Agt4395;"*OPC?"
460   ENTER @Agt4395;Dummy
470   OUTPUT @Agt4395;"AUTO"
480   !
490   ! Anarisys range configuration
500   BEEP
510   Lowerl=6.9E+7
520   Higherl=7.1E+7
530   PRINT
540   PRINT "ANARISYS RANGE"
550   PRINT Lowerl;" - ";Higherl
560   INPUT "Change analysis range?  Y/N",A$
570     IF A$="Y" THEN
580       INPUT "Lower limit [Hz]",Lowerl
590       INPUT "Higher limit [Hz]",Higherl
600     END IF
610   OUTPUT @Agt4395;"MKR ON;MKRO DATA"
```

Figure 8-8. Sample Program : To Analyze a Filter (1/2)

```
620   OUTPUT @Agt4395;"MKRPRM ";Lowerl
630   OUTPUT @Agt4395;"DMKR ON"
640   OUTPUT @Agt4395;"MKRPRM ";Higherl
650   OUTPUT @Agt4395;"SEARSTR"
660   OUTPUT @Agt4395;"PARS ON"
670   !
680   OUTPUT @Agt4395;"SEAM MAX"
690 !
700   OUTPUT @Agt4395;"DMKR ON"
710   OUTPUT @Agt4395;"WIDVTYPE FIX"
720   Trgv=-3
730   INPUT "Input cut-off value[dB]",Trgv
740   OUTPUT @Agt4395;"WIDV ";Trgv;"DB"
750   OUTPUT @Agt4395;"WIDT ON"
760 !
770   OUTPUT @Agt4395;"OUTPMWID?"
780   ENTER @Agt4395;A,B,C
790   Lfreq=B-A/2-(Higherl+Lowerl)/2
800   Rfreq=B+A/2-(Higherl+Lowerl)/2
810   OUTPUT @Agt4395;"MKRVAL?"
820   ENTER @Agt4395;Loss
830 !
840  PRINT "RESULT"
850  PRINT "==========================================================="
860  PRINT "BAND WIDTH : ";A;"Hz"
870  PRINT "CENTER     : ";B;"Hz"
880  PRINT "Quality    : ";C
890  PRINT "Loss       : ";Loss;"dB"
900  PRINT "deltaL. F  : ";Lfreq;"Hz"
910  PRINT "deltaR. F  : ";Rfreq;"Hz"
920  RETURN
930 !
940 END
```

Sample Program : To Analyze a Filter (2/2)

Lines 240 to 360 performs calibration. This sample program performs THRU calibration. An appropriate calibration should be applied depending on your DUT.

This program employs the width function of the marker. When this function is enabled, the same information as the output of this program will appear on the display of the 4395A.

Lines 490 to 660 specify the range of marker search.

Lines 680 to 800 calculate the bandwidth (BW), center frequency of the bandwidth (cent), Q factor (Q), left hand bandwidth (ΔL.F), and right hand bandwidth (ΔR.F) using the bandwidth search function of the marker.

Lines 810 to 820 calculate the insertion loss.

To Analyze a Crystal Filter

The sample program in this section performs a waveform analysis of a bandpass filter which has a passband ripple, such as a crystal filter. Before running this program, verify the analysis range, that is the measurement range where you wish to obtain the spurious and rejection levels. The information can be specified when prompted by the program.

- Spurious level is the frequency difference from the insertion loss to the maximum level between f_2 and the right edge of analysis range (f_2 can take a value in the program).

- Rejection level is the frequency difference from the insertion loss to the maximum level in the range from the left edge of analysis range to f_1 (f_1 can take a value in the program).

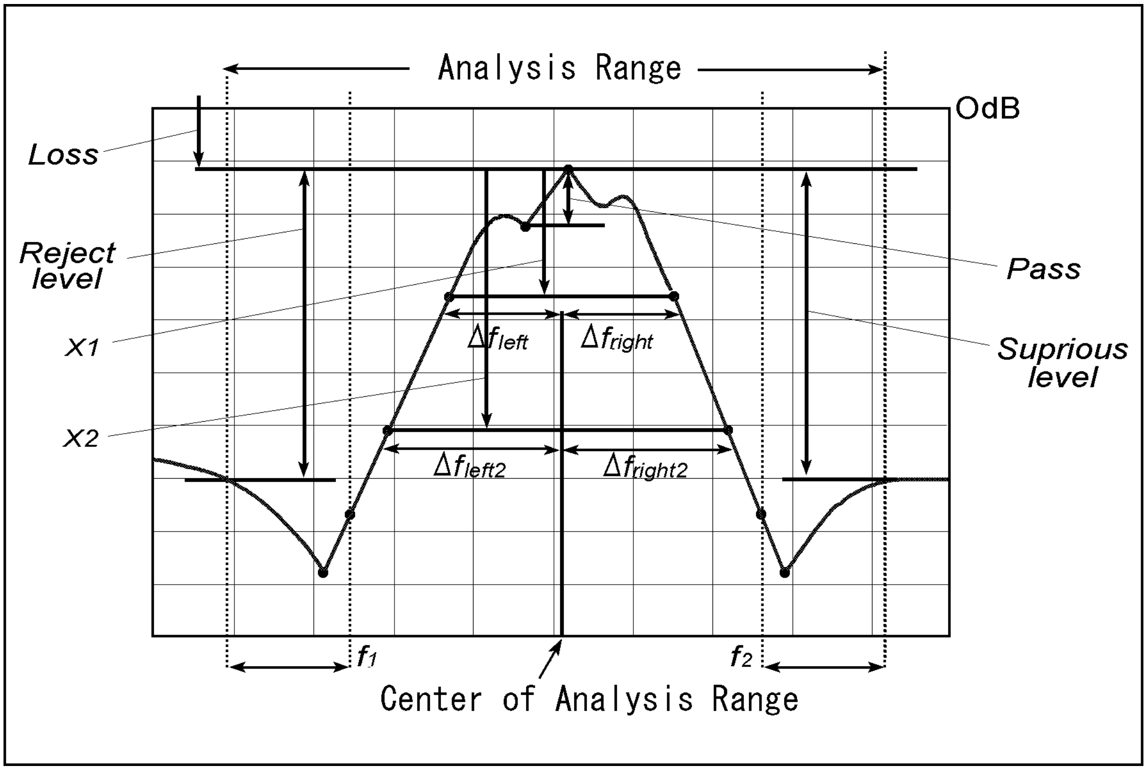


Figure 8-9. Analyzing a Crystal Filter

The table below lists the results obtained from the program.

Parameter	Description
$Loss$	Insertion loss
BW	x_1 dB down bandwidth
f_{cent}	Center frequency
Q	Q
Δf_{left}	Frequency difference between the left cutoff point (f_{cl}) and the middle of the range.
Δf_{right}	frequency difference between the right cutoff point (f_{cr}) and the middle of the range.
Δf_{left2}	Frequency difference between the left cutoff point (f_{cl2}) and the middle of the range.
Δf_{right2}	frequency difference between the left cutoff point (f_{cr2}) and the middle of the range.
$Pass$	Passband ripple
$Reject$	Rejection level
$Spurious$	Spurious level
$Pole_{x1}$	First negative peak found to the left of the maximum point.
$Pole_{stim1}$	Stimulus of $Pole_{x1}$.
$Pole_{x2}$	First negative peaks found to the right of the maximum point.
$Pole_{stim2}$	Stimulus of $Pole_{x2}$.

```
10  !
20  !Fig.8-10  Xtal filater analisys
30  !
40  ASSIGN @Hp4395 TO 717   !When iBASIC is used, change "717" to "800"
50  !
60  Center=7.E+7
70  Spanv=200000.
80  Trgv(1)=-3       ! 1st cut off value [dB]
90  Trgv(2)=-6       ! 2nd cut off value [dB]
100 D=-50            ! Pole value [dB]
110 F1=6.998E+7      ! Range parameter to get reject level [Hz]
120 F2=7.003E+7      ! Range parameter to get suprious level [Hz]
130 !
140 !
150 OUTPUT @Hp4395;"PRES"
160 OUTPUT @Hp4395;"DISA HIHB"
170 GOSUB Setup
180 GOSUB Calibration
190 GOSUB Measurement
200 STOP
210 !
220 Setup:!
230    OUTPUT @Hp4395;"NA"
240    OUTPUT @Hp4395;"MEAS BR"
250    OUTPUT @Hp4395;"CENT ";Center
260    OUTPUT @Hp4395;"SPAN ";Spanv
270    OUTPUT @Hp4395;"*OPC?"
280    ENTER @Hp4395;Dummy
290 RETURN
300 !
310 Calibration: !
320 CLEAR SCREEN
330    BEEP
340    PRINT "Calibration"
350    OUTPUT @Hp4395;"CALI RESP"
360    DISP "Thru cal :press continue"
370    PAUSE
380     OUTPUT @Hp4395;"STANC"
390      OUTPUT @Hp4395;"*WAI"
400     OUTPUT @Hp4395;"RESPDONE"     ! SAVE CAL DATA
410      OUTPUT @Hp4395;"*OPC?"
420      ENTER @Hp4395;Dummy
430 RETURN
440 !
450 Measurement: !
460   DIM Trgv(2),Lfreq(2),Rfreq(2)
470 CLEAR SCREEN
480   BEEP
490   DISP "CONNECT DEVICE (FILTER) and press continue"
500   PAUSE
510   OUTPUT @Hp4395;"SING"
520   OUTPUT @Hp4395;"*OPC?"
530   ENTER @Hp4395;Dummy
540   OUTPUT @Hp4395;"AUTO"
550   !
560   ! Anarisys range configuration
570   BEEP
580   Lowerl=6.995E+7
590   Higherl=7.006E+7
600   PRINT
610   PRINT "ANARISYS RANGE"
```

Figure 8-10. Sample Program : Crystal Filter Analysis (1/4)

```
620  PRINT Lowerl;" - ";Higherl
630  INPUT "Do you change this range?  Y/N",A$
640    IF A$="Y" THEN
650      INPUT "Lower limit",Lowerl
660      INPUT "Higher limit",Higherl
670      GOTO 610
680    END IF
690  OUTPUT @Hp4395;"MKR ON;MKRO DATA"
700  OUTPUT @Hp4395;"MKRPRM ";Lowerl
710  OUTPUT @Hp4395;"DMKR ON"
720  OUTPUT @Hp4395;"MKRPRM ";Higherl-Lowerl
730  OUTPUT @Hp4395;"SEARSTR"
740  OUTPUT @Hp4395;"PARS ON"
750  !
760  OUTPUT @Hp4395;"SEAM MAX"
770  OUTPUT @Hp4395;"DMKR ON"
780  OUTPUT @Hp4395;"WIDVTYPE FIX"
790   OUTPUT @Hp4395;"WIDT ON"
800 FOR I=1 TO 2
810   OUTPUT @Hp4395;"WIDV ";Trgv(I)
820   OUTPUT @Hp4395;"OUTPMWID?"
830   ENTER @Hp4395;A,B,C
840   Lfreq(I)=B-A/2
850   Rfreq(I)=B+A/2
860   IF I=1 THEN Bw=A
870 NEXT I
880   OUTPUT @Hp4395;"MKRVAL?"
890   ENTER @Hp4395;Loss
900 OUTPUT @Hp4395;"WIDT OFF"
910 OUTPUT @Hp4395;"SMKR1 OFF;SMKR2 OFF;SMKR3 OFF"
920 !
930 ! Pass level serch
940 OUTPUT @Hp4395;"PKTHVAL ";Trgv(2)
950 OUTPUT @Hp4395;"PKTHRE ON"
960 OUTPUT @Hp4395;"PKPOL NEG"
970 OUTPUT @Hp4395;"SEAM PEAK"
980 OUTPUT @Hp4395;"SEANPKL"
990 OUTPUT @Hp4395;"MKRVAL?"
1000 ENTER @Hp4395;Ripplel
1010 OUTPUT @Hp4395;"MKRPRM?"
1020  ENTER @Hp4395;Stimmax
1030 OUTPUT @Hp4395;"SEANPKR"
1040  OUTPUT @Hp4395;"SMKRVAL1?"
1050  ENTER @Hp4395;Rippler
1060   IF Ripplel<Rippler THEN
1070     Pass=Ripplel-Loss
1080   ELSE
1090     Pass=Rippler-Loss
1100   END IF
1110 OUTPUT @Hp4395;"PKTHRE OFF"
1120 !
1130 ! Pole search
1140   Polel=0
1150   Xstiml=0
1160   Poler=0
1170   Xstimr=0
1180   OUTPUT @Hp4395;"CLES"
1190     OUTPUT @Hp4395;"*OPC?"
1200     ENTER @Hp4395;Dummy
1210   OUTPUT @Hp4395;"*SRE 4;ESNB 96"
1220   ON INTR 7 GOTO End_search   ! \Set-up to Exit when fails to search poles.
```

Sample Program : Crystal Filter Analysis (2/4)

```
1230   ENABLE INTR 7;2            ! /When iBASIC, change "7" to "8"
1240   OUTPUT @Hp4395;"SEAM PEAK"
1250   LOOP
1260     OUTPUT @Hp4395;"SEANPKL"
1270     OUTPUT @Hp4395;"MKRVAL?"
1280     ENTER @Hp4395;Polel
1290     OUTPUT @Hp4395;"MKRPRM?"
1300     ENTER @Hp4395;Xstiml
1310   EXIT IF Polel<D AND Xstiml<Stimmax
1320   END LOOP
1330   LOOP
1340     OUTPUT @Hp4395;"SEANPKR"
1350     OUTPUT @Hp4395;"MKRVAL?"
1360     ENTER @Hp4395;Poler
1370     OUTPUT @Hp4395;"MKRPRM?"
1380     ENTER @Hp4395;Xstimr
1390   EXIT IF Poler<D AND Xstimr>Stimmax
1400   END LOOP
1410 End_search:  !
1420   IF Xstiml=0 AND Xstimr=0 THEN PRINT "No Pole!"
1430 !
1440   OUTPUT @Hp4395;"MKRPRM ";F2
1450   OUTPUT @Hp4395;"MKRVAL?"
1460   ENTER @Hp4395;Sup_ref
1470 !
1480 ! Reject level search
1490   IF F1-Lowerl<0 THEN GOTO Printing
1500 OUTPUT @Hp4395;"MKRPRM";Lowerl
1510   OUTPUT @Hp4395;"MKRVAL?"
1520   ENTER @Hp4395;Rej_ref
1530 OUTPUT @Hp4395;"DMKR ON"
1540 OUTPUT @Hp4395;"MKRPRM";F1-Lowerl
1550 OUTPUT @Hp4395;"SEARSTR"
1560 OUTPUT @Hp4395;"SEAM MAX"
1570 OUTPUT @Hp4395;"MKRVAL?"
1580 ENTER @Hp4395;Reject
1590 Reject=Reject+Rej_ref
1600 !
1610 ! Suprious level search
1620 OUTPUT @Hp4395;"MKRPRM";F2-Lowerl
1630 OUTPUT @Hp4395;"DMKR ON"
1640   IF Higherl-F2<0 THEN GOTO Printing
1650 OUTPUT @Hp4395;"MKRPRM";Higherl-F2
1660 OUTPUT @Hp4395;"SEARSTR"
1670 OUTPUT @Hp4395;"SEAM MAX"
1680 OUTPUT @Hp4395;"MKRVAL?"
1690 ENTER @Hp4395;Sprious
1700 Sprious=Sprious+Sup_ref
1710 !
1720 Printing:!
1730   IF F1<Lowerl OR F2>Higherl THEN PRINT "F1 or F2 is invalid"
1740 PRINT "RESULT"
1750 PRINT "========================================================="
1760 PRINT "BAND WIDTH (1st val.) : ";Bw;"Hz"
1770 PRINT "CENTER                : ";B;"Hz"
1780 PRINT "Quality               : ";C
1790 PRINT "LOSS                  : ";Loss;"dB"
1800 PRINT "deltaL. F1            : ";Lfreq(1);"Hz"
1810 PRINT "deltaR. F1            : ";Rfreq(1);"Hz"
1820 PRINT "deltaL. F2            : ";Lfreq(2);"Hz"
1830 PRINT "deltaR. F2            : ";Rfreq(2);"Hz"
```

Sample Program : Crystal Filter Analysis (3/4)

```
1840 PRINT "PASS                 : ";Pass;"dB"
1850 PRINT "REJECT               : ";Reject;"dB"
1860 PRINT "SPRIOUS              : ";Sprious;"dB"
1870 PRINT "Right POLE Value     : ";Poler;"dB"
1880 PRINT "Right POLE Stimuras  : ";Xstimr;"Hz"
1890 PRINT "Left POLE Value      : ";Polel;"dB"
1900 PRINT "Left POLE Stimuras   : ";Xstiml;"Hz"
1910 !
1920 DISP "PROGRAM END"
1930 !
1940 END
```

Sample Program : Crystal Filter Analysis (3/4)

This program is an enhancement of the "To Analyze a Filter" program. So lines 150 to 890 work same as the "To Analyze a Filter" program.

Lines 1130 to 1410 calculate the Ripple value of the filter.

Lines 1130 to 1410 calculate the Pole value of the filter. In this routine, when 4395A fails to search Poles, lines 1180 to 1220 let 4395A know it using with SRQ. About SRQ, see "To Wait for Sweep End" in Chapter 5.

Lines 1470 to 1580 calculate Rejection level and line 1600 to 1690 calculate Sprious level.

To Measure Gain Compression

The sample program in this section measures gain compression using the power sweep function. This program uses two channels. One channel is used to measure an absolute value of the output signal from the DUT. The other channel measures a ratio of the input and output signal levels. The program shows the power level of the input and output signal at the -1 dB gain compression point.

In the figure below, the marker shows the input and output power levels at the -1 dB gain compression point relative to the left Δmarker.

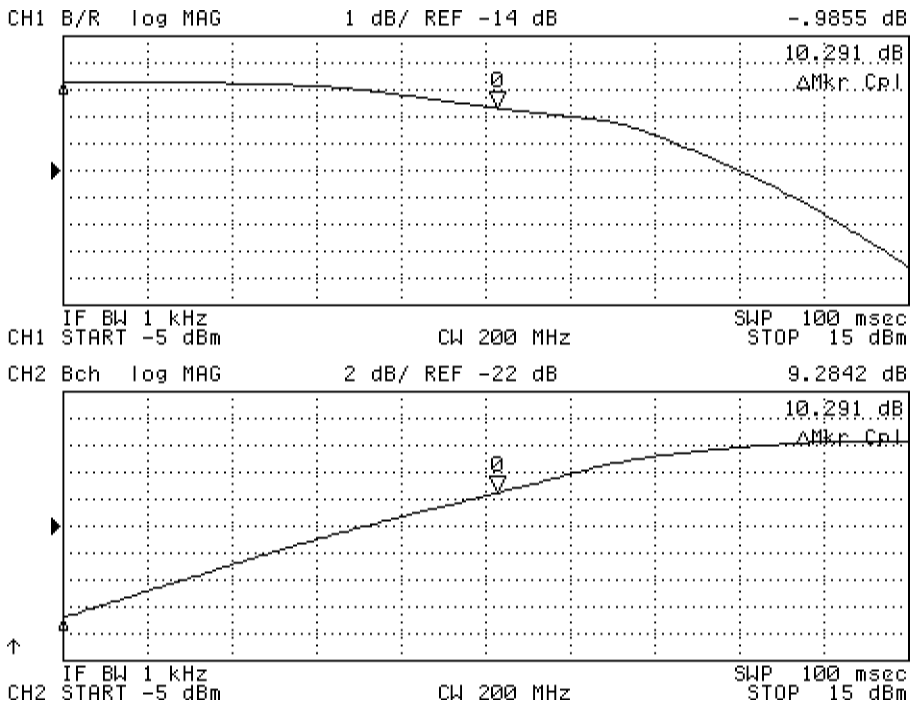


Figure 8-11. Gain Compression Measurement

```
10  !
20  !Fig.8-12  Gain Compression Measurement
30  !
40  ASSIGN @Hp4395 TO 717  ! When iBASIC is used, change "717" to "800"
50  !
60  Targ=-1      ! Gain compressoion target value [dB]
70  !
80  OUTPUT @Hp4395;"PRES"
90  OUTPUT @Hp4395;"DISA HIHB"
100 GOSUB Setup
110 GOSUB Calibration
120 GOSUB Measurement
130  STOP
140 !
150 Setup:!
160    OUTPUT @Hp4395;"DISA ALLI"
170    OUTPUT @Hp4395;"NA"
180    OUTPUT @Hp4395;"COUC ON"
190    OUTPUT @Hp4395;"CHAN1"
200    OUTPUT @Hp4395;"CWFREQ 200E6"
210    OUTPUT @Hp4395;"SWPT POWE"
220    OUTPUT @Hp4395;"STAR -5"
230    OUTPUT @Hp4395;"STOP 15"
240    OUTPUT @Hp4395;"FMT LOGM"
250    OUTPUT @Hp4395;"BW 10E3"
260    OUTPUT @Hp4395;"MEAS BR"
270    !
```

Figure 8-12. Sample Program : Gain Compression Measurement (1/2)

```
280    OUTPUT @Hp4395;"CHAN2"
290    OUTPUT @Hp4395;"MEAS B"
300    OUTPUT @Hp4395;"MKR ON"
310    OUTPUT @Hp4395;"CHAN1"
320    OUTPUT @Hp4395;"MKR ON"
330    OUTPUT @Hp4395;"DUAC ON"
340    OUTPUT @Hp4395;"COUC OFF"
350    OUTPUT @Hp4395;"MKRCOUP ON"
360    OUTPUT @Hp4395;"*OPC?"
370    ENTER @Hp4395;Dummy
380 RETURN
390 !
400 Calibration: !
410 CLEAR SCREEN
420    BEEP
430    PRINT "Calibration"
440    OUTPUT @Hp4395;"CALI RESP"
450    DISP "Thru cal :press continue"
460    PAUSE
470     OUTPUT @Hp4395;"STANC"
480      OUTPUT @Hp4395;"*WAI"
490     OUTPUT @Hp4395;"RESPDONE"     ! SAVE CAL DATA
500      OUTPUT @Hp4395;"*OPC?"
510      ENTER @Hp4395;Dummy
520 RETURN
530 !
540 Measurement: !
550 CLEAR SCREEN
560   BEEP
570   DISP "CONNECT DEVICE and press continue"
580   PAUSE
590   FOR I=2 TO 1 STEP -1
600     OUTPUT @Hp4395;"CHAN";CHR$(48+I)
610     OUTPUT @Hp4395;"SING"
620     OUTPUT @Hp4395;"*WAI"
630     OUTPUT @Hp4395;"AUTO"
640   NEXT I
650   !
660   OUTPUT @Hp4395;"SEAM MAX"
670   OUTPUT @Hp4395;"DMKR ON"
680   OUTPUT @Hp4395;"SEATARG ";Targ
690   OUTPUT @Hp4395;"MKRPRM?"
700   ENTER @Hp4395;Pow1
710   OUTPUT @Hp4395;"ANAOCH2"
720   OUTPUT @Hp4395;"MKRVAL?"
730   ENTER @Hp4395;Ampval
740   OUTPUT @Hp4395;"ANAOCH1"
750   OUTPUT @Hp4395;"DUAC ON"
760   PRINT ""
770   PRINT Targ;"dB Gain Compression ";Pow1;"dBm"
780   PRINT "Absolute Output Power";Ampval;"dB"
790   DISP "END"
800   RETURN
810   !
820   END
```

Sample Program : Gain Compression Measurement (2/2)

Lines 150 to 380 specify basic measurement settings.

Line 410 to 520 perform calibration. The program performs only THRU calibration to simplify the procedure.

Lines 590 to 640 perform sweep. Lines 660 and after move the marker to read the parameters at the point it is located. This program uses `ANAOCH` command to switch the channel from which data is retrieved. Refer to the `ANAOCH` entry in the command reference in this manual.

Programs for the Spectrum Analyzer Mode

This section provides sample programs for your convenience when calculating some spectrum analysis factors.

To Obtain Total Harmonic Distortion (THD)

Most transmitting devices and signal sources contain harmonics as shown in Figure 8-13.

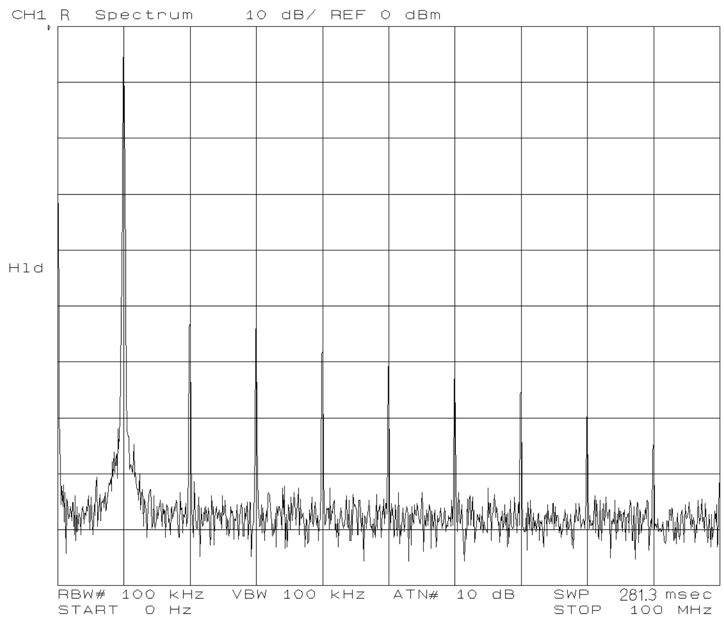


Figure 8-13. Total Harmonic Distortion in a Signal

This program computes the total harmonic distortion (THD) as defined by the following equation:

$$THD = \frac{\sqrt{V_2^2 + V_3^2 + \cdots}}{V_1} \times 100\,[\%] \qquad (8-1)$$

Where,

V_1 Fundamental [V]
V_2 The second harmonic [V]
V_3 The third harmonic [V]
$\vdots$

THD takes into account the power in all the harmonics. Because an infinite number of the harmonics cannot be measured, a finite number will have to suffice.

Before running the program, measure the signal and display the fundamental and harmonics to be computed on the analyzer display (in the dBm format).

```
10   !
20   !Fig.8-14  Total Harmonic Distortion
30   !
40     Vf=1
50     ASSIGN @Hp4395 TO 717  ! When iBASIC is used, replace "717" to "800"
60     OUTPUT @Hp4395;"CLES;*SRE 4;ESNB 96"
70     ON INTR 7 GOTO Done   ! \ When iBASIC is used,
80     ENABLE INTR 7;2       ! / replace "7" to "8"
90     OUTPUT @Hp4395;"STOP?"
100    ENTER @Hp4395;Fstop
110    OUTPUT @Hp4395;"PRSMKRS"
120    OUTPUT @Hp4395;"MKR ON;SEAM PEAK"
130    OUTPUT @Hp4395;"OUTPMKR?"
140    ENTER @Hp4395;Vf,Vf2,Ff    ! Fundamental
150    Vf=SQR(10^(Vf/10)*.05)     ! Vf in V
160    PRINT "Fundamental"
170    Fr=Ff
180    I=2
190    S=0
200   LOOP
210    Fh=Ff*I
220   EXIT IF Fstop<=Fh
230    OUTPUT @Hp4395;"DMKR TRAC;MKRPRM ";Fh-Fr/2
240    OUTPUT @Hp4395;"DMKR ON"
250    OUTPUT @Hp4395;"MKRPRM ";Fr
260    OUTPUT @Hp4395;"PARS ON;SEARSTR"
270    OUTPUT @Hp4395;"SEAM PEAK;DMKR OFF"
280    OUTPUT @Hp4395;"OUTPMKR?"
290    ENTER @Hp4395;Vh,Vh2,F
300    Vh=10^(Vh/10)*.05          ! Vh^2 in V^2
310    PRINT I;" harmonic"
320    S=S+Vh
330    I=I+1
340   END LOOP
350  !
360 Done:   !
370   Thd=SQR(S)/Vf*100
380   PRINT "THD=";Thd;" %"
390   DISP "PROGRAM FINISHED"
400   END
```

Figure 8-14. Sample Program : Total Harmonic Distortion (THD)

In line 120 the marker searches for the fundamental frequency.

In lines 200 to 340 the marker searches for the harmonics on the analyzer display and integrates the squares.

Line 370 calculates the THD and line 380 prints the result.

To Obtain an Integral of a Power

This section provides the sample program which calculates the integral of the power within a specified frequency range. In this program, the frequency range is specified with a center (carrier) frequency and span. Before running this program, measure the frequency characteristic of a signal to obtain the frequency range over which you want to integrate. Then specify the range in the program.

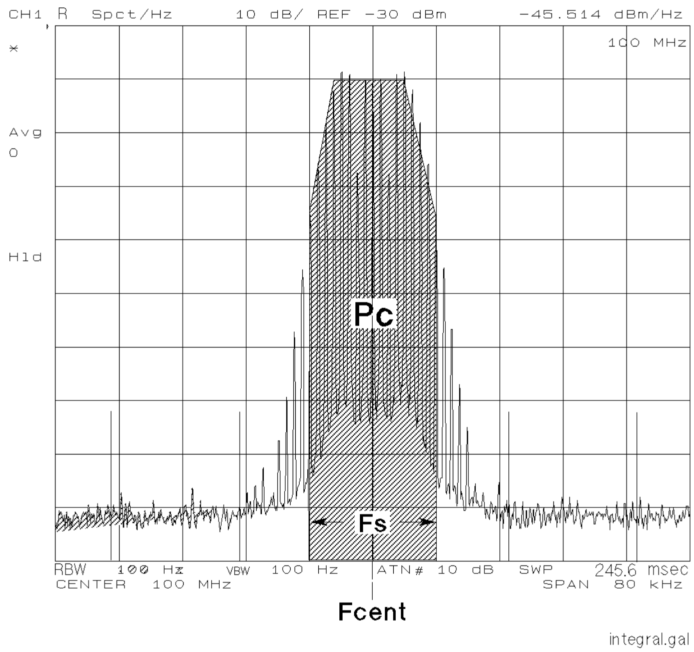


Figure 8-15. Integral Calculation of a Power

```
10 !
20 !Fig.8-16  Power Integral Calculation
30 !
40    ASSIGN @Hp4395 TO 717   ! When iBASIC is used, change "717" to "800"
50    Fs=16000                ! Target band width, Hz
60    Rbw=100                 ! Resolution band width, Hz
70    Fspan=80000             ! Frequency Span, Hz
80    Avg=10                  ! Averaging factor
90    !
100   CLEAR SCREEN
110   INPUT "Enter carrier frequency(Hz).",Fcent
120   OUTPUT @Hp4395;"CENT ";Fcent
130   OUTPUT @Hp4395;"SPAN ";Fspan
140   OUTPUT @Hp4395;"BW ";Rbw
150   OUTPUT @Hp4395;"AVERFACT ";Avg
160   OUTPUT @Hp4395;"FMT NOISE;SAUNIT DBM;ATTAUTO ON;AVER ON"
170   OUTPUT @Hp4395;"HOLD;AVERREST"
180   !
190     INPUT "Connect input port and press Enter.",Dum$
200     DISP "MEASURING"
210     OUTPUT @Hp4395;"TRGS INT"
220     OUTPUT @Hp4395;"CLES"
230     OUTPUT @Hp4395;"*SRE 4;ESNB 1"
240     ON INTR 7 GOTO Sweep_end       ! \ When iBASIC is used,
250     ENABLE INTR 7;2                ! / change "7" to "8"
260       OUTPUT @Hp4395;"NUMG ";Avg
270 Measuring: GOTO Measuring
```

Figure 8-16. Sample Program : Integral Calculation of a Power (1/2)

```
280 Sweep_end:     !
290     DISP "MEASUREMENT COMPLETE"
300     DIM D(1:801)
310     OUTPUT @Hp4395;"FORM3"
320     ASSIGN @Dt TO 717;FORMAT OFF   ! When iBASIC is used,
330     OUTPUT @Hp4395;"OUTPDTRC?"     ! change "717" to "800"
340     ENTER @Dt USING "%,8A";Dum$
350     ENTER @Dt;D(*)
360     ENTER @Dt USING "%,1A";Dum$
370     !
380   Nop=801
390   OUTPUT @Hp4395;"POIN?"
400   ENTER @Hp4395;Nop
410   Pc=FNPower(D(*),Fspan,Fs,Nop,Fcent)
420   !
430   PRINT "Carrier (MHz):",Fcent/1.E+6
440   PRINT "Power   (dBm):",Pc
450   PRINT
460   DISP "PROGRAM FINISHED"
470   END
480   !
490   DEF FNPower(D(*),Fspan,Fs,Nop,Fcent)
500     Fdelta=Fspan/(Nop-1)
510     Ich0=401
520     Ich=PROUND(Ich0,0)
530     I1=Ich-PROUND(Fs/2/Fdelta,0)
540     I2=Ich+PROUND(Fs/2/Fdelta,0)
550     IF I1<1 OR I2>Nop THEN
560       P=0
570       BEEP
580       PRINT "=============================="
590       PRINT "Frequency Range is Invalid!"
600       PRINT "=============================="
610       RETURN P
620     END IF
630     S=0
640     FOR I=I1 TO I2
650       S=S+10^(D(I)/10)         ! Change to Linear
660     NEXT I
670     P=S*Fdelta
680     P=10*LGT(P)                ! Change to log
690     RETURN P
700   FNEND
```

Sample Program : Integral Calculation of a Power (2/2)

Lines 50 to 80 specify typical values to the measurement factor (Fs), frequency span, resolution bandwidth, and averaging factor.

Lines 490 to 700 are the subprogram `FNPower` which accumulates a power at each measurement point within the range specified with the center frequency `Fch` and the frequency span `Fs`. The total power is obtained by the following equations:

$$PWR = 10\log_{10} P_1\,[\mathrm{dBm}] \qquad (8-2-a)$$

$$P_1 = \sum_{x=\mathrm{I1}}^{x=\mathrm{I2}} \frac{D1(x)Fspan}{Nop}\,[\mathrm{mW}] \qquad (8-2-b)$$

$$D1(x) = 10^{(D(x)/10)} \times 0.05\,[\mathrm{mW/Hz}] \qquad (8-2-c)$$

Where,

$D(x)$	Power density spectrum at each measurement point [dBm/Hz]
$Fspan$	Measurement frequency span [Hz]
Nop	Number of measurement points
I1, I2	Measurement points at the left/right end of channel bandwidth (Fs)

When calculating a total power, each measurement value should be converted from logarithmic (dBm) to linear [mW]. They should then be added together and re-converted into decibels [dBm], as shown in the equations (8-2). This program simplifies this procedure by converting values to linear, shown in the line 650 (note that it is not converted to [mW] unit), then re-converting into [dBm] shown in the line 680.

To Obtain Adjacent Channel Power

The adjacent channel power measurement examines the leakage power transmitted into an adjacent channel (that is, the channel next to the carrier channel). This program calculates the ratio of the adjacent channel power leakage to the power of the transmitter (Pl−Pc, Ph−Pc) in dBc.

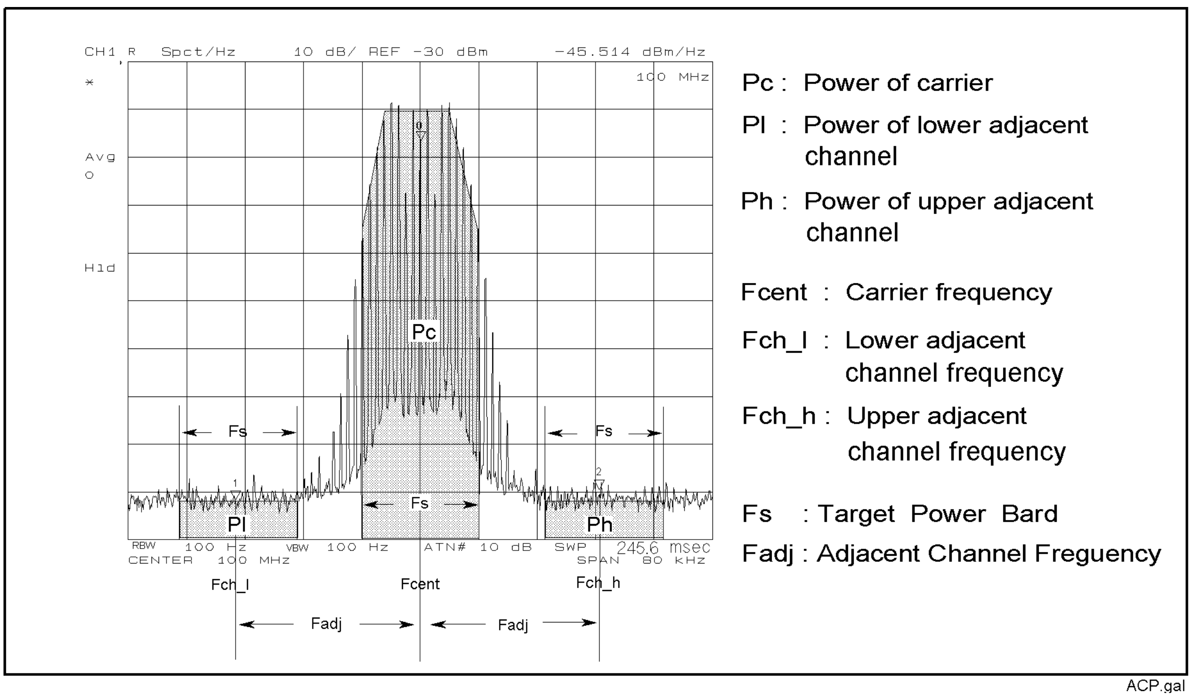


Figure 8-17. Adjacent Channel Power

Before running the program, set up the measurement and connect the signal to the input port.

```
10 !
20 !Fig.8-18  Adjacent Channel Power Calculation
30 !
40    ASSIGN @Hp4395 TO 717   ! When iBASIC is used, change "717" to "800"
50    Fadj=25000              ! Channel spacing, Hz
60    Fs=16000                ! Target band width, Hz
70    Rbw=100                 ! Resolution band width, Hz
80    Nop=801                 ! Number of measurement points
90    Fspan=80000             ! Frequency Span, Hz
100   Avg=10                  ! Averaging factor
110   !
120   CLEAR SCREEN
130   INPUT "Enter carrier frequency(Hz).",Fcent
140   OUTPUT @Hp4395;"CENT ";Fcent
150   OUTPUT @Hp4395;"SPAN ";Fspan
160   OUTPUT @Hp4395;"BW ";Rbw
170   OUTPUT @Hp4395;"AVERFACT ";Avg
180   OUTPUT @Hp4395;"FMT NOISE;SAUNIT DBM;ATTAUTO ON;AVER ON"
190   OUTPUT @Hp4395;"HOLD;AVERREST"
200   !
210     INPUT "Connect input port and press Enter.",Dum$
220     DISP "MEASURING"
230     OUTPUT @Hp4395;"TRGS INT"
240     OUTPUT @Hp4395;"CLES"
250     OUTPUT @Hp4395;"*SRE 4;ESNB 1"
260     ON INTR 7 GOTO Sweep_end       ! \ When iBASIC is used,
270     ENABLE INTR 7;2                ! / change "7" to "8"
280       OUTPUT @Hp4395;"NUMG ";Avg
290 Measuring:GOTO Measuring
300 Sweep_end:     !
310     DISP "MEASUREMENT COMPLETE"
320     DIM D(1:801)
330     OUTPUT @Hp4395;"FORM3"
340     ASSIGN @Dt TO 717;FORMAT OFF   ! When iBASIC is used,
350     OUTPUT @Hp4395;"OUTPDTRC?"     ! change "717" to "800"
360     ENTER @Dt USING "%,8A";Dum$
370     ENTER @Dt;D(*)
380     ENTER @Dt USING "%,1A";Dum$
390     !
400   Fch_l=Fcent-Fadj
410   Fch_h=Fcent+Fadj
420   Pc=FNPower(D(*),Fspan,Fcent,Fs,Nop,Fcent)
430   Pl=FNPower(D(*),Fspan,Fch_l,Fs,Nop,Fcent)
440   Ph=FNPower(D(*),Fspan,Fch_h,Fs,Nop,Fcent)
450   !
460   OUTPUT @Hp4395;"MKR ON;SMKR1 ON;SMKR2 ON"
470   OUTPUT @Hp4395;"MKRPRM ";Fcent
480   OUTPUT @Hp4395;"SMKRPRM1 ";Fch_l
490   OUTPUT @Hp4395;"SMKRPRM2 ";Fch_h
500   PRINT "Carrier (MHz):",Fcent/1.E+6
510   PRINT "Power   (dBm):",Pc
520   PRINT
530   PRINT "Adjacent Channel Freq. Lo(Hz):",Fch_l
540   PRINT "                       Hi(Hz):",Fch_h
550   PRINT
560   PRINT "Adjacent Pow. Pl-Pc(dBc):",Pl-Pc
570   PRINT "              Ph-Pc(dBc):",Ph-Pc
580   DISP "PROGRAM FINISHED"
590   END
600   !
```

Figure 8-18. Sample Program : Adjacent Channel Power Calculation (1/2)

```
610   DEF FNPower(D(*),Fspan,Fch,Fs,Nop,Fcent)
620     Fdelta=Fspan/(Nop-1)
630     Ich0=(Fch-Fcent)/Fdelta+401
631     Ich=PROUND(Ich0,0)
640     I1=Ich-PROUND(Fs/2/Fdelta,0)
650     I2=Ich+PROUND(Fs/2/Fdelta,0)
660     IF I1<1 OR I2>Nop THEN
670       P=0
680       RETURN P
690     END IF
700     S=0
710     FOR I=I1 TO I2
720       S=S+10^(D(I)/10)
730     NEXT I
740     P=S*Fdelta
750     P=10*LGT(P)
760     RETURN P
770   FNEND
```

Sample Program : Adjacent Channel Power Calculation (2/2)

Lines 50 to 100 set the the following variables to typical values: measurement coefficient, Fadj, Fs, frequency span, resolution bandwidth, number of measurement points, and averaging factor.

Line 180 sets the display format to noise format.

In lines 610 to 760 (the subprogram `FNPower`) performs a summation of the power at the measurement points, in the area specified by the center frequency `Fch`, and the frequency span `Fs`. This program calculates based on the same equations as in "To Obtain an Integral of a Power" (See Equations (8-2)).

To Obtain Occupied Power Bandwidth

This program calculates the occupied power bandwidth of the carrier signal. It first computes the combined power of all the signal responses contained in the trace. It then calculates the point for which 0.5 % of the total power lies to the right of the right maker and to the left of the left marker. The frequency difference between these two points is the bandwidth.

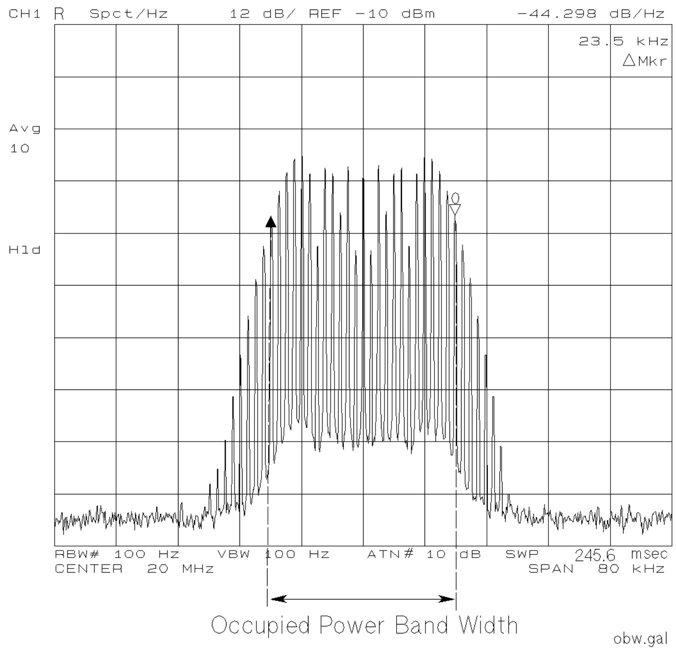


Figure 8-19. 99 % Occupied Power Bandwidth

Before running the program, set up the measurement and connect the signal to the input port.

```
10 !
20 !Fig.8-20  Occupied Power Band Width Calculation
30 !
40    ASSIGN @Hp4395 TO 717 ! When iBASIC is used, change "717" to "800"
50    Rbw=100         ! Resolution band width, Hz
60    Nop=801         ! Number of measurement points
70    Fspan=80000.    ! Frequency Span, Hz
80    Avg=10          ! Averaging factor
90    !
100   INPUT "Enter carrier frequency(Hz).",Fcent
110   OUTPUT @Hp4395;"SPAN";Fspan
120   OUTPUT @Hp4395;"CENT";Fcent
130   OUTPUT @Hp4395;"BW";Rbw
140   OUTPUT @Hp4395;"FMT NOISE;DET POS"
150   OUTPUT @Hp4395;"SAUNIT DBM;ATTAUTO ON;AVER ON"
160   OUTPUT @Hp4395;"AVERFACT";Avg
170   !
180   INPUT "Connect input port and press Enter.",Dum$
190   OUTPUT @Hp4395;"HOLD"
200   OUTPUT @Hp4395;"CLES"
210   OUTPUT @Hp4395;"*SRE 4;ESNB 1"
220   ON INTR 7 GOTO Sweep_end     ! \ When iBASIC is used,
230   ENABLE INTR 7;2              ! / change "7" to "8"
240     OUTPUT @Hp4395;"TRGS INT"
250     OUTPUT @Hp4395;"NUMG";Avg
260 Measuring:GOTO Measuring
270 Sweep_end:  !  Get Data
280   DIM D(1:801)
290   DIM P(1:801)
```

Figure 8-20. Sample Program : Occupied Power Bandwidth Calculation (1/2)

```
300   OUTPUT @Hp4395;"FORM3"
310   ASSIGN @Dt TO 717;FORMAT OFF ! When iBASIC is used,
320   OUTPUT @Hp4395;"OUTPDTRC?"   ! change "717" to "800"
330   ENTER @Dt USING "%,8A";Dum$
340   ENTER @Dt;D(*)
350   ENTER @Dt USING "%,1A";Dum$
360   !
370   Power(D(*),P(*),Rbw,Nop,Fspan)
380   !
390   FOR I=1 TO Nop
400     A=P(I)/P(Nop)
410     IF A>.005 THEN Lower
420   NEXT I
430 Lower:I1=I
440   FOR I=Nop TO 1 STEP -1
450     A=P(I)/P(Nop)
460     IF A<.995 THEN Upper
470   NEXT I
480 Upper:I2=I
490     OUTPUT @Hp4395;"MKR ON"
500     OUTPUT @Hp4395;"MKRP ";I1
510     OUTPUT @Hp4395;"DMKR ON"
520     OUTPUT @Hp4395;"MKRP ";I2
530     OUTPUT @Hp4395;"OUTPMKR?"
540     ENTER @Hp4395;Val,Val2,Flh
550     PRINT "Occupied band width :";
560     PRINT Flh;" Hz"
570   DISP "PROGRAM FINISHED"
580   END
590   !
600   SUB Power(D(*),P(*),Rbw,Nop,Fspan)
610     S=0
620     FOR I=1 TO Nop
630       S=S+10^(D(I)/10)
640       P(I)=S
650     NEXT I
660   SUBEND
```

Sample Program : Occupied Power Bandwidth Calculation (2/2)

Lines 40 to 80 set the following variables to typical values: measurement coefficient, frequency span, resolution bandwidth, number of measurement points, and averaging factor.

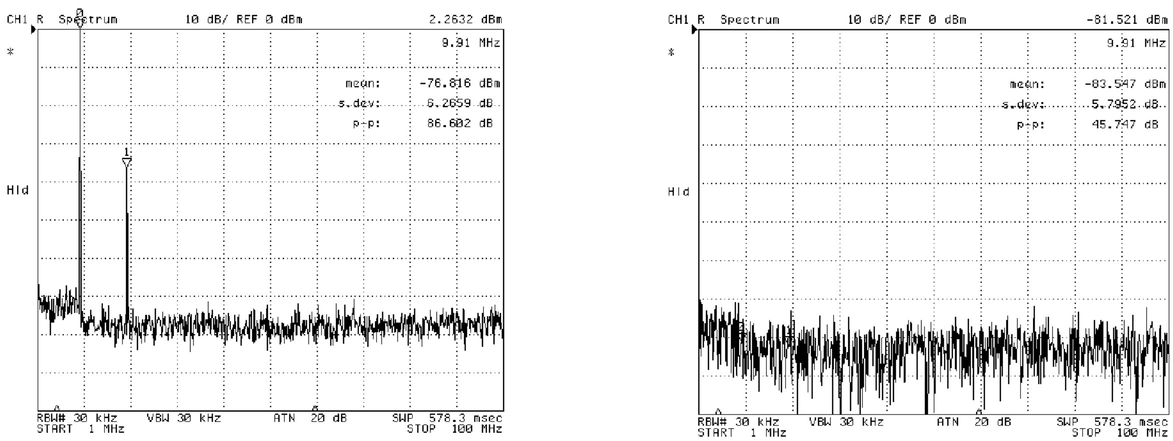
Lines 390 to 430 search from the left for the point where the power is 0.5 % compared to the total power. Lines 440 to 480 do the same search from the right.

Lines 490 to 540 display the marker and Δ marker on the 0.5 % power points and read out the spacing of the markers.

Lines 600 to 660 (subprogram `FNPower`) perform a summation of the power at the measurement points. This summation is done in the area specified by the center frequency (`Fch`) and the frequency span (`Fs`). The same equation (11-2) is used in the “To Obtain Adjacent Channel Power” example.

To Calculate an S/N Ratio

The sample program in this section calculates a signal-to-noise (S/N) ratio. This program measures the power of a signal component (a). It then calculates the power of a noise component (b) defined as the measurement data subtracted IF BW data above and below the measurement signal (b).



a) Signal b) Noise

Figure 8-21. Calculating an S/N ratio

The measurement port should be setup before execution the program.

```
10   !Fig.8-22  Sample program to measure S/N ratio
20   !
30   DIM Dat(1:801)
40   !
50   Fstar=1.E+6  ! Start Frequency
60   Fstop=1.E+8  ! Stop  Frequency
70   Rbw=30000.   ! RBW
80   Rstar=5.E+6  ! Start of Noise BW
90   Rstop=6.E+7  ! Stop  of Noise BW
100  !
110  ASSIGN @Hp4395 TO 717 ! When iBASIC is used, change "717" to "800"
120  ASSIGN @Dt TO 717;FORMAT OFF ! When iBASIC is used, change "717" to "800"
130  !
140  GOSUB Setup
150  GOSUB Measure_signal
160  GOSUB Measure_noise
170  GOSUB Printing
180  STOP
190  !
200 Setup:  !
210  OUTPUT @Hp4395;"HOLD"
220  OUTPUT @Hp4395;"FORM3"
230  OUTPUT @Hp4395;"DISA HIHB"
240  PRINT "SNR Sample Program"
250  !
```

Figure 8-22. Sample Program : Calculating an S/N ratio (1/3)

```
260  OUTPUT @Hp4395;"SA"
270  OUTPUT @Hp4395;"STAR ";Fstar
280  OUTPUT @Hp4395;"STOP ";Fstop
290  OUTPUT @Hp4395;"BW ";Rbw
300  OUTPUT @Hp4395;"MKR ON;MKRUNIT DBM"
310  OUTPUT @Hp4395;"PARS ON;MEASTAT ON"
320  OUTPUT @Hp4395;"MKRPRM ";Rstar
330  OUTPUT @Hp4395;"SEARSTRL"
340  OUTPUT @Hp4395;"MKRPRM ";Rstop
350  OUTPUT @Hp4395;"SEARSTRR"
360 RETURN
370  !
380 Measure_signal:!
390  OUTPUT @Hp4395;"POIN?"
400  ENTER @Hp4395;Poin
410  Nrbw=INT(Rbw/((Fstop-Fstar)/(Poin-1)))+1
420  !
430  DISP "measuring signal power..."
440  OUTPUT @Hp4395;"FMT SPECT"
450  OUTPUT @Hp4395;"SING;*OPC?"
460  ENTER @Hp4395;Dummy
470  !
480  OUTPUT @Hp4395;"SEAM PKSA"
490  OUTPUT @Hp4395;"MKRVAL?"
500  ENTER @Hp4395;Psignal
510  OUTPUT @Hp4395;"MKRPRM?"
520  ENTER @Hp4395;Fsignal
530  OUTPUT @Hp4395;"MKR?"
540  ENTER @Hp4395;Mkr(0)
550  OUTPUT @Hp4395;"MKRP?"
560  ENTER @Hp4395;Mkrp(0)
570  FOR I=1 TO 7
580      OUTPUT @Hp4395;"SMKR"&VAL$(I)&"?"
590      ENTER @Hp4395;Mkr(I)
600      OUTPUT @Hp4395;"SMKRP"&VAL$(I)&"?"
610      ENTER @Hp4395;Mkrp(I)
620  NEXT I
630  OUTPUT @Hp4395;"DATMEM;DISP DATM"
640 RETURN
650  !
660 Measure_noise:!
670  DISP "measuring noise power..."
680  OUTPUT @Hp4395;"FMT NOISE"
690  OUTPUT @Hp4395;"SING;*OPC?"
700  ENTER @Hp4395;Dummy
710  !
720  OUTPUT @Hp4395;"OUTPDATA?"
730  ENTER @Hp4395 USING "8A,#";Head$
740  ENTER @Dt;Dat(*)
750  ENTER @Hp4395 USING "1A,#";Foot$
760  !
770  FOR I=0 TO 7
780      IF Mkr(I) THEN
790          FOR J=-Nrbw TO Nrbw
800              Dat(Mkrp(I)+J)=Dat(Mkrp(I)-Nrbw-1)
810         NEXT J
820      END IF
830  NEXT I
840  !
850  OUTPUT @Hp4395;"INPUDATA #46408";
860  OUTPUT @Dt;Dat(*),END
```

Sample Program : Calculating an S/N ratio (2/3)

```
870  !
880  OUTPUT @Hp4395;"MKRUNIT W"
890  OUTPUT @Hp4395;"OUTPMSTA?"
900  ENTER @Hp4395;Mean,Sdev,Pp
910  !
920  Pnoise=10*LGT(Mean*(Rstop-Rstar)*1000)
930  !
940  DISP "measuring done."
950  RETURN
960  !
970 Printing: !
980  PRINT
990  PRINT USING "K,MDDD.DDD,K,#";"  Signal Power : ",Psignal,"dBm"
1000  PRINT "   @ ";FNEngr$(Fsignal,5);"Hz"
1010 PRINT USING "K,MDDD.DDD,K,#";"  Noise  Power : ",Pnoise,"dBm"
1020 PRINT "   @ ";FNEngr$(Rstar,5);" to ";FNEngr$(Rstop,5);"Hz"
1030 PRINT USING "K,MDDD.DDD,K";"  SNR          : ",Psignal-Pnoise,"dB"
1040 !
1050 OUTPUT @Hp4395;"FMT SPECT"
1060 OUTPUT @Hp4395;"MKRUNIT DBM"
1070 RETURN
1080 !
1090 END
1100 !
1110 DEF FNEngr$(X,Round)
1120     Unit$="afpnum kMGTP"
1130     SELECT ABS(X)
1140     CASE >=1.E+18
1150         RETURN "*****"
1160     CASE <1.E-18
1170         RETURN "0"
1180     CASE ELSE
1190         Unit=INT(LGT(DROUND(ABS(X),Round))/3)
1200         RETURN VAL$(DROUND(X/10^(Unit*3),Round))&Unit$[Unit+7;1]
1210     END SELECT
1220 FNEND
```

Sample Program : Calculating an S/N ratio (3/3)

Lines 200 to 360 prepare for measurement.

Lines 380 to 640 measure signal.

Lines 480 to 560 use the marker function to search and measure the amplitude of and frequency of the signal where the marker is located.

Line 630 copies the measurement data of a signal from the data trace to the memory trace and displays both of the trace.

Lines 660 to 950 subtract the signal component from measurement data to the obtain noise component.

Line 680 set the display format to the noise format.

Lines 720 to 830 retrieve data, then subtract signal component.

Lines 1110 to 1220 are the subprogram which modifies the display format of the stop frequency.

Programs for the Impedance Analyzer Mode

To Perform Calibration

This section provides the sample program which performs calibration and compensation in the impedance analyzer mode. This program should run with 43961 impedance test kit and a test fixture appropriate for your DUT. See *Operation Manual* for detailed information on test fixtures.

```
10  !
20  !Fig.8-23  ZA Calibration
30  !
40   CLEAR SCREEN
50   ASSIGN @Hp4395 TO 717      ! When iBASIC is used, replace "717" to "800"
60   OUTPUT @Hp4395;"PRES"
70   OUTPUT @Hp4395;"ZA"
80   BEEP
90   DISP "Connect 43961A, THEN press continue"
100  PAUSE
110 !
120 Calibration:      !
130    OUTPUT @Hp4395;"CALK APC7"
140    OUTPUT @Hp4395;"BW 300;AVERFACT 8;AVER ON"
150    !
160    BEEP
170    PRINT "CALIBRATION"
180    DISP "OPEN CAL, READY? press continue"
190    PAUSE
200    OUTPUT @Hp4395;"CALI IMP"
210    OUTPUT @Hp4395;"CLASIMPA"
220    OUTPUT @Hp4395;"*OPC?"
230    ENTER @Hp4395;Dummy
240    DISP "SHORT CAL, READY? press continue"
250    PAUSE
260    OUTPUT @Hp4395;"CLASIMPB"
270    OUTPUT @Hp4395;"*OPC?"
280    ENTER @Hp4395;Dummy
290    DISP "LOAD CAL, READY? press continue"
300    PAUSE
310    OUTPUT @Hp4395;"CLASIMPC"
320    OUTPUT @Hp4395;"*WAI"
330   OUTPUT @Hp4395;"SAVIMP"
340   OUTPUT @Hp4395;"*OPC?"
350   ENTER @Hp4395;Dummy
360   !
370   DISP "Connect a fixture and press continue"
380   PAUSE
390   !
400   PRINT "SELECT FIXTURE"
410   PRINT "1 : 16191A"
420   PRINT "2 : 16192A"
430   PRINT "3 : 16193A"
440   PRINT "4 : 16194A"
450 !
460   INPUT " Input number of fixture",Fixt
470 !
480  SELECT Fixt
490    CASE 1
```

Figure 8-23. Sample Program : Calibration (1/2)

```
500      OUTPUT @Hp4395;"FIXT HP16191"
510     CASE 2
520      OUTPUT @Hp4395;"FIXT HP16192"
530     CASE 3
540      OUTPUT @Hp4395;"FIXT HP16193"
550     CASE 4
560      OUTPUT @Hp4395;"FIXT HP16194"
570    END SELECT
580 !
590 !
600    PRINT "COMPENSATION"
610    DISP "OPEN COMPEN. READY? press continue"
620      PAUSE
630    OUTPUT @Hp4395;"COMP;COMCA"
640    OUTPUT @Hp4395;"*OPC?"
650    ENTER @Hp4395;Dummy
660    DISP "SHORT COMPEN. READY? press continue"
670      PAUSE
680    OUTPUT @Hp4395;"COMCB"
690    OUTPUT @Hp4395;"*WAI"
700    OUTPUT @Hp4395;"SAVCOM"
710    OUTPUT @Hp4395;"*OPC?"
720    ENTER @Hp4395;Dummy
730    !
740    DISP "PROGRAM END"
750    END
```

Sample Program : Calibration (2/2)

In the impedance analyzer mode, calibration should be performed with the IF BW set to 300 Hz or less and the averaging factor greater than 7.

Line 200 starts the calibration. If execution comes to this line before the end of calibration, the program resets the calibration procedure.

Lines 210 to 350 perform calibration. Calibration should be done in synchronization with the controller. This program uses `*OPC?` for this purpose.

Lines 400 to 570 calculates the compensation for the test fixture you use, in terms of electrical length.

Lines 600 to 720 perform compensation.

To Measure Capacitance and the factor D

The sample program in this section measures capacitance and the factor D. The program also calibrates your 4395A.

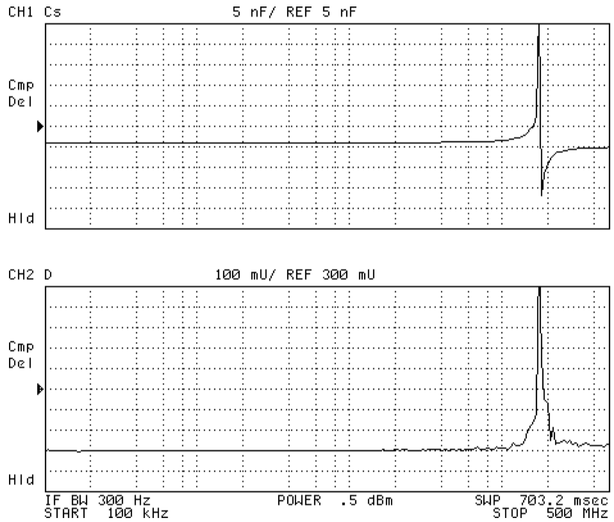


Figure 8-24. C-D Measurement

```
10  !
20  !Fig.8-25  PROGRAM TO MEASURE C-D
30  !
40   CLEAR SCREEN
50   ASSIGN @Hp4395 TO 717      ! When iBASIC is used replace "717" to "800"
60   OUTPUT @Hp4395;"PRES"
70   OUTPUT @Hp4395;"ZA"
80   OUTPUT @Hp4395;"*OPC?"
90   ENTER @Hp4395;Dummy
100  !
110  ! Interrupt Operation Set-up
120   OUTPUT @Hp4395;"CLES"
130   OUTPUT @Hp4395;"*SRE 32;*ESE 36"
140   ON INTR 7 GOSUB Err_report    ! \ When iBASIC is used,
150   ENABLE INTR 7;2               ! / change "7" to "8"
160  !
170  BEEP
180  DISP "Connect 43961A, THEN press continue"
190  PAUSE
200  !
210  GOSUB Setup
220  GOSUB Calibration
230  GOSUB Measure
240  GOTO Ending
250 !
260 !
270 Setup:!
280 ! Measurement Setup
290   OUTPUT @Hp4395;"SWPT LOGF"
300   OUTPUT @Hp4395;"STAR 1E6"
310   OUTPUT @Hp4395;"STOP 500E6"
320   OUTPUT @Hp4395;"CHAN1"
330   OUTPUT @Hp4395;"MEAS CP"
```

Figure 8-25. Sample Program : C-D Measurement (1/4)

```
340   OUTPUT @Hp4395;"CHAN2"
350   OUTPUT @Hp4395;"MEAS D"
360   OUTPUT @Hp4395;"DUAC ON"
370   OUTPUT @Hp4395;"COUC ON"
380   OUTPUT @Hp4395;"MKRCOUP ON"
390   OUTPUT @Hp4395;"CHAN1"
400   OUTPUT @Hp4395;"*OPC?"
410   ENTER @Hp4395;Dummy
420  RETURN
430 !
440 Calibration:      !
450    BEEP
460    PRINT "CALIBRATION"
470    OUTPUT @Hp4395;"CALK APC7"
480    OUTPUT @Hp4395;"BW 300;AVERFACT 8;AVER ON"
490    DISP "OPEN CAL, READY? press continue"
500    PAUSE
510    OUTPUT @Hp4395;"CALI IMP"
520    OUTPUT @Hp4395;"CLASIMPA"
530    OUTPUT @Hp4395;"*OPC?"
540    ENTER @Hp4395;Dummy
550    DISP "SHORT CAL, READY? press continue"
560    PAUSE
570    OUTPUT @Hp4395;"CLASIMPB"
580    OUTPUT @Hp4395;"*OPC?"
590    ENTER @Hp4395;Dummy
600    DISP "LOAD CAL, READY? press continue"
610    PAUSE
620    OUTPUT @Hp4395;"CLASIMPC"
630    OUTPUT @Hp4395;"*WAI"
640   OUTPUT @Hp4395;"SAVIMP"
650   OUTPUT @Hp4395;"*OPC?"
660   ENTER @Hp4395;Dummy
670 !
671   OUTPUT @Hp4395;"USKEY"
680   DISP "Connect a fixture, then press softkey"
690  ON KEY 1 LABEL "16191A" GOTO Fixt1
700  ON KEY 2 LABEL "16192A" GOTO Fixt2
710  ON KEY 3 LABEL "16193A" GOTO Fixt3
720  ON KEY 4 LABEL "16194A" GOTO Fixt4
730  ON KEY 5 LABEL "" GOTO 680
740  ON KEY 6 LABEL "" GOTO 680
750  ON KEY 7 LABEL "" GOTO 680
760  ON KEY 8 LABEL "ABORT" GOTO Fix_end
770  GOTO 770
780 !
790 Fixt1:!
800    OUTPUT @Hp4395;"FIXT HP16191"
810    GOTO Fix_end
820 Fixt2:!
830    OUTPUT @Hp4395;"FIXT HP16192"
840    GOTO Fix_end
850 Fixt3:!
860    OUTPUT @Hp4395;"FIXT HP16193"
870    GOTO Fix_end
880 Fixt4:!
890    OUTPUT @Hp4395;"FIXT HP16194"
900    GOTO Fix_end
910 Fix_end: !
920  OFF KEY
```

Sample Program : C-D Measurement (2/4)

```
930 !
940  DISP "OPEN COMPEN. READY? press continue"
950    PAUSE
960  OUTPUT @Hp4395;"COMP;COMCA"
970  OUTPUT @Hp4395;"*OPC?"
980  ENTER @Hp4395;Dummy
990  DISP "SHORT COMPEN. READY? press continue"
1000   PAUSE
1010 OUTPUT @Hp4395;"COMCB"
1020 OUTPUT @Hp4395;"*WAI"
1030 OUTPUT @Hp4395;"SAVCOM"
1040 OUTPUT @Hp4395;"*OPC?"
1050 ENTER @Hp4395;Dummy
1060 !
1070 RETURN
1080 !
1090 Measure:!
1100 DIM Res(5,2),Freq(5)
1110 DISP "Connect DUT, then press continue"
1120 PAUSE
1130 DISP "Measuring"
1140 !
1150 OUTPUT @Hp4395;"MKR ON"
1160 OUTPUT @Hp4395;"SMKR1 ON;SMKRPRM1 10E6"
1170 OUTPUT @Hp4395;"SMKR2 ON;SMKRPRM2 100E6"
1180 OUTPUT @Hp4395;"SMKR3 ON;SMKRPRM3 200E6"
1190 OUTPUT @Hp4395;"SMKR4 ON;SMKRPRM4 300E6"
1200 OUTPUT @Hp4395;"SMKR5 ON;SMKRPRM5 500E6"
1210 OUTPUT @Hp4395;"MKRPRM 1E6"
1220 !
1230 FOR I=1 TO 2
1240 OUTPUT @Hp4395;"CHAN";CHR$(I+48)
1250 OUTPUT @Hp4395;"*WAI"
1260 OUTPUT @Hp4395;"SING"
1270 OUTPUT @Hp4395;"*WAI"
1280 OUTPUT @Hp4395;"AUTO"
1290    OUTPUT @Hp4395;"MKRVAL?"
1300    ENTER @Hp4395;Res(0,I)
1310    OUTPUT @Hp4395;"MKRPRM?"
1320    ENTER @Hp4395;Freq(0)
1330   FOR J=1 TO 5
1340    OUTPUT @Hp4395;"SMKRVAL";CHR$(J+48);"?"
1350    ENTER @Hp4395;Res(J,I)
1360    OUTPUT @Hp4395;"SMKRPRM";CHR$(J+48);"?"
1370    ENTER @Hp4395;Freq(J)
1380   NEXT J
1390 NEXT I
1400 !
1410 !PRINTING
1420 PRINT ""
1430 PRINT "RESULTS"
1440 PRINT "Freq.              Capacitance(F)              D"
1450 PRINT "=================================================="
1460 FOR I=0 TO 5
1470 PRINT Freq(I),"MHz",Res(I,1),Res(I,2)
1480 NEXT I
1490 DISP "Measurement complete"
1500 !
1510 RETURN
1520 !
1530 Err_report: !
```

Sample Program : C-D Measurement (3/4)

```
1540   OUTPUT @Hp4395;"OUTPERRO"
1550   ENTER @Hp4395;Err,Err$
1560   BEEP
1570   PRINT "ERROR DETECTED!"
1580   PRINT Err,Err$
1590   !
1600   A=SPOLL(@Hp4395)
1610   OUTPUT @Hp4395;"*ESR?"
1620   ENTER @Hp4395;Estat
1630   ENABLE INTR 7                ! \ When iBASIC is used, change "7" to "8"
1640 RETURN
1650 !
1660 Ending:!
1670 DISP "Program Finish"
1680 END
```

Sample Program : C-D Measurement (4/4)

Lines 440 to 1070 perform calibration in the same way as in "To Perform Calibration", except electric length compensation which uses the ON KEY LABEL function.

Lines 1290 to 1370 show that data is read with the marker. Use the OUTPDTRC? command to retrieve all the measurement data.

Lines 1530 to 1630 set interruption to be generated when an GPIB error occurs. See Chapter 5 for information on interruption processing in detail.

To Measure a Varactor Diode Using DC Bias Sweep (With Option 010)

In the 4395A, the list sweep function is required in order to measure DUT characteristics as a function of DC bias voltage. The sample program in this section sweeps DC bias voltage across 15 measurement points with fixed frequencies.

A varactor diode is a device whose barrier capacitance varies depending on inverse bias voltage. It is used in many kinds of circuits including a VCO (Voltage Control Oscillator).

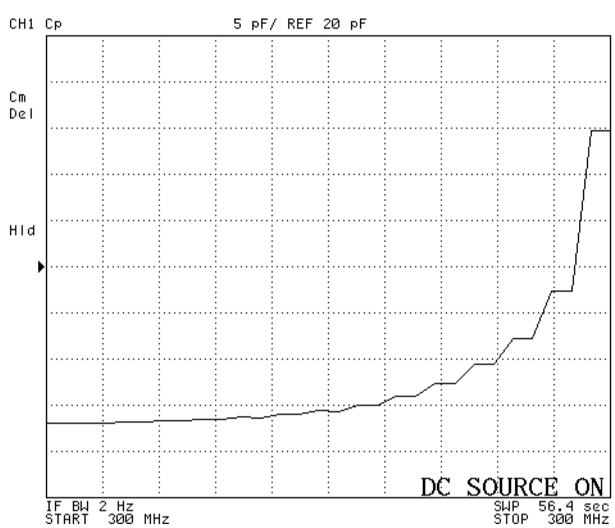


Figure 8-26. Characteristic of a Varactor Diode

```
10   !
20   !Fig.8-27  Varactor Diode MEASUREMENT
30   !
40   ASSIGN @Hp4395 TO 717  ! When iBASIC is used, change "717" to "800"
50   !
60   GOSUB Setup
70   GOSUB Setlist
80   GOSUB Calibration
90   GOSUB Measurement
100   !
110 Setup: !
120  OUTPUT @Hp4395;"PRES"
130  OUTPUT @Hp4395;"ZA"
140  OUTPUT @Hp4395;"CHAN1"
150  OUTPUT @Hp4395;"MEAS CP"
160  OUTPUT @Hp4395;"DCCTL VOLT"
170  RETURN
180  !
190 Setlist: !
200  DIM Voltfact(15)
210  OUTPUT @Hp4395;"AVER OFF"
220  OUTPUT @Hp4395;"LISDOBASE"
230    Freq=3.E+8
240  INPUT "Enter frequency(Hz)",Freq
250  Dcvmin=-30
260  INPUT "ENTER DC sweep level : Lower(V)",Dcvmin
270  Dcvmax=-.1
280  INPUT "ENTER DC sweep level : Upper(V)",Dcvmax
290    Pow=-13
300  INPUT "Enter power level(dBm)",Pow
310    Ifbw=2
320  INPUT "ENTER IFBW (Hz)",Ifbw
330    Curr=100
340  INPUT "ENTER DC bias current limit(mA)",Curr
350  Curr=Curr/1000
360  OUTPUT @Hp4395;"MAXDCI";Curr
370  !
380  Dcstep=(Dcvmax-Dcvmin)/14
390  !
400  OUTPUT @Hp4395;"CLEL"
410  OUTPUT @Hp4395;"EDITLIST"
420  FOR I=1 TO 15
430    Voltfact(I)=Dcvmin+Dcstep*(I-1)
440  NEXT I
450  FOR I=1 TO 15
460    OUTPUT @Hp4395;"SADD"
470    OUTPUT @Hp4395;"STAR ";Freq
480    OUTPUT @Hp4395;"STOP ";Freq
490    OUTPUT @Hp4395;"POIN 2"
500    OUTPUT @Hp4395;"POWE ";Pow
510    OUTPUT @Hp4395;"DCV ";Voltfact(I)
520    OUTPUT @Hp4395;"BW ";Ifbw
530    OUTPUT @Hp4395;"SDON"
540  NEXT I
550  OUTPUT @Hp4395;"EDITDONE"
560  OUTPUT @Hp4395;"*OPC?"
570  ENTER @Hp4395;Dummy
580  OUTPUT @Hp4395;"SWPT LIST"
590  !
600  RETURN
610  !
```

Figure 8-27. Sample Program : Measuring Varactor Diode Characteristic (1/3)

```
620 Calibration:      !
630    BEEP
640    OUTPUT @Hp4395;"AVERFACT 8;AVER ON"
650    PRINT "CALIBRATION"
660    DISP "OPEN CAL, READY? press continue"
670    PAUSE
680    OUTPUT @Hp4395;"CALI IMP"
690    OUTPUT @Hp4395;"CLASIMPA"
700    OUTPUT @Hp4395;"*OPC?"
710    ENTER @Hp4395;Dummy
720    DISP "SHORT CAL, READY? press continue"
730    PAUSE
740    OUTPUT @Hp4395;"CLASIMPB"
750    OUTPUT @Hp4395;"*OPC?"
760    ENTER @Hp4395;Dummy
770    DISP "LOAD CAL, READY? press continue"
780    PAUSE
790    OUTPUT @Hp4395;"CLASIMPC"
800    OUTPUT @Hp4395;"*WAI"
810   OUTPUT @Hp4395;"SAVIMP"
820   OUTPUT @Hp4395;"*OPC?"
830   ENTER @Hp4395;Dummy
840 !
850   OUTPUT @Hp4395;"KEY 47"
860   OUTPUT @Hp4395;"KEY 0"
870   OUTPUT @Hp4395;"KEY 6"
880   DISP "Connect a fixture, then press soft key"
890  ON KEY 1 LABEL "16191A" GOTO Fixt1
900  ON KEY 2 LABEL "16192A" GOTO Fixt2
910  ON KEY 3 LABEL "16193A" GOTO Fixt3
920  ON KEY 4 LABEL "16194A" GOTO Fixt4
930  ON KEY 5 LABEL "" GOTO 880
940  ON KEY 6 LABEL "" GOTO 880
950  ON KEY 7 LABEL "" GOTO 880
960  ON KEY 8 LABEL "ABORT" GOTO Fix_end
970  GOTO 970
980 !
990 Fixt1: !
1000    OUTPUT @Hp4395;"FIXT HP16191"
1010    GOTO Fix_end
1020 Fixt2:!
1030     OUTPUT @Hp4395;"FIXT HP16192"
1040    GOTO Fix_end
1050 Fixt3:!
1060    OUTPUT @Hp4395;"FIXT HP16193"
1070    GOTO Fix_end
1080 Fixt4:!
1090    OUTPUT @Hp4395;"FIXT HP16194"
1100    GOTO Fix_end
1110 Fix_end: !
1120   OFF KEY
1130 !
1140   BEEP
1150   DISP "OPEN COMPEN. READY? press continue"
1160     PAUSE
1170   OUTPUT @Hp4395;"COMP;COMCA"
1180   OUTPUT @Hp4395;"*OPC?"
1190   ENTER @Hp4395;Dummy
1200   DISP "SHORT COMPEN. READY? press continue"
1210     PAUSE
1220   OUTPUT @Hp4395;"COMCB"
```

Sample Program : Measuring Varactor Diode Characteristic (2/3)

```
1230   OUTPUT @Hp4395;"*WAI"
1240   OUTPUT @Hp4395;"SAVCOM"
1250   OUTPUT @Hp4395;"*OPC?"
1260   ENTER @Hp4395;Dummy
1270   !
1280   RETURN
1290   !
1300 Measurement:!
1310  INPUT "Connect DUT, and press Enter.",Dum$
1320  OUTPUT @Hp4395;"DCO ON"
1330  OUTPUT @Hp4395;"CLES"
1340  OUTPUT @Hp4395;"*SRE 4;ESNB 1"
1350  ON INTR 7 GOTO Sweep_end  ! \ When iBASIC is used, change "7" to "8"
1360  ENABLE INTR 7;2           ! /
1370    OUTPUT @Hp4395;"SING"
1380 Measuring: GOTO Measuring
1390  !
1400 Sweep_end: !
1410  DIM Dat(1:30,1:2)
1420  OUTPUT @Hp4395;"AUTO"
1430  OUTPUT @Hp4395;"OUTPDTRC?"
1440  ENTER @Hp4395;Dat(*)
1450    PRINT ""
1460    PRINT " Measurement Result"
1470    PRINT " Measurement Frequency        :";Freq;"Hz"
1480    PRINT " Measurement Power            :";Pow;"dB"
1490    PRINT " "
1500    PRINT " No.          DC(V)                Cp(F)"
1510    PRINT "======================================================"
1520  FOR I=1 TO 15
1530    PRINT I,Dcvmin+Dcstep*(I-1),Dat(I*2-1,1)
1540  NEXT I
1550  !
1560  END
```

Sample Program : Measuring Varactor Diode Characteristic (3/3)

Lines 450 to 570 input segments of a list sweep to 4395A.

Calibration, lines 620 to 1260, should be performed after to have inputed segments of a list sweep.

Lines 500 to 570 perform a sweep measurement. This program uses the status reporting system to work in synchronization with the end of sweep.

File Transfer Function

This section describes how to use the file transfer function, showing you a sample program.

The file transfer function uses the external controller to transfer files between the selected storage device of this instrument (memory disk or diskette) and an external storage device (such as hard disk). This function allows you to:

- Directly access data you want to use on the external controller.

  For example, you can transfer the files saved on the 4395A storage deveice, such as graphic files or data array (ASCII format) files, to the external controller. Then, you can output the graphic file transferred from a printer connected to the external controller, or manage the data array (ASCII format) trasferred using a software running on the external controller.

- Use external storage devices, which have larger capacity compared to the memory disk or a diskette.

  For example, if there are a great number of measurement conditions which require calibration, the amount of the setting data becomes extremely large, including calibration data. In this case, it is impractical to store all of these settings on the memory disk or a single diskette at a time. However, you can realize this functionality by transferring them to the external controller and then storing them on an external storage device.

- Perform remote measurement using the external controller with a few GPIB commands for basic measurement. You do not have to memorize further details (such as GPIB commands used for detailed settings).

  Preparation:

  Use the keys on the front panel to establish the setting required for your measurement. Store it on the storage device of the 4395A, then transfer the file to the external controller, and store it on an external storage device. Repeat this procedure for all of the settings required for your measurement.

  Measurement:

  Choose a necessary setting file from those stored and transfer it to the 4395A using the external controller. Then, recall the file to set the 4395A for the measurement and perform the measurement using the GPIB commands.

The storage device of the 4395A allows you to handle files listed below in the DOS format or the LIF format. For DOS format files, both binary files and ASCII files can be transferred. For LIF format files, only binary files can be transferred.

- Binary files
  - Instrument settings and internal data array (STATE)
  - Internal data arrays (DATA ONLY binary)
  - Graphic images (GRAPHICS)
- ASCII files
  - Internal data arrays (DATA ONLY ascii)
  - HP instrument BASIC programs

File Transfer from 4395A to External Controller

This program transfers a specified file in the current directory (memory disk in the sample program of Figure 8-28) of the 4395A to the current directory of the storage device (A drive in the sample program of Figure 8-28) connected to the external controller, giving a file name you desire.

When executed, this program first prompts you to enter a source file name, as shown below. Enter the name of a file you want to transfer. Note that the file name of the LIF format can be distinguished between capital and small letters.

```
ENTER SOURCE FILE NAME ON INSTRUMENT ?
```

Then, the program prompts you to enter a destination file name as shown below, and in this example, SAMPLE.STA (DOS format) or SAMPLE_S (LIF format) has been entered as the source file name. Enter the file name you want to give on the storage device. Note that a file with the same name will be overwritten, if it already exists.

```
ENTER SOURCE FILE NAME ON INSTRUMENT ?       SAMPLE.STA (SAMPLE_S)
ENTER DESTINATION FILE NAME ON CONTROLLER ?
```

```
10   !
20   !  Fig.8-28  File transfer (Instrument -> Controller)
30   !
40   DIM Src_file$[50],Dst_file$[50]
50   ASSIGN @Agt4395 TO 717
60   OUTPUT @Agt4395;"*rst"
70   !
80   MASS STORAGE IS "a:\"
90   OUTPUT @Agt4395;"STODMEMO"
100  !
110  PRINT "  ENTER SOURCE FILE NAME ON INSTRUMENT ?        ";
120  INPUT Src_file$
130  PRINT Src_file$
140  !
150  PRINT "  ENTER DESTINATION FILE NAME ON CONTROLLER ?   ";
160  INPUT Dst_file$
170  PRINT Dst_file$
180  !
190  Copy_from_instr(@Agt4395,Src_file$,Dst_file$)
200  !
210  END
220  !
230  !  copy_from_instrument
240  !
250  SUB Copy_from_instr(@Agt4395,Src_file$,Dst_file$)
260      DIM Len$[6],Img$[32],Dmy$[2]
270      !
280      ON ERROR GOTO Skip_purge
290      PURGE Dst_file$
300 Skip_purge:  OFF ERROR
310      CREATE Dst_file$,1
320      ASSIGN @Dst_file TO Dst_file$
330      !
340      CLEAR @Agt4395
350      OUTPUT @Agt4395;"CLES"
360      OUTPUT @Agt4395;"ROPEN """;Src_file$;""""
370      IF FNCheck_error(@Agt4395,"<CPFI: ropen>")=-1 THEN SUBEXIT
```

Figure 8-28. Sample Program: File Transfer from 4395A to External Controller (1/2)

```
380     !
390     LOOP
400         OUTPUT @Agt4395;"READ?"
410         ENTER @Agt4395 USING "#,2A";Dmy$
420         ENTER @Agt4395 USING "#,6A";Len$
430         Block_size=VAL(Len$)
440         !
450         IF Block_size=0 THEN
460             ENTER @Agt4395 USING "%,A";Dmy$
470             ASSIGN @Dst_file TO *
480             OUTPUT @Agt4395;"CLOSE"
490             SUBEXIT
500         END IF
510         !
520         ALLOCATE Dat$[Block_size]
530         Img$="#,"&VAL$(Block_size)&"A"
540         ENTER @Agt4395 USING Img$;Dat$
550         ENTER @Agt4395 USING "%,A";Dmy$
560         OUTPUT @Dst_file USING Img$;Dat$
570         DEALLOCATE Dat$
580         !
590         IF FNCheck_error(@Agt4395,"<CPFI: block read>")=-1 THEN SUBEXIT
600     END LOOP
610 SUBEND
620 !
630 !  Instrument Error Check
640 !
650 DEF FNCheck_error(@Agt4395,Str$)
660     DIM Err$[64]
670     OUTPUT @Agt4395;"OUTPERRO?"
680     ENTER @Agt4395;Err$
690     IF Err$<>"+0,""No error""" THEN
700         PRINT "ERROR: ";Str$;"   ";Err$
710         RETURN -1
720     ELSE
730         RETURN 0
740     END IF
750 FNEND
```

Sample Program: File Transfer from 4395A to External Controller (2/2)

Lines 80 to 90 set the current directory of the external controller to A drive and sets the current directory of the 4395A to the memory disk. You can set the current directory of the 4395A to the internal flexible disk using the **STODDISK** command. The A drive in the external controller may not be detected under a certain environment of the external controller, so change the drive depending on the situation.
Lines 110 to 170 accept the entry of the source file name and the destination file name.
Line 190 calls the subprogram to transfer a file from the 4395A to the external controller.
Lines 280 to 320 prepare for writing to the destination file.
Lines 340 to 370 prepare for reading the source file to the external controller.
Line 400 executes the query command to read data.
Lines 410 to 430 read the part indicating the length of the fixed length block data (see Figure R-1) to obtain the length of the data to be transferred.
Lines 450 to 500 check the data length. If the data length is 0, the transfer process is terminated.
Depending on the data length obtained in lines 530 to 550, the program adjusts the format and reads the data part.
Line 560 writes the data to the destination file.
The maximum length of data transferred at a time is 16 Kbytes. Therefore, if the size of the

source file is greater than 16 Kbytes, the transfer routine, lines 400 to 590, is repeated until transferring all of the data is completed.
Lines 650 to 750 provide a function to check that no error has occurred in the 4395A.

File Transfer from External Controller to 4395A

This program transfers a specified file in the current directory of the storage device (A drive in the sample program of Figure 8-29) connected to the external controller to the current directory of the selected storage device (memory disk in the sample program of Figure 8-29) of the 4395A, giving a file name you desire.

Note

Use a **fig8-29.lif** in the sample program disk when you transfer binary files for the LIF format. The WOPEN command in the line 370 takes its arguments in a different way.

This program, when executed, first prompts you to enter a source file name, as shown below. Enter the name of a file you want to transfer. Note that the file name of the LIF format can be distinguished between capital and small letters.

```
ENTER SOURCE FILE NAME ON CONTROLLER ?
```

Then, the program prompts you to enter a destination file name as shown below, and in this example, SAMPLE.STA (DOS format) or SAMPLE_S (LIF format) has been entered as the source file name. Enter the size correctly in bytes.

```
ENTER SOURCE FILE NAME ON CONTROLLER ?       SAMPLE.STA (SAMPLE_S)
ENTER SOURCE FILE SIZE ?
```

Then, the program prompts you to enter the destination file name, as shown below (in this example, the size of SAMPLE.STA or SAMPLE_S is 12288 bytes). Enter the file name you want to give on the destination storage device. Note that a file with the same name will be overwritten, if it already exists.

```
ENTER SOURCE FILE NAME ON CONTROLLER ?       SAMPLE.STA (SAMPLE_S)
ENTER SOURCE FILE SIZE ?                     12288
ENTER DESTINATION FILE NAME ON INSTRUMENT ?
```

```
10   !
20   !  Fig.8-29  File transfer (Controller -> Instrument)
30   !
40   DIM Src_file$[50],Dst_file$[50]
50   ASSIGN @Agt4395 TO 717
60   OUTPUT @Agt4395;"*rst"
70   !
80   MASS STORAGE IS "a:\"
90   OUTPUT @Agt4395;"STODMEMO"
100  !
110  PRINT "  ENTER SOURCE FILE NAME ON CONTROLLER ?        ";
120  INPUT Src_file$
130  PRINT Src_file$
140  !
150  PRINT "  ENTER SOURCE FILE SIZE ?                      ";
160  INPUT Src_size
170  PRINT Src_size
```

Figure 8-29. Sample Program: File Transfer from External Controller to 4395A (1/2)

```
180 !
190 PRINT "  ENTER DESTINATION FILE NAME ON INSTRUMENT ?   ";
200 INPUT Dst_file$
210 PRINT Dst_file$
220 !
230 Copy_to_instr(@Agt4395,Src_file$,Src_size,Dst_file$)
240 !
250 END
260 !
270 !  copy_to_instrument
280 !
290 SUB Copy_to_instr(@Agt4395,Src_file$,Src_size,Dst_file$)
300     DIM Img$[32]
310     Max_bsize=16384
320     !
330     ASSIGN @Src_file TO Src_file$
340     !
350     CLEAR @Agt4395
360     OUTPUT @Agt4395;"CLES"
370     OUTPUT @Agt4395;"WOPEN """;Dst_file$;""""
380     IF FNCheck_error(@Agt4395," <CPTI: wopen>")=-1 THEN SUBEXIT
390     Xfr_done=0
400     !
410     LOOP
420         SELECT (Src_size-Xfr_done)
430             CASE >Max_bsize
440                 Block_size=Max_bsize
450             CASE 0
460                 ASSIGN @Src_file TO *
470                 OUTPUT @Agt4395;"CLOSE"
480                 SUBEXIT
490             CASE ELSE
500                 Block_size=(Src_size-Xfr_done)
510         END SELECT
520         Xfr_done=Xfr_done+Block_size
530         !
540         ALLOCATE Dat$[Block_size]
550         !
560         Img$="#,"&VAL$(Block_size)&"A"
570         ENTER @Src_file USING Img$;Dat$
580         !
590         Img$="8A,ZZZZZZ,"&VAL$(Block_size)&"A"
600         OUTPUT @Agt4395 USING Img$;"WRITE #6",Block_size,Dat$,END
610         DEALLOCATE Dat$
620         IF FNCheck_error(@Agt4395," <CPTI: block write>")=-1 THEN SUBEXIT
630     END LOOP
640 SUBEND
650 !
660 !  Instrument Error Check
670 !
680 DEF FNCheck_error(@Agt4395,Str$)
690     DIM Err$[64]
700     OUTPUT @Agt4395;"OUTPERRO?"
710     ENTER @Agt4395;Err$
720     IF Err$<>"+0,""No error""" THEN
730         PRINT "ERROR: ";Str$;"   ";Err$
740         RETURN -1
750     ELSE
760         RETURN 0
770     END IF
780 FNEND
```

Sample Program: File Transfer from External Controller to 4395A (2/2)

Lines 80 to 90 set the current directory of the external controller to A drive and sets the current directory of the 4395A to the memory disk. You can set the current directory of the 4395A to the internal flexible disk using the **STODDISK** command. The A drive in the external controller may not be detected under a certain environment of the external controller, so change the drive depending on the situation.

Lines 110 to 210 accept the entry of the source file name and its size and the destination file name.

Line 230 calls the subprogram to transfer a file from the external controller to the 4395A.

Lines 370 to 380 prepare for writing the file to the destination storage device.

Lines 420 to 510 calculate the length of the data that has not been transferred based on the source file size previously entered and the length of the data that has been already transferred. If the length of the remaining data does not exceed 16 Kbytes, it is set as the transfer data length; otherwise, 16 Kbytes is set as the transfer data length. Note that, if the length of the data not transferred is 0 at this time, the transfer process is terminated.

Lines 560 to 570 read data, whose amount is specified by the transfer data length, from the source file.

Lines 590 to 600 write data to the destination file in the fixed length block format (see Figure R-1).

The maximum length of data transferred at a time is 16 Kbytes. Therefore, if the size of the source file is greater than 16 Kbytes, the transfer routine, lines 420 to 620, is repeated until transferring all of the data is completed.

Lines 680 to 780 provide a function to check that no error has occurred in the 4395A.

Note

To transfer a file from the external storage device to the 4395A, you must check the file size (number of bytes) in advance .

Displaying List of Files in Current Directory

This program displays the list of the files in the current directory.

```
10   !
20   !  Fig.8-30  File list
30   !
40   ASSIGN @Hp4395 TO 717
50   OUTPUT @Hp4395;"*rst"
60   !
70   Dir_instr(@Hp4395)
80   !
90   END
100  !
110  !  Dir_instr
120  !
130  SUB Dir_instr(@Hp4395)
140      DIM Stor_dev$[5],Curr_dir$[50],File_name$[13]
150      !
160      OUTPUT @Hp4395;"STODMEMO?"
170      ENTER @Hp4395;A
180      IF A=1 THEN
190        Stor_dev$="MEMO"
200      ELSE
210        Stor_dev$="DISK"
220      END IF
230      OUTPUT @Hp4395;"CWD?"
240      ENTER @Hp4395;Curr_dir$
250      PRINT "["&Stor_dev$&"]: "&Curr_dir$
260      PRINT "Size[byte]    File Name"
270      PRINT "------------------------"
280      OUTPUT @Hp4395;"FNUM?"
290      ENTER @Hp4395;File_count
300      IF File_count>=1 THEN
310          FOR I=1 TO File_count
320              OUTPUT @Hp4395;"FNAME? ";I
330              ENTER @Hp4395;File_name$
340              OUTPUT @Hp4395;"FSIZE? """&File_name$&""""
350              ENTER @Hp4395;File_size
360              PRINT USING "XX,DDDDDD,XXXX,K";File_size,File_name$
370          NEXT I
380      END IF
390  SUBEND
```

Figure 8-30. Sample Program: Displaying List of Files in Current Directory of 4395A

Line 70 calls the subprogram to display the list of the files in the current directory.
Lines 160 to 250 check the storage device currently selected and its current directory name, and then display the result.
Lines 280 to 290 check the number of the files in the current directory.
If there are any files in the current directory, lines 300 to 380 check the name and size of every file and display them.

The following is the output result of the program, assuming that the selected storage device is the memory disk and the current directory, \TEST, contains 2 files, FILE1.STA (size: 24576 bytes) and FILE2.TIF (size: 16384 bytes) and 1 directory, DIR1. For size of a directory, -1 is displayed. To view the list of the files in DIR1, use the `CHAD` command to change the current directory to DIR1 and then execute this program again.

```
[MEMO]: \TEST
Size[byte]    File Name
------------------------
      -1    ..\
      -1    DIR1\
   24576    FILE1.STA
   16384    FILE2.TIF
```

Hints and Notes on Programming

This section provides hints and information that can make your program better in practical applications.

Increasing your program execution speed

Because the analyzer's CPU interleaves processing measurements and executing a program, program execution speed depends on the measurement conditions. The display process also requires processing time.

To increase program speed (increase throughput), set the analyzer to the following conditions:

- If you do not need to measure the DUT when executing a program, set TRIGGER MODE to HOLD.
- If you need to measure the DUT but do not need to display the traces on the screen, set DISPLAY ALLOCATION to ALL BASIC.
- If you need to measure the DUT and display traces, but do not need to use the marker function, preset all markers.
- When you use the I/O port, use the `READIO` and `WRITEIO` commands to input or output data to the port directly.
- If you change channels in a program, set Dual Channel to ON before changing channels to avoid the setup time for the channel. For example, when you change channels in a program, set Dual Channel to ON and Display Allocation to All BASIC to decrease the switching time between channels 1 and 2.

`ANAOCH` Command

If you want to retrieve measurement data from a channel that is not active, you can switch the channel effectively by using the GPIB command `ANAOCH`. This command swaps the channel to be processed, while the active channel remains unchanged, thereby accelerating your program execution. Refer to the `ANAOCH` entry in the command reference in this manual.

Note Using `ANAOCH` is valid only if the dual channel function is ON by the GPIB command `DUAC ON`.

Self-assigning of an GPIB Address

When you want to make your program distinguish the system where it is running, the external controller, or the Instrument BASIC, it is helpful to use `SYSTEM$("SYSTEM ID")` as shown below:

```
10   IF SYSTEM("SYTEM ID")="HP4395A" THEN
20     ASSIGN @Hp4395 TO 800
30   ELSE
40     ASSIGN @Hp4395 TO 717
50     ABORT 7
60     CLEAR @Hp4395
70   END IF
```

For example, the line below, which appears in most of the programs in this manual,

```
10 ASSIGN @Hp4395 TO 717   ! WHEN iBASIC is used, change "717" to "800".
```

can be replaced with the program shown above. This enables the program to assign its GPIB address by itself.

Note In the example above, lines 50 and 60 contain the commands to reset the GPIB bus and are not essential for changing the GPIB address.

Key Stroke Recording

Key stroke recording is one of the functions provided with HP Instrument BASIC. This function allows you to write a program with a sequence of key strokes that you make on the front panel. No keyboard is required. See "Easy Program Writing" in Chapter 1 for how to use the function. The function also helps you to find the GPIB command corresponding GPIB command to the key you press.

Solving Problems on Your Program

This section provides typical troubleshooting for when your program does not work as you expected. Check that the following items are satisfied.

If There Is No Response From an Instrument on the GPIB Bus

Check all GPIB addresses and cable connections.

If the Disk Cannot Be Read

- Check that you specify the correct mass storage where your data should reside.
- Check if the disk is corrupted. You may want to verify the disk on PC.

If an GPIB Command Error Occurs

- Verify the GPIB command is appropriate and is used properly.
- Check if the 4395A operates in synchronization with the controller.

If a Query Error Occurs

- Check if the 4395A operates in synchronization with the controller.
- Check that the number of returned values is equal to that you expect in the program, if the query returns multiple values.

9

Introducing HP Instrument BASIC System

This chapter introduces the analyzer's HP Instrument BASIC (IBASIC) and describes how to connect and use a keyboard. Read this chapter before using HP Instrument BASIC with the analyzer for the first time. The topics covered in this chapter are:

- Overview of HP Instrument BASIC
- Controlling the analyzer
- Using HP Instrument BASIC for the first time
- Entering BASIC Statements from the front panel keys
- Getting into/out of the EDIT mode
- Editing programs in the EDIT mode
- Listing programs
- Saving programs
- Listing file names
- Getting programs
- On Key Label function
- Increasing program speed
- Pass Control Between the External Controller
- External RUN/CONTinue connector
- Graphics
- Softkeys used for HP Instrument BASIC operation

Overview of HP Instrument BASIC

HP Instrument BASIC (IBASIC) can be used for a wide range of applications from simple recording and playback of measurement sequences to remote control of other instruments.

HP Instrument BASIC is a complete system controller residing inside your analyzer. It communicates with your analyzer via GPIB commands and can also communicate with other instruments, computers, and peripherals over the GPIB interface.

The HP Instrument BASIC's programming interface includes an editor and a set of programming utilities. The utilities allow you to perform disk I/O, renumber, secure, or delete all or part of your program.

The HP Instrument BASIC command set is similar to the command set of HP 9000 Series 200/300 BASIC. Therefore, HP Instrument BASIC programs can be run on any HP BASIC workstation with few if any changes. Porting information can be found in the *HP Instrument BASIC Programming Techniques* of the *HP Instrument BASIC Users Handbook.*

Controlling the Analyzer

HP Instrument BASIC can control the analyzer (itself) through the "internal" GPIB bus. This means that an analyzer with HP Instrument BASIC includes both a controller and an analyzer in the same instrument. They are connected through an internal GPIB bus.

Note

The select code of the internal GPIB interface is 8, and the GPIB address of the analyzer can be any number from 0 to 30. In this manual, we use "800" for the device selector of the analyzer.

For more information on GPIB addresses and device selectors, see "Device Selectors" in the *HP Instrument BASIC Interfacing Techniques* of the *HP Instrument BASIC Users Handbook* and "Available I/O Interfaces and Select Codes".

Using HP Instrument BASIC for the First Time

Allocating Screen Area for HP Instrument BASIC

Because all of the analyzer's screen is allocated for analyzer operation after power ON, you must allocate screen area for HP Instrument BASIC when you want to use it. The analyzer provides four display allocation types. Select one of them using `DISPLAY ALLOCATION` under (Display).

Let's try

1. Press the following key and softkeys:

   (Display) `MORE` `DISPLAY ALLOCATION`

2. Press the following softkey.

   `ALL BASIC`

   The screen is cleared and all of the screen area is allocated for HP Instrument BASIC.

3. Press the following softkey.

   `ALL INSTRUMENT`

   The total screen area is reallocated as the analyzer display.

4. Press the following softkey:

   `HALF INSTR HALF BASIC`

   The screen area is allocated so that the upper half of the screen is used for the analyzer operation and the lower half is used for HP Instrument BASIC.

5. Press the following softkey:

   `BASIC STATUS`

   Three blank lines appear at the display line (lower area of the screen). This area is used by HP Instrument BASIC to input commands and to display messages.

Entering BASIC Statements from the Front Panel Keys

The analyzer's HP Instrument BASIC allows you to enter and execute statements from the front panel keys (if the external mini-DIN keyboard is not connected).

Press the following key and softkeys from the front panel:

(System) IBASIC MORE [1/3] MORE [2/3] COMMAND ENTRY

The Command Entry menu is displayed on the softkey menu area, and the active entry area displays the letters, the digits 0 through 9, and some special characters including mathematical symbols. Three sets of letters can be scrolled using the step keys, (⇑) and (⇓). To enter a statement, press the step keys for the desired letter set, rotate the knob until the arrow "↑" points at the first letter, then press SELECT LETTER. Repeat this until the complete statement is entered, then press DONE to execute the statement.

Getting into/out of the EDIT Mode

Pressing the following key and softkey allows you to enter the EDIT mode immediately, irrespective of Display Allocation.

Getting into the EDIT Mode

Press the following key and softkeys from the front panel:

(System) IBASIC Edit

Entering the EDIT Mode from the Keyboard

Use the following keys to enter the EDIT mode with the cursor positioned at the specified line number. The *line_number* can be omitted. Press the following key among the 3 menus which leads to the (Shift) - (F9) key.

EDIT *line_number* (Enter)

or type as follows:

EDIT *line_number* (Enter)

To use the keyboard, the Keyboard Input Line must be allocated on the screen. If it is not, press (Display) MORE DISPLAY ALLOCATION and select any allocation except ALL INSTRUMENT.

Getting Out of the EDIT Mode

The EDIT mode is exited by pressing (Shift) - (Alt) - (F4), (ESC), and (Home) from the keyboard (or by pressing the END EDIT softkey).

Editing Programs in the EDIT Mode

This section describes how to edit a program while in the EDIT mode, the topics are:

- Deleting characters
- Inserting characters
- Moving the cursor
- Scrolling lines and pages
- Jumping lines
- Inserting/deleting/recalling lines
- Clearing lines

See "The Keyboard" for more information on functions of each key.

Deleting Characters

There are two functions you can use to delete characters: "Back space" and "Delete characters."

Back Space

Pressing [Back space] on the front panel (or on the keyboard) erases the character to the left of the cursor and moves the cursor left to the position of the erased character.

Deleting Characters

Pressing [Delete char] from the keyboard deletes the character at the cursor's position.

Inserting Characters

The EDIT mode is always in the insert mode. Characters you type at the keyboard are inserted before the current cursor position. (Pressing [Insert] performs no function.)

Moving the Cursor

The following key operations allow you to move the cursor horizontally along a line:

From the front panel	From the keyboard
Turning the knob	Pressing [◀] and [▶]

Scrolling Lines and Pages

Scrolling Lines

The following key operations enable you to scroll lines up and down:

From the front panel	From the keyboard
Pressing [⇧] and [⇩]	pressing [▲] and [▼]

Scrolling Pages

Pressing Page Up and Page Down from the keyboard causes the display to scroll up and down in one page increments.

Jumping from the Current Line

Jumping to a Specified Line

You can specify a line by using a line number or a label name when jumping from the current line as follows:

GOTO LINE *line_number* Enter

or

GOTO LINE *label_name* Enter

If the label specified is not defined in the program, an error will occur.

Jumping to the Top/Bottom of a Program

Pressing the following keys allows you to jump to top or bottom of the program:

Shift-▲

Shift-▼

Inserting/Deleting/Recalling Lines

Shift - Insert inserts a new line above the current cursor position.

Shift - Delete deletes the line at which the cursor is.

RECALL LINE recalls the last deleted line.

Clearing Line

Pressing Shift - End clears a line from the current cursor position to the end of the line.

Renumbering Program Line Numbers

The `REN` command allows you to renumber the program currently in memory. You should execute the `REN` command after exiting the EDIT mode. Press the following key among the 3 menus which leads to the Shift - F9 key.

`RENumber` Enter

or

`REN` Enter

You can specify the starting value, increment value, beginning line number, and the ending line number when renumbering a program as follows:

`RENumber` *starting_value*, *increment* `IN` *beginning_line_number*, *ending_line_number* Enter

or type as follows:

`REN` *starting_value*, *increment* `IN` *beginning_line_number*, *ending_line_number* Enter

line_label can be also use instead of *line_number*. For more information, see the *HP Instrument BASIC Language Reference* of the *HP Instrument BASIC Users Handbook.*

Listing Programs

The system can list the program on the screen and to a printer.

Listing on the Screen

You can list a program on the screen as follows:

1. Because the system lists a program in the print area, the Print Area must be allocated on the screen. For example:

   Display `MORE` `DISPLAY ALLOCATE` `ALL BASIC`

   All of the screen area is allocated for the print area.
2. Type as follows:

   `LIST` Enter

Listing to the Printer

Note For hard copy output, an parallel cable must connect the analyzer to the printer.

1. Set the output device to a printer as follows:

   `PRINTER IS PRT` Enter
2. Type and press as follows:

LIST (Enter)

The program is listed on the printer.

3. Set the output device to LCD as follows:

   PRINTER IS CRT (Enter)

Saving Programs (SAVE)

1. To use the built-in disk drive, insert a 2DD disk or 2HD disk into the disk drive.
2. If you are using a flexible disk for the first time, set the disk format to LIF or DOS and initialize the disk. See "To Save and Recall" in *Operation Manual* for the procedure.

Note When you turn ON the 4395A, you can have it automatically execute a particular program which was backuped in the memory disk. See "Memory Disk" in Chapter 10 for how to backup a program in the memory disk.

3. If the display allocation is ALL INSTRUMENT, change the allocation to either HALF INSTRument HALF BASIC or ALL BASIC. For example:

   (Display) MORE DISP ALLOCATION ALL BASIC

4. Select the storage units: the built-in flexible disk drive and the RAM disk memory.

   For the built-in disk drive, enter
   MSI ":INTERNAL" (Enter) or MSI ":INTERNAL,4,0" (Enter)
   For the RAM disk memory, enter
   MSI ":MEMORY,0" (Enter) or MSI ":MEMORY,0,0" (Enter)

5. Press the following key among the 3 menus which leads to the (Shift) - (F9) key. And type in the filename to which you will store the program as follows:

   SAVE *file_name* (Enter)

   You can also save the file from the keyboard. Type and press as follows:

   SAVE *file_name* (Enter)

   The program is stored on the disk.

Note If you get the error −257, "File name error", a file on the disk already has the name you are trying to use. In this case, you have three choices:

- Pick a new file name that doesn't already exist. To determine which file names are already being used, use the "CAT" command (see below).
- Replace an existing file, use the "RE-SAVE" statement.
- Purge the old file using the PURGE command, then save the new one.

Listing File Names (CAT)

Listing to Screen

Press the following key and softkeys:

1. If the display allocation is ALL INSTRUMENT or BASIC STATUS, change the allocation to either HALF INSTRument HALF BASIC or ALL BASIC. For example:

   (Display) MORE DISP ALLOCATION ALL BASIC

2. Press the following key among the 3 menus which leads to the (Shift) - (F9) key:

   CAT (Enter)

   You can list from the keyboard as follows:

   CAT (Enter)

   The file names stored on the disk are listed on the screen.

Note

Because the CAT statement outputs 80 columns to a line and the maximum number of columns to a screen is 61, each line is wrapped at the 62th column. If you do not want the list to wrap around, execute the following statement before executing the CAT command.

```
PRINTER IS CRT;WIDTH 80
```

Listing to Printer

Note

For hard copy output, an parallel cable must connect the analyzer to the printer.

1. Set the output device to be a printer as follows:

   PRINTER IS PRT;WIDTH 80 (Enter)

   Type and press as follows:

   CAT (Enter)

   The program is listed on the printer.

2. Get the output device back to LCD:

   PRINTER IS CRT (Enter)

Getting Programs (GET)

You can retrieve a program from the disk as follows:

1. If the display allocation is ALL INSTRUMENT, change the allocation to either HALF INSTRument HALF BASIC or ALL BASIC. For example:

   (Display) `MORE` `DISP ALLOCATION` `ALL BASIC`

2. Press the following key among the 3 menus which leads to the (Shift) - (F9) key and type the filename you want to retrieve:

   `GET` *file_name* (Enter)

   You can get the file from the keyboard.

On Key Label Function

The HP Instrument BASIC allows you to define softkeys from within a program. The softkey labels you define will appear when pressing `ON KEY LABELS` or the (Shift) - (F10) key on the Keyboard. The labels are displayed while running the program.

Example:

```
......
100  ON KEY 1 GOTO 150
110  ON KEY 2 LABEL "Print" GOSUB Report
......
```

You can also use the `KEY` statement to automatically display the label. This prevents you from pressing (Shift) - (F10) or choosing `ON KEY LABELS` while the user program is running. The `KEY` statement is used to display the softkey labels defined. The following set of statements is the same as the key strokes (System) `IBASIC` `ON KEY LABELS`:

```
......
200 OUTPUT Hp4395;"KEY 47"    ! SYSTEM key
210 OUTPUT Hp4395;"KEY 0"     ! IBASIC softkey
220 OUTPUT Hp4395;"KEY 7"     ! ON KEY LABELS softkey
......
```

For more information on the ON KEY statement, see the *HP Instrument BASIC Language Reference* of the *HP Instrument BASIC Users Handbook.*

Pass Control Between the External Controller

This section describes how to pass control between Intstrument BASIC and the controller.

Pass Control

To pass active control to HP Instrument BASIC:

```
PASS CONTROL 717 Enter
```

Pass Control (On External Controller)

While the 4395A has control, it is free to address devices to talk and listen as needed. As the active controller, the 4395A can send messages to and read replies back from printers and plotters.

Note The ability to assert the GPIB interface clear line (IFC) and remote enable line (REN) are reserved for the system controller. Even when HP Instrument BASIC has active control, it is denied these functions.

`ABORT 7`	*assert the interface clear line (IFC)*
`REMOTE 7`	*assert the remote enable line (REN)*

To return active control to the system controller:

```
PASS CONTROL 721 Enter
```

Return Control (On HP Instrument BASIC)

Or, you can return control to the external controller by resetting the GPIB as follows:

```
ABORT 7 Enter
```

Return Control (On External Controller)

To Execute an HP Instrument BASIC Command from the External Controller

```
10 !
20 ! To Transfer the Program to iBASIC (on External Controller)
30 !
40    ABORT 7
50    ASSIGN @Hp4395 TO 717
60    INPUT "FILENAME?",File_name$
70    OUTPUT @Hp4395;"PROG:DEL:ALL"
80    OUTPUT @Hp4395;"PROG:DEF #0"
90    ASSIGN @File TO File_name$
100   ON ERROR GOTO Done
110     DIM Line$[1024]
120     LOOP
130       Line$=""
140       ENTER @File;Line$
150       OUTPUT @Hp4395;Line$
160     END LOOP
170   Done:  !
180       OFF ERROR
190       OUTPUT @Hp4395;" " END
200   !
210   OUTPUT @Hp4395;"PROG:EXEC ""RUN"""
220   END
```

Figure 9-1. Sample Program : To Transfer the Program to IBASIC (on External Controller)

This Program transfers the program file in the mass storage of the external controller.

Lines 70 to 80 scratch any program that currently exists in the tester's HP Instrument BASIC editor and open the editor.

Lines 90 to 160 transfer the program by line to the analyzer.

Line 190 closes the HP Instrument BASIC Editor.

Line 210 executes the transferred program.

To Load an Array in an HP Instrument BASIC Program to the External Controller

```
10  !
20  ! To Load iBASIC Program Array (on External Controller)
30  !
40    ABORT 7
50    ASSIGN @Hp4395 TO 717
60    DIM Passed(1:801,1:2)
70    OUTPUT @Hp4395;"PROG:NUMB? ""Dat"";"
80    ENTER @Hp4395;Passed(*)
90    END
```

Figure 9-2.
Sample Program : To Load HP Instrument BASIC Program Array (on External Controller)

This program retrieves the array generated in the sample program listed in Figure 3-5 when that program is executed in HP Instrument BASIC. This information is transferred to the external controller.

Lines 70 to 80 returns the program array `Dat(1:801,1:2)` of Figure 3-5 using `PROG:NUMB? "Dat"` query. The array is entered into `Passed(1:801,1:2)`.

Available I/O Interfaces and Select Codes

Available interfaces and their select codes in the analyzer's HP Instrument BASIC are listed in the following table:

Select Codes	Devices
1	LCD
2	Keyboard
7	External GPIB interface
8	Internal GPIB interface

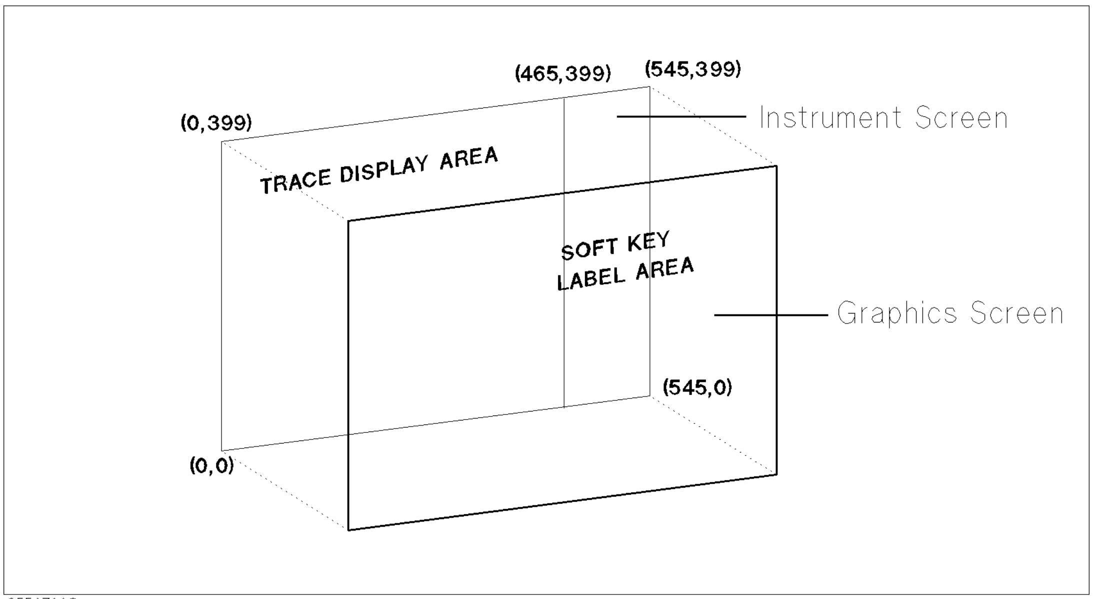
External RUN/CONTinue Connector

You can trigger RUN or CONT of the HP Instrument BASIC program externally by applying a TTL signal through the RUN/CONT connector on the rear panel of the 4395A. The signal should be more than 20μsec in width and follow the negative logic. The program is triggered at the trailing edge of the pulse.

Graphics

HP Instrument BASIC adds graphics capability to the analyzer. You can draw pictures on the LCD independent of the grids and traces.

The analyzer has two screens, the instrument screen and the graphics screen. These two screens are always displayed together on the LCD and are not separately selectable. The instrument screen consists of a trace display area and a softkey label area. The HP Instrument BASIC editor is also displayed on the trace display area. The graphics screen covers the entire instrument screen as shown in Figure 9-3. The graphics screen is like an independent transparent overlay in front of the instrument screen. Therefore, you can draw figures in both the trace display and softkey label areas.



C5507002

Figure 9-3. Screen Structure

Each point on the graphics screen is addressable using a coordinate address as shown in Figure 9-3. The bottom left corner is the origin (0,0) and the top right corner is the maximum horizontal and vertical end points (393,299). The MOVE and DRAW statement parameters are specified using these coordinate values. Because the aspect ratio of a graphics screen is 1, you need not adjust the aspect ratio when drawing figures.

HP Instrument BASIC Graphics Commands

The analyzer's HP Instrument BASIC has three graphics commands; MOVE, DRAW, and GCLEAR.

`MOVE` Moves the pen from its current position to the specified coordinates.

`DRAW` Draws a line from the current pen position to the specified coordinates.

`GCLEAR` Clears the graphics screen, moves the pen from its current position to the origin (0,0), and selects pen 1.

Note

The total times of executing the MOVE and DRAW commands is up to 1933, even if the pen position is not changed.

Hard Copies

Graphics hard copies can be obtained with the printing function. Select PRINT under (Copy).

Initial settings

When power is turned ON, the default settings are as follows:

- MOVE 0,0

Example of Graphics Programming

This section describes an example of a simple program for drawing lines on the graphics screen.

Drawing a Straight Line

The following HP Instrument BASIC program will draw a line from coordinate (50,200) to coordinate (300,200) on the display.

```
GCLEAR             ! INITIALIZE GRAPHICS MODE
MOVE 50,200        ! MOVE PEN TO COORDINATE  (50,200)
DRAW 300,200       ! DRAW A LINE TO COORDINATE (300,200)
END
```

Drawing a Circle

Trying to express all graphical images using only straight lines is tedious, slow, and difficult. This example describes a subprogram you can use to draw a circle. It can draw a circle by passing the center coordinates and the radius as arguments to the following subroutine. This subroutine can be used as a base for drawing arcs, setting different values for Theta, etc.

```
SUB Drawcircle(Centx,Centy,R)   !
  DEG                           ! USE DEGREES FOR ANGLE EXPRESSIONS
  X=Centx+R                     !
  Y=Centy                       !
  MOVE X,Y                      ! MOVE PEN TO INITIAL POINT
  For Theta=1 to 360            !
    X=INT(COS(Theta)*R+Centx)   ! NEXT X COORDINATE ON CIRCLE
    Y=INT(SIN(Theta)*R+Centy)   ! NEXT Y COORDINATE ON CIRCLE
    DRAW X,Y                    ! UNTIL STARTING POINT IS REACHED
  NEXT Theta                    !
SUBEND                          !
```

The Keyboard

This section provides the key binding information of the keyboard.

Character Entry Keys

The character entry keys are arranged in the familiar QWERTY typewriter layout, but with additional features.

(Caps) Sets the unshifted keyboard to either upper-case (which is the default after power ON) or lower-case (normal typewriter operation).

(Shift) You can enter standard upper-case and lower-case letters, using the (Shift) key to access the alternate case.

(Enter) Has three functions:

- When a running program prompts you for data, respond by typing in the requested data and then press (Enter). This signals the program that you have provided the data and that it can resume execution.
- When typing in program source code, the (Enter) key is used to store each line of program code.
- After typing in a command, the (Enter) key causes the command to be executed.

(CTRL) In the EDIT mode, (CTRL) allows you to control the editor in the same as using the cursor-control, display-control, and editing keys. For more detail, see "Using (CTRL) Key in Edit Mode".

(Backspace) Erases the character to the left of the cursor and moves the cursor to the erased character's position on the line.

(Tab) Performs no function.

Cursor-Control and Display-Control Keys

(▲) (▼) Allow you to scroll lines up and down in the print display area. Shifted, these keys cause the display to scroll toward the top or bottom of the display.

(▶) (◀) Allow you to move horizontally along a line. Shifted, these keys allow you to "jump" to the left and right limits of the current line.

(Page Up) (Page Down) Cause the display to scroll up or down in one page increments.

(Home) Performs no function.

Numeric Keypad

The numerical keypad provides a convenient way to enter numbers and perform arithmetic operations. Just type in the arithmetic expression you want to evaluate, then press (Enter). The result is displayed in the lower-left corner of the screen.

(Enter)	Performs the same function as the (Enter) key. The numerical keypad serves the same function as the numerical keypad on the front panel of the analyzer.
(Num Lock)	Performs no function. Pressing the (Num Lock) key causes the LED ON/OFF, but the keys are performes as the numerical keypad only.

Editing Keys

(Insert)	Performs no function. The HP Instrument BASIC is always in the insert mode. The characters you type are always inserted to the left of the cursor.
(Shift) - (Insert)	Inserts a new line above the cursor's current position (edit mode only).
(Delete)	Deletes the character at the cursor's position.
(Shift) - (Delete)	Deletes the line containing the cursor (edit mode only).
(End)	Delete the line containing the cursor except the line number.
(Shift) - (End)	Clears from the current cursor position to the end of the line.
(Home)	Clears the entire alpha screen. In EDIT mode, this exits the EDIT mode.

Program Control Keys

The following keys allow you to control execution of the program stored in the analyzer's memory.

(Pause)	(Pause) or (Alt) - (F4) pauses program execution after the current line. Pressing `Continue` in the System menu resumes program execution from the point where it paused. (Shift) - (Alt) - (F4) stops program execution after the current line. To restart the program, press `Run` in the System menu. When in the editor mode, (Shift) - (Alt) - (F4) exits the edit mode.
(Ctrl) - (Break)	(Ctrl) - (Break) resets program execution immediately without erasing the program from memory (**BASIC RESET**). Pauses program execution when the computer is performing or trying to perform an I/O operation. Press (Alt) - (F5) instead of (Pause) or (Alt) - (F4) when the computer is hung up during an I/O operation, because (Pause) or (Alt) - (F4) works only after the computer finishes the current program line.

System Control Keys

Shift - Page Up (Recall)	Shift - Page Up (Recall) recalls the last line the you entered, executed, or deleted. Several previous lines can be recalled this way. Recall is particularly handy to use when you mistype a line. Instead of retyping the entire line, you can recall it, edit it using the editing keys, and enter or execute it again. Shift-Page Down moves forward through the recall stack.
Alt - F3 (Run)	Starts a program running from the beginning.
Alt - F2 (Continue)	Resumes program execution from the point where it paused.
F12 (IBASIC)	Allows you to type BASIC commands on Keyboard Input Line. If Display Allocation is ALL INSTRUMENT, pressing this key changes the Display Allocation to BASIC STATUS. Shift-F12 changes Display Allocation to ALL INSTRUMENT.

Softkeys and Softkey Control

There are eight softkeys (labeled f1 through f8). The softkey labels are indicated on the right of the analyzer's screen.

Softkey Control Keys

Pressing the following:

F9	Leads to the IBASIC menu, which controls programs and the editor.
Shift - F9	leads to the BASIC menu from which to control a BASIC program. This menu is the same menu displayed when pressing SYSTEM IBASIC from the front panel. In the edit mode, pressing F10 leads to the Edit System menu, which provides softkeys to conveniently enter BASIC commands. Shift - F10 (User) leads to the ON KEY LABEL menu, which are user defined softkeys in a BASIC program. (For information on getting to this menu through the HP Instrument BASIC, see "On Key Label Function".)

Softkeys

F9 and F10 keys leads to the IBASIC menu. Pressing a softkey performs the command labeled or produces a sequence of characters on the keyboard input line (or on the "current line" in the EDIT mode).

Pressing the softkeys on the front panel of the analyzer performs the same functions as pressing the f1 through f8 function keys.

Softkeys Accessed from Shift - F9 Key

IBASIC Menu

Pressing the following:

`Step`	Produces the command "`Step`" on the keyboard input line. Step executes a program at every line.
`Continue`	Produces the command "`Continue`" on the keyboard input line. Resumes program execution from the point where it paused.
`Run`	Produces the command "`Run`" on the keyboard input line. Immediately executes a program.
`Pause`	Produces the command "`Pause`" on the keyboard input line. Pauses program execution after the current program line is executed.
`Stop`	Produces the command "`Stop`" on the keyboard input line. Stops program execution after the current line. To restart the program, press `Run`.
`EDIT`	Produces the command "`EDIT`" on the keyboard input line. After `EDIT` is entered, pressing Enter enters the edit mode.
`ON KEY LABELS`	Leads to a softkey menu defined during program execution, if the softkey menu has been defined.
`CAT`	Produces the command "`CAT`". `CAT` lists the contents of a mass storage directory.
`SAVE`	Produces the command "`SAVE`". `SAVE` creates an ASCII file and copies program lines as strings into that file.
`RE-SAVE`	Produces the command "`RE-SAVE`". `RE-SAVE` creates a specified ASCII file if it does not exist; otherwise, it rewrites a specified ASCII file by copying program lines as strings into that file.
`GET`	Produces the command "`GET`". `GET` reads the specified ASCII file and attempts to store the strings into memory as program lines.
`PURGE`	Produces the command "`PURGE`". `PURGE` deletes a file or directory from the directory of a mass storage media.
`INITIALIZE`	Produces the command "`INITIALIZE`". `INITIALIZE` prepares mass storage media for use by the computer. When `INITIALIZE` is executed, any data on the media is lost.
`MSI []`	Produces the command "`MSI []`" on the keyboard input line. MSI [] specifies the mass strage. `INTERNAL` specifies the internal flexible disk, `MEMORY` specifies the RAM disk.
`SCRATCH`	Produces the command "`SCRATCH`". The `SCRATCH` erases the program in memory. After `SCRATCH` is entered, pressing Enter executes the command.
`RENumber`	Produces the command "`REN`". `REN` renumbers all of the program lines currently in memory.
`LIST`	Produces the command "`LIST`". Lists the program on the screen.

`COMMAND ENTRY`	Leads to the Command entry menu, which allows you to execute the HP Instrument BASIC commands from the front panel keys.
`CLEAR I/O`	Produces the command "`CLEAR I/O`". Pauses I/O operation program. To restart the program, press `Continue`.
`RESET`	Produces the command "`RESET`". Aborts the program.

Softkeys Accessed form (F10) Key

(F10) key allows you to access three different softkey flows dependent on conditions as follows:

- Pressing (F10) accesses the Program Control menu
- In editor mode, pressing (F10) accesses the Edit System menu
- Pressing (Shift)-(F10) accesses the On Key Label menu.

The menus listed above are described in "Instrument BASIC Menu" in the *Operation Manual.*

Using (CTRL) Key in Edit Mode

In the edit mode, pressing (CTRL), holding it down and pressing another key, allows you to control the editor in the same way as pressing control keys such as (▲), (▼), (Insert line), etc.

If you press ...	It performs ...
(CTRL)-(a)	Moves the cursor to beginning of line, (the same function as (Shift)-(◀)).
(CTRL)-(b)	Moves cursor backward one character, (the same function as (◀)).
(CTRL)-(d)	Deletes a character, (the same function as (Delete)).
(CTRL)-(e)	Moves the cursor to end of the line, (the same function as (Shift)-(▶)).
(CTRL)-(f)	Moves cursor forward character along a line, (the same function as (▶)).
(CTRL)-(g)	Allows you to move the cursor to any line number or label, after press (CTRL)-(g), type a line number or label name and press (Enter), the cursor moves to the specified line, (the same function as `GOTO LINE`).
(CTRL)-(h)	Deletes backward one character, (the same function as (Back Space)).
(CTRL)-(j)	Performs the same function as (Enter).
(CTRL)-(k)	Deletes a line from the cursor's current position to the end of the line.
(CTRL)-(m)	Performs the same function as (Enter).
(CTRL)-(n)	Moves the cursor to the next line, (the same function as (▼)).
(CTRL)-(o)	Inserts a new line above the cursor's current position, (the same function as (Shift) - (Insert)).
(CTRL)-(p)	Moves the cursor to the previous line, (the same function as (▲)).

Run Light Indications

⊔ (blank)	Program stopped; can execute commands; CONTINUE not allowed.
_	Program paused; can execute commands; CONTINUE is allowed.
?	BASIC program waiting for input from keyboard; cannot execute commands.
*	This indication has two possible meanings:

- Program running; CANNOT execute commands. CONTINUE not allowed.
- System executing commanded entered from keyboard; CANNOT enter commands.

BASIC Commands Specific to 4395A

The following commands are *not* listed in the *HP Instrument BASIC Language Reference* of the *HP Instrument BASIC Users Handbook*, but are available in the analyzer's HP Instrument BASIC.

DATE

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This command converts a date (given in Day Month Year) into Julian seconds.

- Example Commands

```
PRINT DATE("21 MAY 1991")          ! Displays the date May/21/1991 in Julian seconds
SET TIMEDATE DATE("1 Jan 1991")    ! Set the real time clock to
                                   ! 0:00 Jan/1/1991
Days=(DATE("1 JAN 1991")-DATE("11 NOV 1990")) DIV 86400!
```

Note

The Julian seconds format is used in the real time clock and is defined as the number of seconds that passed since 0:00 Nov/24/−4713

DATE$

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This command formats the number of seconds into a date (DD MMM YYY)

- Example Commands

```
PRINT DATE$(TIMEDATE)        ! Displays the real time clock in a date format
DISP DATE$(2.111510608E+11)  ! Displays 2.111510608E+11 Julian seconds in a date format
```

EXECUTE

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This command executes specific GPIB commands faster than the `OUTPUT` statement.

- Example Commands

```
EXECUTE "SING"
```

READIO

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This command reads the contents of the register used for an I/O port or EXECUTE command.

Item	Description	Range
select code	numeric expression	8: EXECUTE register 15: I/O port
register number	numeric expression	0 to 800 (Select code 8) 0: I/O port

- Example Commands

```
Ioport=READIO(15,0)          ! Substitue data from the I/O port to Ioport

100  EXECUTE "OUTPRESO?"     ! Execute OUTPRESO? query
110  Za=READIO(8,0)          ! Substitue the first return value from the register 0 to Za
120  Fa=READIO(8,1)          ! Substitue the second return value from the register 1 to Fa
130  Zr=READIO(8,2)          ! Substitue the third return value from the register 2 to Zr
140  Fr=READIO(8,3)          ! Substitue the fourth return value from the register 3 to Fr
```

SET TIME

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This command resets the time-of-day given by the real-time clock.

- Example Commands

```
SET TIME TIME("22:00:30")       ! Set the real time clock to 22:00:30
SET TIME Hours*3600+Minutes*60  ! Set the real time clock to Hours:Minutes
```

SET TIMEDATE

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This command resets the absolute seconds (time and day) given by the real-time clock.

- Example Commands

```
SET TIMEDATE DATE("4 JAN 1993")+TIME("10:00:00") ! Set the real time clock to

                                 ! 10:00 Jan/4/1993
SET TIMEDATE TIMEDATE+86400                ! Set the real time clock 1 day ahead
```

TIME

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This command converts data formatted as time of day (HH:MM:SS), into the number of seconds past midnight.

- Example Commands

```
Seconds=TIME("8:37:20")           ! Seconds passed since 0:00 until 8:37:20
SET TIME TIME("8:37:20")          ! Set the real time cloc to 8:37:20
ON TIME TIME("12:10") GOSUB Lunch  ! Go to "Lunch" at 12:10
```

TIME$

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This command converts the number of seconds past midnight into a string representing the time of day (HH:MM:SS).

- Example Commands

```
DISP "The time is: ";TIME$(TIMEDATE) ! Shows the current time based on the real time clock
PRINT TIME$(45296)                   ! Shows the time that passed 45296 seconds since 0:00
```

WRITEIO

Keyboard Executable	Yes
Programmable	Yes
In an IF ... THEN ...	Yes

This statement writes register data in decimal notation to a specified EXECUTE command parameter register or to a specified I/O port.

Item	Description	Range
select code	numeric expression	8: EXECUTE register 15: I/O port
register number	numeric expression	0 to 800 (Select code 8) 0: I/O port
register data	numeric expression	−2147483648 to +2147483647 0 to 255: I/O port

- Example Commands

```
WRITEIO 15,0;12              ! Writes 12 on the I/O port

100  WRITEIO 8,0; 100E6       ! Writes the first argument on the register 0
110  WRITEIO 8,1; 200E6       ! Writes the second argument on the register 1
```

BASIC Commands Not Implemented

The following commands are listed in the *HP Instrument BASIC Language Reference* of the *HP Instrument Users Handbook*, but are not implemented in the analyzer's HP Instrument BASIC.

- `OFF CYCLE`
- `ON CYCLE`

Note

`GCLEAR` and `ON TIMEOUT` commands are available, but the following supplementary items are added.

- GCLEAR

  Moves the pen to (0,0).

- OFF TIMEOUT and ON TIMEOUT

  The interface select code is 7 only.

10

Facilitating Program Execution and Utilizing Storage Devices

You can use the 4395A's softkey interface to run a program previously saved on a storage device (floppy disk or memory disk). Also, you can have the 4395A to automatically execute a certain program whenever the power is turned ON. This chapter explains how to use these useful features as well as how to use the storage devices of the 4395A.

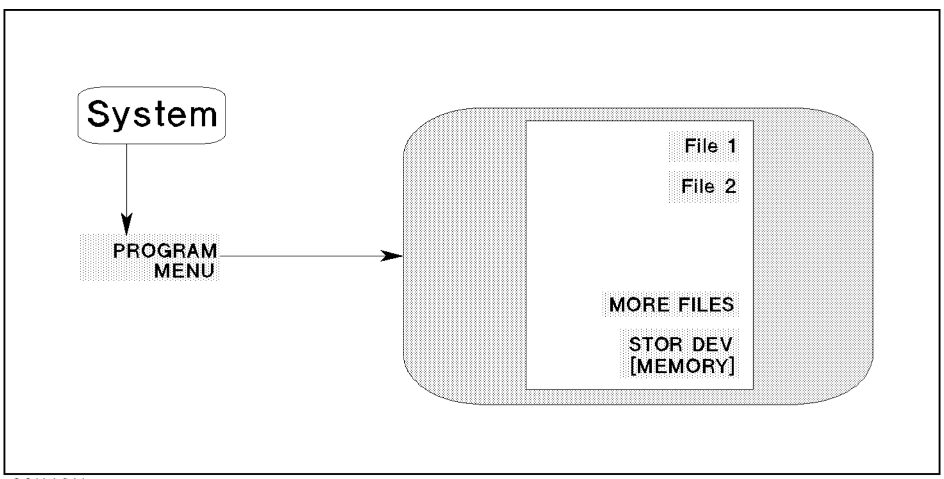
Topics covered include:

- Running a program through the softkey interface
- Automatically starting a program at power-ON time
- Using storage devices

Running a Program through the Softkey Interface

With the 4395A, you can run a program by just choosing the softkey associated with the program. To do so, follow these steps:

1. Open the Program Menu as illustrated below:



CC100201

2. To tell the 4395A where program files are located, toggle the STOR DEV softkey as follows:

Storage Device	Toggle To
Floppy disk drive	STOR DEV [DISK]
Memory disk	STOR DEV [MEMORY]

3. The 4395A displays a menu of softkeys that correspond to available program files residing on the selected storage device. Select your desired program by choosing the associated softkey. Then the 4395A executes the selected program.

Note

- The 4395A may fail to recognize a program file that was created on an external controller such as a PC. If this is the case, use Instrument BASIC to load and re-save the program without adding any file name extension.
- For information on how to save a program file on the memory disk, refer to "Memory Disk".

Automatically Starting a Program at Power-ON (AUTOST)

You can have the 4395A automatically execute a particular program whenever it is turned ON. To use this feature, save the program under the name of "AUTOST" on a floppy disk (note that the floppy disk containing the "AUTOST" program must be kept in the disk drive for the autostart feature to work).

To use the autostart feature, choose the following softkeys in order:

(SYSTEM) PROGRAM MENU STOR DEV

Alternatively, you can save an "AUTOST" program file on the memory disk, and back up the memory disk as explained in "Memory Disk". In this case, the "AUTOST" program will be automatically executed from the backup copy of the memory disk next time you turn ON the 4395A.

When the power is turned ON, the 4395A checks for any "AUTOREC" file. If an "AUTOREC" file exists, it reads the information contained in the file, and then loads and executes the "AUTOST" program.

Note

If the autostart feature does not work, make sure that your selected storage device contains an "AUTOST" program.

Using Storage Devices

The 4395A has two storage devices: a floppy disk drive and a RAM disk. These storage devices support the LIF and DOS formats. The 4395A automatically identifies the data format, almost transparently to the user.

BASIC commands for setting up the storage devices include:

`MSI ":INTERNAL"` and `MSI ":INTERNAL,4,0"` are commands to select the floppy disk drive.

`MSI ":MEMORY"` and `MSI ":MEMORY,0,0"` are commands to select the memory disk.

Note

Use `STODDISK`(floppy disk drive) or `STODMEMO`(memory disk) to set up storage units that are accessed through the following GPIB commands:

- CHAD
- CRED
- DISF
- INID
- PURG
- RECD
- RESAVD
- SAVDASC
- SAVDDAT
- SAVDGRA
- SAVDSTA

The GPIB `FILC` command allows you to copy files between the floppy disk drive and memory disk.

Note

`FILC` does not support copying files between different data formats (LIF and DOS).

You can format a disk through the front panel interface (see Chapter 6 of *Operation Manual* or by using GPIB commands. The following is a sample program that uses GPIB commands to format a disk:

```
10  ASSIGN Hp4395 TO 800
20  OUTPUT Hp4395;"STODDISK"  ! Select a disk drive
30  OUTPUT Hp4395;"DISF DOS"  ! Select DOS format
40  OUTPUT Hp4395;"INID"      ! Start initialization
50  END
```

Floppy Disk Drive

- The `INITIALIZE` command of HP Instrument BASIC can format a disk into the 1.44 MB (2HD) format only; it does not support the 720 KB (2DD) and 270 KB formats.
- The `INITIALIZE` command accepts only the default format option parameter (0), which provides 256 byte sectors.
- The 4395A can read and write a DOS format disk that meets the following specifications:
  720 KB, 80 tracks, double-sided, 9 sectors/track
  1.44 MB, 80 tracks, double-sided, 18 sectors/track
- The `CREATE` and `CREATE DIR` commands are not available for a LIF format disk.

Note The 4395A edition of HP Instrument BASIC does not support an external storage device. The MASS STORAGE IS(MSI) command accepts either "`INTERNAL,4`" (floppy disk drive) or "`MEMORY,0`" (memory disk).

Memory Disk

The 4395A allows you to use 512 KB RAM space as volatile memory disk.

The memory disk must be formatted into either DOS or LIF. If you have backed up the memory disk, turning on the power causes the 4395A to restore the memory disk from the backup copy retaining the original format, so you don't need to reformat the memory disk.

Note When the 4395A is turned OFF, the data residing on the memory disk is lost. You can create a backup copy of the memory disk so that the memory disk is automatically restored from the backup copy next time the 4395A is turned on. To back up the memory disk, press (Save) `BACK UP MEMO DISK`. Alternatively, you can use the `STORMDISK` command over the GPIB.

Note Backup is also important as a means of recovering your data in the event of power interruption or operator error. For example, even if you inadvertently formatted the 4395A memory disk, you could easily recover the data from the backup copy; all you have to do is turn OFF and ON the 4395A or issue the `RESTMDISK` command over GPIB.

Note The memory disk can endure approximately 100,000 cycles of backup operation. This should be more than needed for the lifetime of the product, but you should avoid backing up the memory disk more frequently than actually needed.

11

Command Reference

This chapter summarizes the GPIB commands according to the equivalent front panel keys and softkeys as follows. See Appendixes A to Z for detailed description of the GPIB commands. The appendixes also discribe the GPIB commands that have no corresponding front panel keys or softkeys.

About service commands, see *Service Manual*

Front Panle Key	GPIB Command
Chan 1	CHAN1
Chan 2	CHAN2
Meas	
Network Analyzer	
NETWORK: A/R	MEAS AR
B/R	MEAS BR
A/B	MEAS AB
MORE	
NETWORK: R	MEAS R
A	MEAS A
B	MEAS B
Return	
CONVERSION [OFF] → See Conversion menu	
S-PARAMETERS → See S-parameters menu	
ANALYZER TYPE → See Analyzer type menu	
NA S-parameters menu	
Refl: FWD S11 [A/R]	MEAS S11
Trans:FWD S21 [B/R]	MEAS S21
Trans:REV S12 [B/R]	MEAS S12
Refl: REV S22 [A/R]	MEAS S22
INPUT PORTS → See Input port menu	
CONVERSION [OFF] → See Conversion menu	
ANALYZER TYPE → See Analyzer type menu	
Conversion menu	
CONVERSION OFF	CONV OFF
Z:Refl	CONV ZREF
Z:Trans	CONV ZTRA
Y:Refl	CONV YREF
Y:Trans	CONV YTRA
1 S	CONV ONEDS
MORE	
CONVERSION 4xPHASE	CONV MP4
8xPHASE	CONV MP8
16xPHASE	CONV MP16
RETURN	
RETURN	

Front Panle Key	GPIB Command
Analyzer type menu	
NETWORK ANALYZER	NA
SPECTRUM ANALYZER	SA
IMPEDANCE ANALYZER	ZA
RETURN	
Spectrum Analyzer	
SPECTRUM: R	MEAS R
A	MEAS A
SUSCEPT(B)	MEAS B
DETECTION [POSITIVE]	
DETECTION:POS PEAK	DET POS
NEG PEAK	DET NEG
SAMPLE	DET SAM
RETURN	
ANALYZER TYPE → See Analyzer type menu	

Front Panle Key	GPIB Command
Impedance Analyzer	
ZA More menu 1/5	
IMPEDANCE: MAG(|Z|)	MEAS IMAG
PHASE(θ_z)	MEAS IPH
RESIST(R)	MEAS IRE
REACT(X)	MEAS IIM
MORE 1/5	
FIXTURE [NONE] → See Fixture menu	
ANALYZER TYPE → See Analyzer type menu	
ZA More menu 2/5	
ADMITTNCE: MAG(|Y|)	MEAS AMAG
PHASE(θ_y)	MEAS APH
CONDUCT(G)	MEAS ARE
SUSCEPT(B)	MEAS AIM
MORE 2/5	
FIXTURE [NONE] → See Fixture menu	
ANALYZER TYPE → See Analyzer type menu	
ZA More menu 3/5	
REFL.COEF: MAG(|Γ|)	MEAS RCM
PHASE($\theta\Gamma$)	MEAS RCPH
REAL(Γ_x)	MEAS RCR
IMAG(Γ_y)	MEAS RCIM
MORE 3/5	
FIXTURE [NONE] → See Fixture menu	
ANALYZER TYPE → See Analyzer type menu	
ZA More menu 4/5	
CAPCITNCE: PRL(Cp)	MEAS CP
SER(Cs)	MEAS CS
INDUCTNCE: PRL(Lp)	MEAS LP
SER(Ls)	MEAS LS
MORE 4/5	
FIXTURE [NONE] → See Fixture menu	
ANALYZER TYPE → See Analyzer type menu	
ZA More menu 5/5	
RESISTNCE: PRL(Rp)	MEAS RP
SER(Rs)	MEAS RS
D FACTOR(D)	MEAS D
Q FACTOR(Q)	MEAS Q
MORE 5/5	
FIXTURE [NONE] → See Fixture menu	
ANALYZER TYPE → See Analyzer type menu	

Front Panle Key	GPIB Command
Fixture menu	
SELECT FIXTURE	
FIXTURE: NONE	FIXT NONE
16191	FIXT HP16191
16192	FIXT HP16192
16193	FIXT HP16193
16194	FIXT HP16194
USER	FIXT USED
RETURN	
SAVE USER FXTR KIT	SAVUFIXT
MODIFY [NONE]	MODIFIX
DEFINE EXTENTION	FIXE <*numeric*>
LABEL FIXTURE	LABEFIX <*string*>
KIT DONE (MODIFIED)	FIXKDONE
RETURN	

Front Panle Key	GPIB Command
Format	
Network Analyzer	
FORMAT:LOG MAG	FMT LOGM
PHASE	FMT PHAS
DELAY	FMT DELA
SMITH CHART	FMT SMITH
POLAR CHART	FMT POLA
MORE	
FORMAT:LIN MAG	FMT LINM
SWR	FMT SWR
REAL	FMT REAL
IMAGINARY	FMT IMAG
ADMITTANCE CHART	FMT ADMIT
RETURN	
PHASE UNIT [DEG]	PHAU {DEG|RAD}
EXP PHASE ON off	EXPP {ON|OFF}
PHASE UNIT [DEG]	PHAU {DEG|RAD}
EXP PHASE ON off	EXPP {ON|OFF}
Spectrum Analyzer	
FORMAT:SPECTRUM	FMT SPECT
NOISE	FMT NOISE
UNIT: dBm	SAUNIT DBM
dBV	SAUNIT DBV
dBuV	SAUNIT DBUV
WATT	SAUNIT W
VOLT	SAUNIT V
Impedance Analyzer	
FORMAT:LIN Y-AXIS	FMT LINY
LOG Y-AXIS	FMT LOGY
POLAR CHART	FMT POLA
SMITH CHART	FMT SMITH
ADMITTANCE CHART	FMT ADMIT
COMPLEX PLANE	FMT COMP
PHASE UNIT [DEG]	PHAU {DEG|RAD}
EXP PHASE ON off	EXPP {ON|OFF}

Front Panle Key	GPIB Command
Display	
DUAL CHAN on OFF	DUAC {ON|OFF}
DISPLAY[DATA]	DISP DATA
DISPLAY: DATA	DISP DATA
DATA→MEMORY	DATMEM
RETURN	
DATA→MEMORY	DATMEM
OVERLAY TRACES	
DATA→OVERLAY	DATOVE
SELECT PEN COLOR	PEN {1~6}
CLEAR GRAPHICS	GCLEAR
RETURN	
DATA HOLD [OFF]	
HOLD: OFF	DHOLD OFF
MAX	DHOLD MAX
MIN	DHOLD MIN
RETURN	
DATA MATH [DATA]	
DATA MATH: DATA	MATH DATA
DATA+MEM	MATH DPLM
DATA-MEM	MATH DMNM
DATA/MEM	MATH DDVM
DEFAULT GAIN & OFS	DEFGO
OFFSET	
MKR→OFFSET	MKROFS
OFFSET	DATOVAL <*numeric*>
AUX OFFSET	DATAOVAL <*numeric*>
RETURN	
GAIN	DATGAIN <*numeric*>
RETURN	
MORE → Display more menu	

Front Panle Key	GPIB Command
NA/SA Display more menu	
SPLIT DISP ON off	SPLD {ON|OFF}
DISP ALLOC	
ALL INSTRUMENT	DISA ALLI
HALF INSTR HALF BASIC	DISA HIHB
ALL BASIC	DISA ALLB
BASIC STATUS	DISA BASS
RETURN	
TITLE → See Enter text menu	TITL <*string*>
ADJUST DISPLAY → See Adjust display menu	
FREQUENCY BLANK	FREO
RETURN	
ZA Display more menu	
SPLIT DISP ON off	SPLD {ON|OFF}
DISP ALLOC	
ALL INSTRUMENT	DISA ALLI
HALF INSTR HALF BASIC	DISA HIHB
ALL BASIC	DISA ALLB
BASIC STATUS	DISA BASS
RETURN	

Front Panle Key	GPIB Command
EQUIV CKT MENU	
SELECT EQV CKT [A]	DISECIRC {ON|OFF}
CKT A	EQUC CIRA
B	EQUC CIRB
C	EQUC CIRC
D	EQUC CIRD
E	EQUC CIRE
CALCULATE EQV PARAMS	CALECPARA
SIMULATE F-CHRST	SIMFCHAR
RETURN	
DISP PARM on OFF	DISECPARA {ON|OFF}
DEFINE EQV PARAMS	
PARAMETER R1	DEFECR1 <*numeric*>
C1	DEFECC1 <*numeric*>
L1	DEFECL1 <*numeric*>
C0	DEFECC0 <*numeric*>
SIMULATE F-CHRST	SIMFCHAR
RETURN	
CALCULATE EQV PARAMS	CALECPARA
SIMULATE F-CHRST	SIMFCHAR
RETURN	
TITLE → See Enter text menu	TITL <*string*>
ADJUST DISPLAY → See Adjust display menu	
FREQUENCY BLANK	FREO
RETURN	

Front Panle Key	GPIB Command
Adjust display menu	
INTENSITY	INTE <*numeric*>
BACKGROUND INTENSITY	BACI <*numeric*>
MODIFY COLORS	
CH1 DATA → Color adjust menu	COLO CH1D
CH1 MEM/ LIMIT LINE → Color adjust menu	COLO CH1M
CH2 DATA → Color adjust menu	COLO CH2D
CH2 MEM/ LIMIT LINE → Color adjust menu	COLO CH2M
GRATICULE → Color adjust menu	COLO GRAT
WARNING → Color adjust menu	COLO WARN
MORE	
TEXT → Color adjust menu	COLO TEXT
IBASIC → Color adjust menu	COLO IBT
MORE	
PEN 1 → Color adjust menu	COLO PEN1
PEN 2 → Color adjust menu	COLO PEN2
PEN 3 → Color adjust menu	COLO PEN3
PEN 4 → Color adjust menu	COLO PEN4
PEN 5 → Color adjust menu	COLO PEN5
PEN 6 → Color adjust menu	COLO PEN6
RETURN	
RETURN	
RETURN	
DEFAULT COLORS	DEFC
SAVE COLORS	SVCO
RECALL COLORS	RECC
RETURN	
Color adjust menu	
TINT	TINT <*numeric*>
BRIGHTNESS	CBRI <*numeric*>
COLOR	COLOR <*numeric*>
RESET COLOR	RSCO
RETURN	

Front Panle Key	GPIB Command
Scale Ref	
Network Analyzer	
AUTO SCALE	AUTO
SCALE/DIV	SCAL <*numeric*>
REFERENCE POSITION	REFP <*numeric*>
REFERENCE VALUE	REFV <*numeric*>
MKR→REFERENCE	MKRREF
SCALE FOR [DATA]	SCAF {DATA|MEMO}
D&M SCALE [COUPLE]	SCAC {ON|OFF}
ATTENUATOR MENU	
ATTEN R	ATTR <*numeric*>
ATTEN A	ATTA <*numeric*>
ATTEN B	ATTB <*numeric*>
TEST SET ATTEN PT1	ATTP1 <*numeric*>
TEST SET ATTEN PT1	ATTP2 <*numeric*>
RETURN	
Spectrum Analyzer	
PEAK→REFERENCE	PEAKREF
SCALE/DIV	SCAL <*numeric*>
REFERENCE VALUE	REFV <*numeric*>
SCALE FOR [DATA]	SCAF {DATA|MEMO}
D&M SCALE [COUPLE]	SCAC {ON|OFF}
ATTENUATOR MENU	
ATTEN R	ATTR <*numeric*>
ATTEN A	ATTA <*numeric*>
ATTEN B	ATTB <*numeric*>
ATT AUTO on OFF	ATTAUTO {ON|OFF}
RETURN	

Front Panle Key	GPIB Command
Impedance Analyzer	
AUTO SCALE	AUTO
SCALE/DIV	SCAL <*numeric*>
REFERENCE POSITION	REFP <*numeric*>
REFERENCE VALUE	REFV <*numeric*>
MKR→REFERENCE	MKRREF
SCALE FOR [DATA]	SCAF {DATA|MEMO}
D&M SCALE [COUPLE]	SCAC {ON|OFF}
MORE	
AUTO SCALE	AUTO
SCALE/DIV	SCAL <*numeric*>
TOP VALUE	TOPV <*numeric*>
BOTTOM VALUE	BOTV <*numeric*>
MKR→REFERENCE	MKRREF
SCALE FOR [DATA]	SCAF {DATA|MEMO}
D&M SCALE [COUPLE]	SCAC {ON|OFF}
MORE	
AUTO SCALE	AUTO
SCALE/DIV	SCAL <*numeric*>
REFERENCE X VALUE	REFX <*numeric*>
REFERENCE Y VALUE	REFY <*numeric*>
MKR→REFERENCE	MKRREF
SCALE FOR [DATA]	SCAF {DATA|MEMO}
D&M SCALE [COUPLE]	SCAC {ON|OFF}
D&M SCALE [COUPLE]	SCAC {ON|OFF}
MORE	

Front Panle Key	GPIB Command
Bw/Avg	
Network Analyzer	
AVERAGING RESTART	AVERREST
AVERAGING on OFF	AVER {ON|OFF}
AVERAGING FACTOR	AVERFACT <*numeric*>
IF BW auto MAN	BWAUTO {ON|OFF}
IF BW	BW <*numeric*>
AUTO IFBW LIMIT	BWLMT <*numeric*>
GROUP DELY APERTURE	GRODAPER <*numeric*>
Spectrum Analyzer	
AVERAGING RESTART	AVERREST
AVERAGING on OFF	AVER {ON|OFF}
AVERAGING FACTOR	AVERFACT <*numeric*>
RES BW AUTO man	BWAUTO {ON|OFF}
RES BW	BW <*numeric*>
RBW/SPAN RATIO	BWSRAT <*numeric*>
VBW TYPE [LIN]	VBWT {LIN|LOG}
VIDEO BW	VBW <*numeric*>
Impedance Analyzer	
AVERAGING RESTART	AVERREST
AVERAGING on OFF	AVER {ON|OFF}
AVERAGING FACTOR	AVERFACT <*numeric*>
IF BW auto MAN	BWAUTO {ON|OFF}
IF BW	BW <*numeric*>
AUTO IFBW LIMIT	BWLMT <*numeric*>

Front Panle Key	GPIB Command
Cal	
Network Analyzer	
CORRECTION on OFF	CORR {ON|OFF}
CALIBRATE MENU → See NA Cal menu	
RESUME CAL SEQUENCE	RESC
CAL KIT [7mm] → See NA Cal kit menu	
MORE	
PORT EXTENSIONS	
EXTENSIONS on OFF	PORE {ON|OFF}
EXTENSION INPUT R	PORTR <*numeric*>
EXTENSION INPUT A	PORTA <*numeric*>
EXTENSION INPUT B	PORTB <*numeric*>
EXTENSION PORT 1	PORT1 <*numeric*>
EXTENSION PORT 2	PORT2 <*numeric*>
RETURN	
VELOCITY FACTOR	VELOFACT <*numeric*>
SET Z0	SETZ <*numeric*>
ELECTRICAL DELAY MENU	
MKR→DELAY	MKRDELA
ELECTRICAL DELAY	ELED <*numeric*>
PHASE OFFSET	PHAO <*numeric*>
RETURN	
RETURN	
NA Cal menu	
CALIBRATE:NONE	CALI NONE
RESPONSE	CALI RESP
SHORT	STANA
OPEN	STANB
THRU	STANC
DONE	RESPDONE
RESPONSE & ISOL'N	CALI RAI
RESPONSE → See Response Standard menu	RAIRESP
ISOL'N STD	RAIISOL
DONE RESP ISOL'N CAL	RAID
S11 1-PORT	CALI S111
[S11]: OPEN	CLASS11A
SHORT	CLASS11B
LOAD	CLASS11C
DONE: 1-PORT CAL	SAV1
S22 1-PORT	CALI S221
[S22]: OPEN	CLASS22A
SHORT	CLASS22B
LOAD	CLASS22C
DONE: 1-PORT CAL	SAV1

Front Panle Key	GPIB Command
FULL 2-PORT	CALI FUL2
REFLECT'N	REFL
(S11):OPEN	CLASS11A
SHORT	CLASS11B
LOAD	CLASS11C
(S22):OPEN	CLASS22A
SHORT	CLASS22B
LOAD	CLASS22C
REFLECT'N DONE	REFD
TRANS-MISSION	TRAN
FWD.TRANS.THRU	FWDT
FWD.MATCH THRU	FWDM
FWD.TRANS.THRU	REVT
FWD.MATCH THRU	REVM
TRANS. DONE	TRAD
ISOLATION	ISOL
OMIT ISOLATION	OMII
FWD.ISOL'N ISOL'N STD	FWDI
REV.ISOL'N ISOL'N STD	REVI
ISOLATION DONE	ISOD
DONE: 2-PORT CAL	SAV2
ONE PATH 2-PORT	CALI ONE2
REFLECT'N	REFL
OPEN	CLASS11A
SHORT	CLASS11B
LOAD	CLASS11C
REFLECT'N DONE	REFD
TRANS-MISSION	TRAN
FWD.TRANS.THRU	FWDT
FWD.MATCH THRU	FWDM
TRANS. DONE	TRAD
ISOLATION	ISOL
OMIT ISOLATION	OMII
FWD.ISOL'N ISOL'N STD	FWDI
REV.ISOL'N ISOL'N STD	REVI
ISOLATION DONE	ISOD
DONE: 2-PORT CAL	SAV2

Front Panle Key	GPIB Command
Response standard menu	
SHORT	STANA
OPEN	STANB
THRU	STANC
DONE:RESPONSE	DONE
OPEN standard menu	
OPEN[M]	STANA
OPEN[F]	STANB
DONE:OPEN	DONE
DONE:OPEN	DONE
SHORT standard menu	
SHORT[M]	STANA
SHORT[F]	STANB
DONE:SHORT	DONE
LOAD standard menu	
defined std 1	STANA
defined std 2	STANB
defined std 3	STANC
defined std 4	STAND
defined std 5	STANE
defined std 6	STANF
defined std 7	STANG
DONE:LOAD	DONE
THRU standard menu	
defined std 1	STANA
defined std 2	STANB
defined std 3	STANC
defined std 4	STAND
defined std 5	STANE
defined std 6	STANF
defined std 7	STANG
DONE:THRU	DONE

Front Panle Key	GPIB Command
NA Cal kit menu	
CAL KIT:7mm	CALK APC7
3.5mm	CALK APC35
N 50 Ω	CALK N50
N 75 Ω	CALK N75
USER KIT	CALK USED
SAVE USER KIT	SAVEUSEK
MODIFY [7mm]	MODI1
DEFINE STANDARD	
STD NO.1 [SHORT] → See NA/ZA Standard type menu	DEFS 1
STD NO.2 [OPEN] → See NA/ZA Standard type menu	DEFS 2
STD NO.3 [LOAD] → See NA/ZA Standard type menu	DEFS 3
STD NO.4 [DEL/THRU] → See NA/ZA Standard type menu	DEFS 4
STD NO.5 [LOAD] → See NA/ZA Standard type menu	DEFS 5
STD NO.6 [LOAD] → See NA/ZA Standard type menu	DEFS 6
STD NO.7 [SHORT] → See NA/ZA Standard type menu	DEFS 7
STD NO.8 [OPEN] → See NA/ZA Standard type menu	DEFS 8
SPECIFY CLASS	
SPECIFY: S11A	SPECS11A <*numeric*, ... >
S11B	SPECS11B <*numeric*, ... >
S11C	SPECS11C <*numeric*, ... >
SPECIFY: S22A	SPECS22A <*numeric*, ... >
S22B	SPECS22B <*numeric*, ... >
S22C	SPECS22C <*numeric*, ... >
MORE	

Front Panle Key	GPIB Command
SPECIFY:FWD.TRANS.	SPECFWDT <*numeric*, ... >
REV.TRANS.	SPECREVT <*numeric*, ... >
FWD.MATCH	SPECFWDM <*numeric*, ... >
REV.MATCH	SPECREVM <*numeric*, ... >
RESPONSE	SPECRESP <*numeric*, ... >
RESPONSE & ISO'N	SPECRESI <*numeric*, ... >
RETURN	
CLASS DONE (SPEC'D)	CLAD
CLASS DONE (SPEC'D)	CLAD
LABEL CLASS	
LABEL: S11A	LABES11A <*string*>
S11B	LABES11B <*string*>
S11C	LABES11C <*string*>
LABEL: S22A	LABES22A <*string*>
S22B	LABES22B <*string*>
S22C	LABES22C <*string*>
MORE	
LABEL:FWD.TRANS.	LABEFWDT <*string*>
REV.TRANS.	LABEREVT <*string*>
FWD.MATCH	LABEFWDM <*string*>
REV.MATCH	LABEREVM <*string*>
RESPONSE	LABERESP <*string*>
RESPONSE & ISO'N	LABERESI <*string*>
LABEL DONE	
LABEL DONE	
LABEL KIT	LABK <*string*>
KIT DONE (MODIFIED)	KITD
RETURN	

Front Panle Key	GPIB Command
NA/ZA Standard type menu	
STD TYPE: OPEN	STDT OPEN
C0	C0 <*numeric*>
C1	C1 <*numeric*>
C2	C2 <*numeric*>
SPECIFY OFFSET → NA/ZA Specify offset menu	
LABEL STD	LABS <*string*>
STD DONE (DEFINED)	STDD
SHORT	STDT SHOR
SPECIFY OFFSET → NA/ZA Specify offset menu	
LABEL STD	LABS <*string*>
STD DONE (DEFINED)	STDD
LOAD	STDT LOAD
SPECIFY OFFSET → NA/ZA Specify offset menu	
LABEL STD	LABS <*string*>
STD DONE (DEFINED)	STDD
DELAY/THRU	STDT DELA
SPECIFY OFFSET → NA/ZA Specify offset menu	
LABEL STD	LABS <*string*>
STD DONE (DEFINED)	STDD
ARBITRARY IMPEDANCE	STDT ARBI
TERMINAL IMPEDANCE	TERI <*numeric*>
SPECIFY OFFSET → NA/ZA Specify offset menu	
LABEL STD	LABS <*string*>
STD DONE (DEFINED)	STDD
RETURN	
See NA/ZA Specify offset menu	
OFFSET DELAY	OFSD <*numeric*>
OFFSET LOSS	OFSL <*numeric*>
OFFSET Z0	OFSZ <*numeric*>
STD OFFSET DONE	

Front Panle Key	GPIB Command
Impedance Analyzer	
CALIBRATE MENU	CALI IMP
OPEN	CLASIMPA
SHORT	CLASIMPB
LOAD	CLASIMPC
DONE: CORRECTION	SAVIMP
RESUME CAL SEQUENCE	RESC
FIXTURE COMPEN	
COMPEN MENU	COMP
OPEN	COMCA
SHORT	COMCB
LOAD	COMCC
DONE: COMPEN	SAVCOM
RESUME COMP SEQ	RESCOM
OPEN on OFF	COMCDATA {ON|OFF}
SHORT on OFF	COMCDATB {ON|OFF}
LOAD on OFF	COMCDATC {ON|OFF}
RETURN	
CAL KIT [IMP 7mm]	
CAL KIT: IMP 7mm	CALK APC7
3.5mm	CALK APC35
N 50 Ω	CALK N50
N 75 Ω	CALK N75
USER KIT	CALK USED
SAVE USER KIT	SAVEUSEK
MODIFY [IMP 7mm] → CAL KIT Setup menu	MODI1

Front Panle Key	GPIB Command
COMPEN KIT [USER]	
SAVE COMPEN KIT	SAVUCOMK
MODIFY [USER]	MODICOMK
DEFINE STANDARD	
OPEN: CONDUCT(G)	DEFSOPENG <*numeric*>
CAP.(C)	DEFSOPENC <*numeric*>
SHORT: RESIST.(R)	DEFSSHORR <*numeric*>
INDUCT.(L)	DEFSSHORL <*numeric*>
LOAD: RESIST.(R)	DEFSLOADR <*numeric*>
INDUCT.(L)	DEFSLOADL <*numeric*>
STD DONE (DEFINED)	COMSDONE
LABEL KIT	LABECOMK <*string*>
KIT DONE (MODIFIED)	COMKDONE
RETURN	
MORE	
PORT EXTENSION	
EXTENSION ON off	PORE {ON|OFF}
EXTENSION VALUE	PORTZ <*numeric*>
RETURN	
VELOCITY FACTOR	VELOFACT <*numeric*>
SET Z0	SETZ <*numeric*>
RETURN	
Spectrum Analyzer	
LVL CAL DATA R	LVCDTR <*numeric*>
LVL CAL DATA A	LVCDTA <*numeric*>
LVL CAL DATA B	LVCDTB <*numeric*>
INPUT Z	SETZ <*numeric*>

Front Panle Key	GPIB Command
DEFINE STANDARD	
STD NO.1 [SHORT] → See NA/ZA Standard type menu	DEFS 1
STD NO.2 [OPEN] → See NA/ZA Standard type menu	DEFS 2
STD NO.3 [LOAD] → See NA/ZA Standard type menu	DEFS 3
STD NO.4 [DEL/THRU] → See NA/ZA Standard type menu	DEFS 4
STD NO.5 [LOAD] → See NA/ZA Standard type menu	DEFS 5
STD NO.6 [LOAD] → See NA/ZA Standard type menu	DEFS 6
STD NO.7 [SHORT] → See NA/ZA Standard type menu	DEFS 7
STD NO.8 [OPEN] → See NA/ZA Standard type menu	DEFS 8
SPECIFY CLASS	
SPECIFY: IMP A	SPECIMPA <*numeric*>
IMP B	SPECIMPB <*numeric*>
IMP C	SPECIMPC <*numeric*>
CLASS DONE (SPEC'D)	CLAD
LABEL CLASS	
LABEL: IMP A → See Enter text menu	LABEIMPA <*string*>
IMP B → See Enter text menu	LABEIMPB <*string*>
IMP C → See Enter text menu	LABEIMPC <*string*>
LABEL DONE	
LABEL KIT	LABK <*string*>
KIT DONE (MODIFIED)	KITD
RETURN	

Front Panle Key	GPIB Command
Sweep	
Network/Impedance Analyzer	
SWEEP TIME AUTO man	SWETAUTO {ON|OFF}
SWEEP TIME	SWET <*numeric*>
: h:m:s	
RETURN	
NUMBER OF POINTS	POIN <*numeric*>
COUPLED CH ON off	COUC {ON|OFF}
SWEEP TYPE MENU	
SWEEP TYPE:LIN FREQ	SWPT LINF
LOG FREQ	SWPT LOGF
LIST FREQ	SWPT LIST
POWER SWEEP	SWPT POWE
EDIT LIST	EDITLIST
SEGMENT	
EDIT → See NA/ZA segment menu	SEDI [<*numeric*>]
DELETE	SDEL [<*numeric*>]
ADD → See NA/ZA segment menu	SADD [<*numeric*>]
CLEAR LIST	CLEL
LIST DONE	EDITDONE
SEGMENT WAIT	SWAI <*numeric*>
LIST DISP [FREQ BASE]	LISD{FBASE|OBASE}
RETURN	
NA/ZA segment menu	
SEGMENT: START	STAR <*numeric*>
STOP	STOP <*numeric*>
CENTER	CENT <*numeric*>
SPAN	SPAN <*numeric*>
MKR→MENU	
MKR→START	MKRSTAR
MKR→STOP	MKRSTOP
MKR→CENTER	MKRCENT
RETURN	
MORE	
NUMBER of POINTS	POIN <*numeric*>
POWER	POWE <*numeric*>
IF BW	BW <*numeric*>
DC VOLTAGE	DCV <*numeric*>
DC CURRENT	DCI <*numeric*>
RETURN	
SEGMENT QUIT	SQUI
SEGMENT DONE	SDON

Front Panle Key	GPIB Command
Spectrum Analyzer	
SWEEP TIME AUTO man	SWETAUTO {ON|OFF}
SWEEP TIME	SWET <*numeric*>
: h:m:s	
RETURN	
NUMBER OF POINTS	POIN <*numeric*>
SWEEP TYPE MENU	
SWEEP TYPE:LIN FREQ	SWPT LINF
LIST FREQ	SWPT LIST
EDIT LIST	EDITLIST
SEGMENT	
EDIT → See SA segment menu	SEDI [<*numeric*>]
DELETE	SDEL [<*numeric*>]
ADD → See SA segment menu	SADD [<*numeric*>]
CLEAR LIST	CLEL
LIST DONE	EDITDONE
SEGMENT WAIT	SWAI <*numeric*>
LIST DISP [FREQ BASE]	LISD{FBASE|OBASE}
RETURN	
SA segment menu	
SEGMENT: START	STAR <*numeric*>
STOP	STOP <*numeric*>
CENTER	CENT <*numeric*>
SPAN	SPAN <*numeric*>
MKR→MENU	
MKR→START	MKRSTAR
MKR→STOP	MKRSTOP
MKR→CENTER	MKRCENT
RETURN	
MORE	
NUMBER of POINTS	POIN <*numeric*>
RES BW	BW <*numeric*>
DC VOLTAGE	DCV <*numeric*>
DC CURRENT	DCI <*numeric*>
RETURN	
SEGMENT QUIT	SQUI
SEGMENT DONE	SDON

Front Panle Key	GPIB Command
Source	
Network/Impedance Analyzer	
POWER	POWE <*numeric*>
CW FREQ	CWFREQ <*numeric*>
DC SRC [VOLTAGE]	DC??SRC|CTL?? {VOLT|CURR}
DC VOLTAGE	DCV <*numeric*>
DC CURRENT LIMIT	DCILMT <*numeric*>
DC OUT ON off	DCO {ON|OFF}
Spectrum Analyzer	
POWER	POWE <*numeric*>
RF OUT off ON	RFO {ON|OFF}
DC SRC [VOLTAGE]	DCCTL {VOLT|CURR}
DC VOLTAGE	DCV <*numeric*>
DC CURRENT LIMIT	DCILMT <*numeric*>
DC OUT ON off	DCO {ON|OFF}

Front Panle Key	GPIB Command
Trigger	
Network/Impedance Analyzer	
SWEEP:HOLD	HOLD
SINGLE	SING
NUMBER of GROUPS	NUMG <*numeric*>
CONTINUOUS	CONT
TRIGGER:[FREE RUN]	
FREE RUN	TRGS INT
EXTERNAL	TRGS EXT
MANUAL	TRGS MAN
TRIG EVENT[ON SWEEP]	TRGEVE {SWE|POIN}
TRIG PLRTY POS neg	TRGP {POS|NEG}
RETURN	
MEASURE RESTART	REST
Spectrum Analyzer	
SWEEP:HOLD	HOLD
SINGLE	SING
NUMBER of GROUPS	NUMG <*numeric*>
CONTINUOUS	CONT
TRIGGER:[FREE RUN]	
FREE RUN	TRGS INT
EXTERNAL	TRGS EXT
MANUAL	TRGS MAN
GATE [LEVEL]	TRGS GAT
GATE CTL:LEVEL	GATCTL LEV
EDGE	GATCTL EDG
GATE DELAY	GATDLY <*numeric*>
GATE LENGTH	GATLEN <*numeric*>
RETURN	
TRIG PLRTY POS neg	TRGP {POS|NEG}
RETURN	
MEASURE RESTART	REST

Front Panle Key	GPIB Command
Center	CENT <*numeric*>
STEP SIZE AUTO man	CNTSAUTO {ON|OFF}
CENTER STEP SIZE	CNTS <*numeric*>
MKR→CNTR STEP	MKRCSTE
MKRΔ→CNTR STEP	MKRDCSTE
MKR→CENTER	MKRCENT
MKRΔ→CENTER	MKRDCENT
PEAK→CENTER	PEAKCENT
Span	SPAN <*numeric*>
FULL SPAN	FULS
ZERO SPAN	SPAN 0
MKRΔ→SPAN	MKRDSPAN
Start	STAR <*numeric*>
Stop	STOP <*numeric*>

Front Panle Key	GPIB Command
Marker	
Network/Impedance Analyzer	
SUB MKR → See Sub-marker menu	
CLEAR SUB MKR → See Sub-marker menu	
PRESET MKRS	MKR OFF
MKR ON [DATA]	MKRO {DATA|MEMO}
MKR [UNCOUPLE]	MKRCOUP {ON|OFF}
MKR [CONT]	MKRCONT {ON|OFF}
ΔMODE MENU	
ΔMKR	DMKR ON
FIXED ΔMKR	DMKR FIX
TRACKING ΔMKR	DMKR TRAC
ΔMODE OFF	DMKR OFF
ΔMKR SWP PRM	DMKRPRM <*numeric*>
FIXED ΔMKR VALUE	DMKRVAL <*numeric*>
FIXED ΔMKR AUX VALUE	DMKRAUV <*numeric*>
RETURN	
Spectrum Analyzer	
SUB MKR → See Sub-marker menu	
CLEAR SUB MKR → See Sub-marker menu	
PRESET MKRS	MKR OFF
MKR ON [DATA]	MKRO {DATA|MEMO}
MKR [UNCOUPLE]	MKRCOUP {ON|OFF}
MKR [CONT]	MKRCONT {ON|OFF}
ΔMODE MENU	
ΔMKR	DMKR ON
FIXED ΔMKR	DMKR FIX
TRACKING ΔMKR	DMKR TRAC
ΔMODE OFF	DMKR OFF
ΔMKR SWP PRM	DMKRPRM <*numeric*>
FIXED ΔMKR VALUE	DMKRVAL <*numeric*>
RETURN	
Sub-marker menu	
SUB MKR 1	SMKR1 {ON|OFF}
2	SMKR2 {ON|OFF}
3	SMKR3 {ON|OFF}
4	SMKR4 {ON|OFF}
5	SMKR5 {ON|OFF}
6	SMKR6 {ON|OFF}
7	SMKR7 {ON|OFF}
RETURN	

Front Panle Key	GPIB Command
Marker→	
MKR→CENTER	MKRCENT
MKR→START	MKRSTAR
MKR→STOP	MKRSTOP
MKR→REFERENCE	MKRREF
PEAK→CENTER	PEAKCENT
MKR ZOOM	MKRZM
ZOOMING APERTURE	ZMAPER <*numeric*>
MKR→XCH MENU	
MKR→XCH CENTER	XMKRCENT
MKR→XCH START	XMKRSTAR
MKR→XCH STOP	XMKRSTOP
PEAK→XCH CENTER	XPEAKCENT
MKR XCH ZOOM	XMKRZM
ZOOMING APERTURE	ZMAPER <*numeric*>
RETURN	

Front Panle Key	GPIB Command
Search	
Network/Impedance Analyzer	
SEARCH: PEAK → See Peak menu	SEAM PEAK
MAX	SEAM MAX
MIN	SEAM MIN
TARGET	SEAM TARG
TARGET	SEATARG <*numeric*>
SEARCH LEFT	SEAL
SEARCH RIGHT	SEAR
SUB MKR → See Sub-marker menu	
RETURN	
MULTIPLE PEAKS → See Print setup menu	
WIDTHS [OFF]	
SEARCH IN	WIDSIN
SEARCH OUT	WIDSOUT
WIDTHS on OFF	WIDT {ON|OFF}
WIDTH VALUE	WIDV <*numeric*>
MKRVAL/$\sqrt{2}$	WIDVTYPE DIVS2
MKRVAL*$\sqrt{2}$	WIDVTYPE MULS2
MKRVAL/2	WIDVTYPE DIV2
FIXED VALUE	WIDVTYPE FIX
RETURN	
RETURN	
SRCH TRACK on OFF	TRACK {ON|OFF}
SRCH RANGE MENU → See Search range menu	
Peak menu	
PEAK	SEAM PEAK
NEXT PEAK	SEANPK
NEXT PEAK LEFT	SEANPKL
NEXT PEAK RIGHT	SEANPKR
PEAK DEF MENU → See Peak definition menu	
SUB MKR → See Sub-marker menu	
RETURN	
Print setup menu	
SEARCH: PEAKS ALL	SEAM PKSA
PEAKS RIGHT	SEAM PKSR
PEAKS LEFT	SEAM PKSL
PEAK DEF MENU → See Peak definition menu	
SRCH TRACK on OFF	TRACK {ON|OFF}
RETURN	

Front Panle Key	GPIB Command
Search range menu	
PART SRCH on OFF	PARS {ON|OFF}
MKRΔ→SEARCH RNG	SEARSTR
MKR→LEFT RNG	SEARSTRL
MKR→RIGHT RNG	SEARSTRR
RETURN	
NA/ZA Define peak menu	
THRESHOLD on OFF	PKTHRE {ON|OFF}
THRESHOLD VALUE	PKTHVAL <*numeric*>
MKR→THRESHOLD	MKRTHRE
PEAK PLRTY POS neg	PKPOL {POS|NEG}
PEAK DEF: ΔX	PKDLTX <*numeric*>
PEAK DEF: ΔY	PKDLTY <*numeric*>
MKR→PEAK DELTA	MKRPKD
RETURN	
Spectrum Analyzer	
SEARCH: PEAK → See Peak menu	SEAM PEAK
MAX	SEAM MAX
MIN	SEAM MIN
MULTIPLE PEAKS → See Print setup menu	
SGNL TRACK on OFF	SGTRK {ON|OFF}
SRCH TRACK on OFF	TRACK {ON|OFF}
SRCH RANGE MENU → See Search range menu	
SA Define peak menu	
THRESHOLD on OFF	PKTHRE {ON|OFF}
THRESHOLD VALUE	PKTHVAL <*numeric*>
MKR→THRESHOLD	MKRTHRE
PEAK DEF: ΔY	PKDLTY <*numeric*>
RETURN	

Front Panle Key	GPIB Command
Utility	
Network/Impedance Analyzer	
MKR LIST on OFF	MKRL {ON|OFF}
STATISTICS on OFF	MEASTAT {ON|OFF}
MKR TIME on OFF	MKRTIME {ON|OFF}
SMTH/POLAR MENU	
REAL IMAG	CIRF RI
LIN MAG PHASE	CIRF LIN
LOG MAG PHASE	CIRF LOG
R+jX	CIRF RX
G+jB	CIRF GB
SWR PHASE	CIRF SWR
RETURN	
Spectrum Analyzer	
MKR LIST on OFF	MKRL {ON|OFF}
STATISTICS on OFF	MEASTAT {ON|OFF}
MKR TIME on OFF	MKRTIME {ON|OFF}
NOISE FORM on OFF	MKRNOI {ON|OFF}
MKR UNIT MENU	
UNIT:dBm	MKRUNIT DBM
dBV	MKRUNIT DBV
dBuV	MKRUNIT DBUV
WATT	MKRUNIT W
VOLT	MKRUNIT V
RETURN	

Front Panle Key	GPIB Command
System	
PROGRAM MENU	
STOR DEV [DISK]	STOD{DISK|MEMO}
LIMIT MENU → See Lmit test menu	
RECALL MESSAGE	
MORE	
SET CLOCK	
TIME HH:MM:SS	
HOUR	
MIN	
SEC	
ENTER	SETCTIME <*hour*,*minute*,*second*>
CANCEL	
DATE DD/MM/YY	
MONTH	
DAY	
YEAR	
ENTER	SETCDATE <*year*,*month*,*day*>
CANCEL	
DATE MODE:MonDayYear	MONDYEAR
DayMonYear	DAYMYEAR
RETURN	
BEEP DONE ON off	BEEPDONE {ON|OFF}
BEEP WARN on OFF	BEEPWARN {ON|OFF}
FIRMWARE VERSION	DIAG:FREV?
RETURN	

Front Panle Key	GPIB Command
Lmit test menu	
LIMIT LINE on OFF	LIMILINE {ON|OFF}
LIMIT TEST on OFF	LIMITEST {ON|OFF}
BEEP FAIL on OFF	BEEPFAIL {ON|OFF}
EDIT LIMIT LINE	EDITLIML
SEGMENT	LIMSEDI <*numeric*>
EDIT → Limit line entry menu	LIMSEDI
DELETE	LIMSDEL
ADD	LIMSADD
CLEAR LIST	LIMCLEL
DONE	LIMEDONE
LIMIT LINE OFFSETS	
SWP PARAM OFFSET	LIMIPRMO <*numeric*>
AMPLITUDE OFFSET	LIMIAMPO <*numeric*>
MKR→AMP.OFS.	MKRAMPO
RETURN	
RETURN	
Limit line entry menu	
SWP PARAM	LIMPRM <*numeric*>
MKR→SWP PARAM	MKRSWPRM
UPPER LIMIT	LIMU <*numeric*>
LOWER LIMIT	LIML <*numeric*>
DELTA LIMIT	LIMD <*numeric*>
MIDDLE VALUE	LIMM <*numeric*>
MKR→MIDDLE	MKRMIDD
DONE	LIMSDON
Local	
SYSTEM CONTROLLER	
ADDRESS-ABLE ONLY	
SET ADDRESSES	
ADDRESS:INSTR	
ADDRESS:CONTROLLER	ADDRCONT <*numeric*>
RETURN	
Preset	PRES

Front Panle Key	GPIB Command
Copy	
PRINT [STANDARD]	PRINALL
COPY ABORT	COPA
COPY SKEY ON off	PRSOFT {ON|OFF}
COPY TIME on OFF	COPT {ON|OFF}
PRINT SETUP → See Print setup menu	
ORIENT [[PORTRAIT]	LANDSCAPE {ON|OFF}
FORM FEED ON off	FORMFEED {ON|OFF}
MORE → See Copy more menu	
NA Copy more menu	
LIST VALUES → See Screen menu	LISV
OPERATING PARAMETERS → See Screen menu	OPEP
CAL KIT DEFINITION → See Copy cal kit menu	
LIST SWEEP TABLE—-> copy list sweep meu	
LIMIT TEST TABLE → See Copy limit test menu	
RETURN	
SA Copy more menu	
LIST VALUES → See Screen menu	LISV
OPERATING PARAMETERS → See Screen menu	OPEP
LIST SWEEP TABLE → See Copy list sweep menu	
LIMIT TEST TABLE → See Copy limit test menu	
RETURN	
ZA Copy more menu	
LIST VALUES → See Screen menu	LISV
OPERATING PARAMETERS → See Screen menu	OPEP
CAL KIT DEFINITION → See Copy cal kit menu	
COMPEN KIT DEFINITION → See Copy compen kit menu	
LIST SWEEP TABLE → See Copy list sweep menu	
LIMIT TEST TABLE → See Copy limit test menu	
RETURN	

Front Panle Key	GPIB Command
Print setup menu	
PRINT STANDARD	PRIS
COLOR	PRIC
PRINT COLOR [FIXED]	PRIC{FIXE|VARI}
DPI	DPI <*numeric*>
TOP MARGIN	TMARG <*numeric*>
LEFT MARGIN	LMARG <*numeric*>
DEFAULT SETUP	DFLT
RETURN	
Screen menu	
PRINT [STANDARD]	PRINALL
COPY ABORT	COPA
COPY TIME on OFF	COPT {ON|OFF}
PRINT SETUP → See Print setup menu	
NEXT PAGE	NEXP
PREV PAGE	PREP
RESTORE DISPLAY	RESD
Copy cal kit menu	
STANDARD DEFINITION → See Copy standard no. menu	
CLASS ASSIGNMENT	CALCASSI
RETURN	
Copy limit test menu	
DISPLAY LIST	DISLLIST
DISP MODE: UPR & LWR	DISMAMP UL
MID & DLT	DISMAMP MD
RETURN	
Copy list sweep menu	
DISPLAY LIST	DISL
DISP MODE: ST & SP	DISMPRM STSP
CTR & SPAN	DISMPRM CTSP
RETURN	
Copy standard no. menu	
STD NO.1	CALS 1
STD NO.2	CALS 2
STD NO.3	CALS 3
STD NO.4	CALS 4
STD NO.5	CALS 5
STD NO.6	CALS 6
STD NO.7	CALS 7
STD NO.8	CALS 8

Front Panle Key	GPIB Command
Save	
STATE	SAVDSTA <*string*>
DATA ONLY	
SAVE BINARY	SAVDDAT <*string*>
SAVE ASCII	SAVDASC <*string*>
DEFINE SAVE DATA → See Define save data menu	
STOR DEV [DISK]	STOD{DISK|MEMO}
RETURN	
GRAPHICS	SAVDTIF <*string*>
RE-SAVE FILE	RESAVD <*string*>
BACK UP MEMO DISK	STORMDISK
FILE UTILITIES	
PURGE FILE	PURG <*string*>
CREATE DIRECTORY	CRED <*string*>
CHANGE DIRECTORY	CHAD <*string*>
COPY FILE	FILC <*string*>,<*string*>,<*string*>,<*string*>
INITIALIZE DISK	
INIT DISK: YES	INID
FORMAT [LIF]	DISF {LIF|DOS}
STOR DEV [DISK]	STOD{DISK|MEMO}
RETURN	
STOR DEV [DISK]	STOD{DISK|MEMO}
RETURN	
STOR DEV [DISK]	STOD{DISK|MEMO}
Define save data menu	
RAW on OFF	SAVRAW {ON|OFF}
CAL on OFF	SAVCAL {ON|OFF}
DATA on OFF	SAVDAT {ON|OFF}
MEM on OFF	SAVMEM {ON|OFF}
DATA TRACE on OFF	SAVDTRC {ON|OFF}
MEM TRACE on OFF	SAVMTRC {ON|OFF}
RETURN	
Recall	
file name	
STOR DEV [DISK]	STOD{DISK|MEMO}

12

Manual Changes

Introduction

This appendix contains the information required to adapt this manual to earlier versions or configurations of the analyzer than the current printing date of this manual. The information in this manual applies directly to the 4395A Network/Spectrum Analyzer serial number prefix listed on the title page of this manual.

Manual Changes

To adapt this manual to your 4395A, see Table 12-1 and Table 12-2, and make all the manual changes listed opposite your instrument's serial number and firmware version.

Instruments manufactured after the printing of this manual may be different from those documented in this manual. Later instrument versions will be documented in a manual changes supplement that will accompany the manual shipped with that instrument. If your instrument's serial number is not listed on the title page of this manual or in Table 12-1, it may be documented in a *yellow MANUAL CHANGES* supplement.

In additions to change information, the supplement may contain information for correcting errors (Errata) in the manual. To keep this manual as current and accurate as possible, Agilent Technologies recommends that you periodically request the latest *MANUAL CHANGES* supplement.

For information concerning serial number prefixes not listed on the title page or in the *MANUAL CHANGE* supplement, contact the nearest Agilent Technologies office.

Turn on the line switch or execute the `*IDN?` command by GPIB to confirm the firmware version.

An example of `*IDN?` command execution:

```
10 ALLOCATE A$[50]
20 OUTPUT 717;"*IDN?"
30 ENTER 717;A$
40 PRINT A$
50 END
```

Table 12-1. Manual Changes by Serial Number

Serial Prefix or Number	Make Manual Changes

Table 12-2. Manual Changes by Firmware Version

Version	Make Manual Changes
REV 1.00	Change 1

Serial Number

Agilent Technologies uses a two-part, nine-character serial number that is stamped on the serial number plate (see Figure 12-1) attached to the rear panel. The first four digits and the letter are the serial prefix and the last five digits are the suffix.

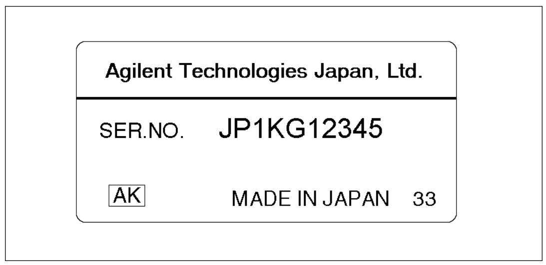


Figure 12-1. Serial Number Plate

Change 1

The firmware revision 1.00 does not support the following commands. Please delete the descriptions about these commands in this manual.

```
CLOSE
CWD?
FNAME?
FNUM?
FSIZE?
READ?
ROPEN
WOPEN
WRITE
```

A

Commands in Entry Block A

ADDRCONT␣<*numeric*>

Sets the GPIB address the analyzer will use to communicate with the external controller. (`ADDRESS: CONTROLLER` under (Local))

Parameter	Range	Unit
<*numeric*>	0 to 30	

- Query Response

  {*numeric*} <new line><^END>

ANAOCH{1|2}

Selects channel 1 or 2 for analysis.

This command has effects on the commands listed below:

```
CIRF
DMKR {ON|FIX|TRAC}
DMKR OFF
DMKRAUV
DMKRPRM
DMKRVAL
INPUDTRC
LIMIAMPO
LIMILINE
LIMIPRMO
LIMITEST
MEASTAT
MKR OFF
MKR ON
MKRAMPO
MKRAUV?
MKRDISP
MKRL
MKRNOI
MKRO
MKRP
MKRPKD
MKRPRM
MKRTHRE
MKRTIME
MKRVAL?
OUTPDMKR?
OUTPDTRC?
OUTPDTRCP?
OUTPFAIP?
OUTPLIMF?
OUTPLIML?
OUTPLIMM?
OUTPMKR?
OUTPMSTA?
OUTPMTRC?
OUTPMTRCP?
OUTPMWID?
OUTPSMKR{1-7}?
PARS
PKDLTX
PKDLTY
PKPOL
PKTHRE
PKTHVAL
PRSMKRS
SAUNIT
SEAL
SEAM
SEANPK
SEANPKL
SEANPKR
SEAR
SEARSTR
SEARSTRL
SEARSTRR
SEATARG
SMKR{1-7}
SMKRAUV{1-7}?
SMKRP{1-7}
SMKRPRM{1-7}
SMKRVAL{1-7}?
TRACK
WIDSIN
WIDSOUT
WIDT
WIDV
WIDVTYPE
```

ANAOCH{1|2}

■ Query Response

Parameter	Description
OFF or 0	Analysis for channel 1 (or 2) is off.
ON or 1	Analysis for channel 1 (or 2) is on.

ATT[R|A|B]␣<*numeric*>[DB]

Controls the attenuation at port R, A or B.

Parameter	Range	Unit
<*numeric*>	0, 10, 20, 30, 40, 50	dB

■ Query Response

{*numeric*} <new line><^END>

ATTAUTO␣{OFF|ON|0|1}

Sets the automatic and manual spectrum analyzer input attenuator of the S input. (Spectrum analyzer only) (ATTEN AUTO man under Scale Ref)

When the automatic attenuator is selected, the value selected ensures that the level meets the following equation:

$$\text{Attnuator value(dB)} = \text{(Reference value)} - \text{(20dB)}$$

Parameter	Description
OFF or 0	Manual attenuator
ON or 1	Automatic attenuator

■ Query Response

{0|1} <new line><^END>

ATTP{1|2}␣<*numeric*>[DB]

Controls the attenuation at port 1 or port 2 of an S-parameter Test Set connected to the analyzer. (Network analyzer only) (`ATTENUATOR PORT 1` under (Source))

Parameter	Range	Unit
<*numeric*>	0, 10, 20, 30, 40, 50, 60, 70	dB

- Query Response

{*numeric*} <new line><^END>

AUTO

Brings the trace data, defined by the `SCAF` command, in view on the display. (Network and impedance analyzers only) (`AUTO SCALE` under (Scale Ref); No query)

AVER␣{OFF|ON|0|1}

Turns the averaging function ON or OFF for the active channel. (`AVERAGING ON off` under (Bw/Avg))

Parameter	Description
`OFF` or `0`	Averaging function OFF
`ON` or `1`	Averaging function ON

- Query Response

{0|1} <new line><^END>

AVERFACT␣<*numeric*>

Makes the averaging factor for the active function. (AVERAGING FACTOR under Bw/Avg)

Parameter	Range	Unit
<*numeric*>	1 to 999	

- Query Response

  {*numeric*} <new line><^END>

AVERREST

Resets the sweep-to-sweep averaging and restarts the sweep count at 1 at the beginning of the next sweep. (AVERAGING RESTART under Bw/Avg; No query)

B

Commands in Entry Block B

BACI␣<*numeric*>[PCT]

Sets the background intensity of the display as a percent of the white level.
(BACKGROUND INTENSITY under Display)

Parameter	Range	Unit
<*numeric*>	0 to 100	%

- Query Response

{*numeric*} <new line><^END>

BEEPDONE␣{OFF|ON|0|1}

Sets an annunciator that sounds to indicate completion of certain operations such as calibration or instrument state save. (BEEP DONE ON off under System)

Parameter	Description
OFF or 0	Operation completion beeper OFF
ON or 1	Operation completion beeper ON

- Query Response

{0|1} <new line><^END>

BEEPFAIL␣{OFF|ON|0|1}

Turns the limit fail beeper ON or OFF. When the limit testing is ON and the fail beeper is ON, a beep is emitted each time a limit test is performed and a failure is detected. (BEEP FAIL ON off under (System))

Parameter	Description
OFF or 0	Limit fail beeper OFF
ON or 1	Limit fail beeper ON

- Query Response

  {0|1} <new line><^END>

BEEPWARN␣{OFF|ON|0|1}

Sets the warning annunciator. When the annunciator is ON, it sounds a warning when a cautionary message is displayed. (BEEP WARN ON off under (System))

Parameter	Description
OFF or 0	Warning beeper OFF
ON or 1	Warning beeper ON

- Query Response

  {0|1} <new line><^END>

BLIGHT␣{OFF|ON|0|1}

Sets backlighting the LCD screen ON or OFF.

Parameter	Description
OFF or 0	Backlighting OFF
ON or 1	Backlighting ON

- Query Response

  {0|1} <new line><^END>

BOTV␣<*numeric*>

Defines the bottom border of the display and adjusts the scale value. (BOTTOM VALUE under (Scale Ref); Network and impedance analyzers only)

Parameter	Range	Unit
<*numeric*>	-1×10^9 to 1×10^9	y-axis unit

- Query Response

{*numeric*} <new line><^END>

BW␣<*numeric*>[HZ]

Sets the bandwidth value for IF bandwidth reduction, or sets the IF bandwidth of the list sweep table. This command is valid only if the automatic IF bandwidth setting is off by BWAUTO OFF command. (Network and impedance analyzers) (IF BW under (Bw/Avg), or IF BW under (Sweep))

Sets the bandwidth value for the resolution bandwidth reduction, or sets the resolution bandwidth of the list sweep table. This command is valid only if the automatic resolution bandwidth setting is off by BWAUTO OFF command. (Spectrum analyzer) (RES BW under (Bw/Avg), or RES BW under (Sweep))

Parameter	Range	Unit
<*numeric*>	2, 10, 30, 100, 300, 1000 (=1k), 3000 (=3k), 10000 (=10k), 30000 (=30k) (Network and impedance analyzers)	
	1, 3, 10, 30, 100, 300, 1000 (=1k), 3000 (=3k), 10000 (=10k), 30000 (=30k), 100000 (=100k), 300000 (=300k), 1000000 (=1M), 3000000 (=3M) (span>0 in spectrum analyzer)	Hz
	3000 (=3k), 5000 (=5k), 10000 (=10k), 20000 (=20k), 40000 (=40k), 100000 (=100k), 200000 (=200k), 400000 (=400k), 800000 (=800k), 1500000 (=1.5M), 3000000 (=3M), 5000000 (=5M) (span=0 in spectrum analyzer)	Hz

- Query Response

{*numeric*} <new line><^END>

BWAUTO␣{OFF|ON|0|1}

When log frequency sweeping mode is selected, sets either the automatic or manual IF bandwidth ON. (Network and impedance analyzers) (`RES BW AUTO man` under `Bw/Avg`)

When linear frequency sweeping mode is selected, sets either the automatic or manual resolution bandwidth ON. (Spectrum analyzer only) (`RES BW AUTO man` under `Bw/Avg`)

Parameter	Description
`OFF` or `0`	Manually sets the IF bandwidth or resolution bandwidth
`ON` or `1`	Automatically sets the IF bandwidth or resolution bandwidth

- Query Response

  {0|1} <new line><^END>

BWLMT <*numeric*>

Sets the limit value for IF BW(IF bandwidth). (Network and impedance analyzers only)

This command is valid only if the automatic IF bandwidth setting is on by `BWAUTO ON` command.

Parameter	Range	Unit
<*numeric*>	2, 10, 30, 100, 300, 1000 (=1k), 3000 (=3k), 10000 (=10k), 30000 (=30k)	??Hz??

BWSRAT␣<*numeric*>[PCT]

Sets the RBW/SPAN (resolution bandwidth/span) ratio that specifies the resolution bandwidth in the AUTO mode. (Spectrum analyzer only) (`RBW/SPAN RATIO` under `Bw/Avg`)

This command is valid only if the automatic resolution bandwidth setting is on by `BWAUTO ON` command.

Parameter	Range	Unit
<*numeric*>	0.01 to 10 (relative to span)	%

- Query Response

  {*numeric*} <new line><^END>

C

Commands in Entry Block C (*C included)

C0␣<*numeric*>

Enters the C_0 term, which is the constant term of the capacitance model equation. (Network and impedance analyzers only) (C0 under Cal; No query)

Parameter	Range	Unit
<*numeric*>	-10000 to 10000 (x 10^{-15})	F

- Query Response

{*numeric*} <new line><^END>

C1␣<*numeric*>

Enters the C_1 term, which is the constant term of the capacitance model equation. (Network and impedance analyzers only) (C1 under Cal; No query)

Parameter	Range	Unit
<*numeric*>	-10000 to 10000 (x 10^{-27})	F/Hz

- Query Response

{*numeric*} <new line><^END>

C2␣<*numeric*>

Enters the C_2 term, which is the constant term of the capacitance model equation. (Network and impedance analyzers only) (`C2` under (Cal); No query)

Parameter	Range	Unit
<*numeric*>	-10000 to 10000 (x 10^{-36})	F/Hz^2

- Query Response

{*numeric*} <new line><^END>

CALCASSI

Shows the tabular listing of the calibration kit class assignment. (`CLASS ASSIGNMENT` under (Copy); No query)

CALECPARA

Calculates and displays the equivalent circuit parameters. (`CALCULATE EQV PARAMS` under (Display); No query; Impedance analyzer only)

CALI␣{NONE|RESP|RAI|S111|S221|FUL2|ONE2|IMP}

Selects the measurement calibration type. (Network and impedance analyzers only) (`CALIBRATE:NONE`, `RESPONSE`, `RESPONSE & ISOL'N`, `S11 1-PORT`, `S22 1-PORT`, `FULL 2-PORT`, `ONE PATH 2-PORT` under (Cal) of network analyzer mode or `CALIBRATE MENU` under (Cal) of impedance analyzer mode.)

Parameter	Description
NONE	No calibration (Network and impedance analyzers only)[1]
RESP	Response measurement calibration (Network analyzer only)
RAI	Response and isolation measurement calibration (Network analyzer only)
S111	1-Port measurement calibration at port 1 (Network analyzer only)
S221	1-Port measurement calibration at port 2 (Network analyzer only)
FUL2	Full 2-Port measurement calibration (Network analyzer only)
ONE2	One-path 2-Port measurement calibration (Network analyzer only)
IMP	Calibration of the impedance analyzer mode. (Impedance analyzer only)

1 Error correction will be turned off and all the coefficients in the array for error correction can no longer be used.

- Query Response

{NONE|RESP|RAI|S111|S221|FUL2|ONE2|IMP} <new line><^END>

CALK␣{APC7|APC35|N50|N75|USED}

Selects one of the default calibration kits available for different connector types. (Network and impedance analyzers only) (`CAL KIT:7mm`, `3.5mm`, `N 50 Ω`, `N 75 Ω`, or `USER KIT` under (Cal))

Parameter	Description
APC7	7 mm
APC35	3.5 mm
N50	Type-N 50 Ω
N75	Type-N 75 Ω
USED	User-defined

- Query Response

{APC7|APC35|N50|N75|USED} <new line><^END>

CALS␣<*numeric*>

Provides the tabular listing of the standard definitions. (Network and impedance analyzers only) (`STD NO.1` to `STD NO.8` under (Copy); No query)

Parameter	Range	Unit
<*numeric*>	1 to 8	

CBRI␣<*numeric*>[PCT]

Adjusts the brightness of the color being modified. (`BRIGHTNESS` under (Display))

Parameter	Range	Unit
<*numeric*>	0 to 100	%

- Query Response

{*numeric*} <new line><^END>

CENT␣<*numeric*>[HZ|DBM]

Defines the center value of the sweep range, or the center value of the segment to be edited in the list sweep table. (Center, or CENTER under Sweep)

Parameter	Range	Unit
<*numeric*>	0 to 510000000	Hz (frequency)
	10 to 510000000 (Network and impedance analyzers, when editing a list sweep table)	Hz (frequency)
	−50 to +15 (Network and impedance analyzers)	dBm (power)

- Query Response

{*numeric*} <new line><^END>

CHAD␣<*string*>

Changes the current directory on a DOS-formatted disk. (Specify a directory name you wish to change to) (CHANGE DIRECTORY under Save; No query)

Parameter	Description
<*string*>	Directory path

CHAN{1|2}

Selects channel 1 or 2 as the active channel. (Chan 1 or Chan 2)

- Query Response

{0|1} <new line><^END>

CIN

Set the port C, a 24-bit I/O port, as the input port.

CIRF␣{RI|LIN|LOG|RX|GB|SWR}

Selects format to readout the value of a Smith, polar, or admittance chart using markers. (Network and impedance analyzers only) (REAL IMAG, LIN MAG PHASE, LOG MAG PHASE, R+jX, G+jB, SWR PHASE under (Utility))

Parameter	Description
RI	Real and imaginary form
LIN	Linear magnitude and phase form
LOG	Log magnitude and phase form
RX	Complex impedance form (R+jX)
GB	Complex admittance form (G+jB)
SWR	SWR and phase form

- Query Response

  {RI|LIN|LOG|RX|GB|SWR} <new line><^END>

CLAD

Completes the class assignment and stores it. (Network and impedance analyzers only) (CLASS DONE (SPE'D) under (Cal); No query)

CLASIMP{A|B|C}

Selects and acquires the impedance calibration classes. (CALIBRATION OPEN, SHORT, or LOAD under (Cal), respectively; No query; Impedance analyzer only)

The order in which you acquire the OPEN, SHORT, and LOAD is changeable. You can suspend a calibration sequence and do a different operation, and then resume the calibration sequence.

CLASS11{A|B|C}

Selects port 1 (S11) calibration standard class: S11A (open), S11B (short), or S11C (load). (Network analyzer only) ([S11] : OPEN , SHORT , LOAD under Cal; No query)

CLASS22{A|B|C}

Selects port 2 (S22) calibration standard class: S22A (open), S22B (short), or S22C (load), and starts calibration. (Network analyzer only) ([S22] : OPEN , SHORT , LOAD under Cal; No query)

CLEL

Clears the entire list. (CLEAR LIST under Sweep; No query)

CLES

Provides the same function as *CLS command. (No query)

*CLS

Clears the error queues, the Status Byte register, the Operational Status register, the Standard Event Status register, and the Event Status register B (Instrument Event Status register). (No query)

CLOSE

Returns a file, which has been read/write-enabled using the ROPEN command or WOPEN command, to access-disabled status. If this command is executed before reading process using the READ? command completes, an error occurs.

Generally, this command is used in combination with the ROPEN command and READ? command or the WOPEN command and the WRITE command, as shown in Figure R-2. (No query)

CNTS␣<*numeric*>[HZ|DBM]

Changes the step size for the center frequency function. (`CENTER STEP SIZE` under (Center))

Parameter	Range	Unit
<*numeric*>	0.001 to 510000000	Hz
	0.001 to 65	dBm

- Query Response

{*numeric*} <new line><^END>

CNTSAUTO␣{OFF|ON|0|1}

Sets CENTER step policy. (`STEP SIZE AUTO man` under (Center))

Parameter	Description
`OFF` or `0`	Linear step
`ON` or `1`	1-2-5 step

- Query Response

{0|1} <new line><^END>

COLO␣<*parameter*>

Specifies the display element to change color. (`CH1 DATA`, `CH1 MEM LIMIT LN`, `CH2 DATA`, `CH2 MEM LIMIT LN`, `GRATICULE`, `IBASIC`, `PEN 1`, `PEN 2`, `PEN 3`, `PEN 4`, `PEN 5`, `PEN 6`, `TEXT`, `WARNING` under (Display))

COLO␣<*parameter*>

Parameter	Description
CH1D	Channel 1 data trace
CH1M	Channel 1 memory and limit lines
CH2D	Channel 2 data trace
CH2M	Channel 2 memory and limit lines
GRAT	Graticule and a portion of softkey text
WARN	Warning annotation
TEXT	All the non-data text
IBT	Text on the BASIC screen
PEN1	Pen 1
PEN2	Pen 2
PEN3	Pen 3
PEN4	Pen 4
PEN5	Pen 5
PEN6	Pen 6

- Query Response

  {CH1D|CH1M|CH2D|CH2M|WARN|TEXT|GRAT|IBT|PEN1|PEN2|PEN3|PEN4|PEN5|PEN6} <new line><^END>

  - Query Response

    {*numeric (hue)*}, {*numeric (sat)*}, {*numeric (lum)*} <new line><^END>

COLOR␣<*numeric*>[PCT]

Adjusts the degree of whiteness of the color being modified. (COLOR under Display)

Parameter	Range	Unit
<*numeric*>	0 to 100	%

- Query Response

  {*numeric*} <new line><^END>

COMC{A|B|C}

Measures the standards for the fixture compensation. (COMPEN OPEN, SHORT, or LOAD under (Cal); No query; Impedance analyzer only)

Parameter	Description
A	Measures OPEN.
B	Measures SHORT.
C	Measures LOAD.

COMCDAT{A|B|C}␣{OFF|ON|0|1}

Sets the OPEN, SHORT, and LOAD fixture compensation ON or OFF. (OPEN ON off, SHORT ON off, or LOAD ON off under (Cal); Impedance analyzer only)

Parameter	Description
A	Uses OPEN compensation data.
B	Uses SHORT compensation data
C	Uses LOAD compensation data
ON or 1	Turns on the selected data.
OFF or 0	Turns off the selected data.

- Query Response

{1|0} <new line><^END>

COMKDONE

Complete modifying the fixture compensation kit. (KIT DONE (MODIFIED) under (Cal) COMPEN KIT [USER] MODIFY [USER]; Impedance analyzer only)

COMP

Call the fixture compensation menu. You need send this command before sending COMC. (COMPEN MENU under (Cal) FIXTURE COMPEN ; No query; Impedance analyzer only)

COMS

Displays the fixture compensation definition on the display. (COMPEN KIT DEFINITION under (Copy); No query; Impedance analyzer only)

COMSDONE

Complete defining the standard for the fixture compensation kit. (STD DONE (DEFINED) under (Cal) COMPEN KIT [USER] MODIFY [USER] ; Impedance analyzer only)

CONT

Triggers sweep automatically and continuously and the trace is updated with each sweep. (CONTINUOUS under (Trig))

- Query Response

  {0|1} <new line><^END>

CONV␣<*parameter*>

Selects the measurement data conversion setting (impedance, admittance, or multiple phase). (Network analyzer only) (OFF , Z:Refl , Z:Trans , Y:Refl , Y:Trans , 1/S , 4xPHASE , 8xPHASE , 16xPHASE under (Meas))

Parameter	Description
OFF	Conversion OFF
ZREF	Z: reflection
ZTRA	Z: transmission
YREF	Y: reflection
YTRA	Y: transmission
ONEDS	Reciprocal (1/S)
MP4	Multiply phase by 4
MP8	Multiply phase by 8
MP16	Multiply phase by 16

- Query Response

{OFF|ZREF|ZTRA|YREF|YTRA|ONEDS|MP4|MP8|MP16} <new line><^END>

COPA

Aborts a print in progress. (`COPY ABORT` under (Copy); No query)

COPT␣{OFF|ON|0|1}

Turns printing time and date (the time stamp function) ON or OFF. (`COPY TIME ON off` under (Copy))

Parameter	Description
`OFF` or `0`	Time stamp function OFF
`ON` or `1`	Time stamp function ON

- Query Response

{0|1} <new line><^END>

CORR␣{OFF|ON|0|1}

Turns error correction ON or OFF. (Network and impedance analyzers only) (`CORRECTION ON off` under (Cal), This softkey is Network analyzer only)

Parameter	Description
`OFF` or `0`	Error correction OFF
`ON` or `1`	Error correction ON

- Query Response

{0|1} <new line><^END>

COUC␣{OFF|ON|0|1}

Sets the channel coupling of sweep parameter values. (Between network or between impedance analyzers only) (COUPLED CH ON off under [Sweep])

Parameter	Description
OFF or 0	Channel coupling OFF
ON or 1	Channel coupling ON

- Query Response

  {0|1} <new line><^END>

COUT

Sets the port C, a 24-bit I/O port, as the output port.

CRED␣<*string*>

Create a new directory in a DOS format disk. (CREATE DIRECTORY under [Save]; No query)

Parameter	Description
<*string*>	Up to 8 characters for directory name (and up to 3 characters for extension)

CWFREQ␣<*numeric*>[HZ]

Sets the frequency for power sweep. (Network and impedance analyzers only) (CWFREQ under [Source])

Parameter	Range	Unit
<*numeric*>	0 to 510000000 (0.001 resolution)	Hz

- Query Response

  {*numeric*} <new line><^END>

CWD?

Returns the name of the current directory. (Query only)

- Query Response

  {*string*} <new line><^END>

D

Commands in Entry Block D

DATAOVAL␣<*numeric*>

Defines the imaginary part of the offset value when using the Smith, Polar, and admittance chart format. (AUX OFFSET under Display)

Parameter	Range	Unit
<*numeric*>	−500000 to 500000	

- Query Response

{*numeric*} <new line><^END>

DATGAIN␣<*numeric*>

Defines the gain value of the data math function. (GAIN under Display)

Parameter	Range	Unit
<*numeric*>	−100 to 100 (0 excluded)	

- Query Response

{*numeric*} <new line><^END>

DATMEM

Stores the current active measurement data in the memory trace of the active channel. Also use this command to store data in the data trace to the memory trace. (DATA→MEMORY under Display; No query)

DATOVAL␣<*numeric*>

Defines the offset value. When using Smith, Polar, and admittance chart format, this command defines the real part of the offset value. (OFFSET under Display)

Parameter	Range	Unit
<*numeric*>	−500000 to 500000	

- Query Response

{*numeric*} <new line><^END>

DATOVE

Copy the current data trace to an overlay trace on the LCD. (DATA→OVERLAY under Display; No query)

DAYMYEAR

Changes the displayed date to the "day:month:year" format. (DayMonYear under System)

- Query Response

{0|1} <new line><^END>

Parameter	Description
0	"month:day:year" format
1	"day:month:year" format

DCCTL␣{VOLT|CURR}

Sets the DC OUT port to control either voltage or current.

Parameter	Description
VOLT	The DC OUT port controls voltage (voltage control mode).
CURR	The DC OUT port controls current (current control mode).

DCI␣<*numeric*>[A]

Sets DC current for the DC OUT port. When editing a segment in a list sweep table, sets DC current for the segment currently being edited.

Parameter	Range	Unit
<*numeric*>	$-0.000002(=20\mu)$ to 0.1	A
	$0.000002(=20\mu)$ to 0.1 (20μA resolution)	

DCO␣{OFF|ON|0|1}

Turns the DC OUT port ON or OFF.

Parameter	Description
`OFF` or `0`	The DC OUT port is turned OFF.
`ON` or `1`	The DC OUT port is turned ON.

- Query Response

  {0|1} <new line><^END>

DCV␣<*numeric*>[V]

Sets DC voltage for the DC OUT port. When editing a segment in a list sweep table, sets DC voltage for the segment currently being edited.

Parameter	Range	Unit
<*numeric*>	−40 to 40 (1mV resolution)	V

DEFC

Returns all the color settings back to the factory-set default values. (`DEFAULT COLORS` under (Display); No query)

DEFEC{R1|C1|L1|C0}␣<*numeric*>

Defines the specified equivalent circuit parameter for simulation. (PARAMETER R1, C1, L1, C0 under [Display]; Impedance analyzer only)

Parameter	Description
R1	Parameter R_1
C1	Parameter C_1
L1	Parameter L_1
C0	Parameter C_0

Parameter	Range	Unit
<*numeric*>	-1×10^{18} to 1×10^{18}	F(C0,C1) H(L1) OHM(R1)

- Query Response

<*numeric*> <new line><^END>

DEFGO

Returns the gain and offset values back to the default values (gain=1, offset=0). (DEFAULT GAIN & OFS under [Display]; No query)

DEFS␣{1-8}

Defines the number of the calibration standards to be modified. (Network and impedance analyzers only) (DEFINE STANDARD under [Cal]; No query)

Parameter	Description
1	Standard no. 1 (SHORT)
2	Standard no. 2 (OPEN)
3	Standard no. 3 (LOAD)
4	Standard no. 4 (DEL/THRU)
5	Standard no. 5 (LOAD)
6	Standard no. 6 (LOAD)
7	Standard no. 7 (SHORT)
8	Standard no. 8 (OPEN)

- Query Response

{ STAN{1-8} } <new line><^END>

DEFSLOAD{R|L}␣<*numeric*>

Defines the LOAD standard by entering resistance and reactance value. (LOAD: RESIST.(R), INDUCT.(L) under (Cal) CAL KIT []. Impedance analyzer only)

Parameter	Description
R	Resistance value of the LOAD fixture compensation standard.
L	Inductance value of the LOAD fixture compensation standard.

Parameter	Range	Unit
<*numeric*>	-1×10^{6} to 1×10^{6}	Ω (R)
<*numeric*>	-1×10^{6} to 1×10^{6}	H (L)

- Query Response

{*numeric*} <new line><^END>

DEFSOPEN{G|C}␣<*numeric*>

Defines the OPEN standard by entering conductance and capacitance value. (OPEN: CONDUCT.(G), CAP.(C) under (Cal) CAL KIT []. Impedance analyzer only)

Parameter	Description
G	Conductance value of the OPEN fixture compensation standard.
C	Capacitance value of the OPEN fixture compensation standard.

Parameter	Range	Unit
<*numeric*>	-1×10^{6} to 1×10^{6}	S (G)
<*numeric*>	-1×10^{6} to 1×10^{6}	fF (C)

- Query Response

{*numeric*} <new line><^END>

DEFSSHOR{R|L}␣<*numeric*>

Defines the SHORT calibration standard by entering resistance and inductance value. (`SHORT: RESIST.(R)`, `INDUCT.(L)` under (Cal) `CAL KIT []`. Impedance analyzer only)

Parameter	Description
R	Resistance value of the SHORT fixture compensation standard.
L	Inductance value of the SHORT fixture compensation standard.

Parameter	Range	Unit
<*numeric*>	-1×10^6 to 1×10^6	Ω (R)
<*numeric*>	-1×10^6 to 1×10^6	H (L) (L)

- Query Response

{*numeric*} <new line><^END>

DET␣{POS|NEG|SAM}

Selects the detection mode for the active channel. (Spectrum analyzer only) (`POS PEAK`, `NEG PEAK`, `SAMPLE` under (Meas))

Parameter	Description
POS	Positive Detection
NEG	Negative Detection
SAM	Sample Detection

- Query Response

{POS|NEG|SAM} <new line><^END>

DHOLD␣␣{OFF|MAX|MIN}

Selects the data hold operation. When the format is changed, the value held is initiated. (HOLD: OFF, MAX, MIN under (Display))

Parameter	Description
OFF	Data hold operation is turned off
MAX	Maximum data hold
MIN	Minimum data hold

- Query Response

{OFF|MAX|MIN} <new line><^END>

DIN

Sets the port D, a 24-bit I/O port, as the input port.

DISA␣{ALLI|HIHB|ALLB|BASS}

Selects the display allocation mode. (DISP ALLOC [ALL INST] under (Display))

Parameter	Description
ALLI	All instrument
HIHB	Half instrument and half HP Instrument BASIC
ALLB	All HP Instrument BASIC
BASS	HP Instrument BASIC status

- Query Response

{ALLI|HIHB|ALLB|BASS} <new line><^END>

DISECIRC {OFF|ON|0|1}

Displays the equivalent circuit models. (SELECT EQV CKT [A] under (Display); Impedance analyzer only)

Parameter	Description
OFF or 0	Turns off the equivalent circuit parameter display.
ON or 1	Turns on the equivalent circuit parameter display.

- Query Response

{0|1} <new line><^END>

DISECPARA {OFF|ON|0|1}

Displays the equivalent circuit parameters. (DISP EQV PARM [ON] or [OFF] under (Display); Impedance analyzer only)

Parameter	Description
OFF or 0	Turns off the equivalent circuit parameter display.
ON or 1	Turns on the equivalent circuit parameter display.

- Query Response

{0|1} <new line><^END>

DISF␣{DOS|LIF}

Selects the disk format (LIF or DOS) to be used when initializing a new disk. (FORMAT [ ] under (Save); No equivalent SCPI command)

Parameter	Description
DOS	DOS format
LIF	Logical Interchange format

- Query Response

{DOS|LIF} <new line><^END>

DISL

Displays the list sweep table on the display. (DISPLAY LIST under (Copy); No query)

DISLLIST

Displays the limit testing table on the display. (DISPLAY LIST under (Copy); No query)

DISMAMP␣{UL|MD}

Selects the amplitude format to display the limit testing table to list on the screen. (DISP MODE: UPR & LWR, MID & DLT under (Copy); No equivalent SCPI command)

Parameter	Description
UL	Upper and lower format
MD	Middle and delta format

- Query Response

  {UL|MD} <new line><^END>

DISMPRM␣{STSP|CTSP}

Selects the sweep parameter range format to display the list sweep table on the screen. (DISP MODE: ST & SP, CTR & SPAN under (Copy))

Parameter	Description
STSP	Start and stop format
CTSP	Center and span format

- Query Response

  {STSP|CTSP} <new line><^END>

DISP␣{DATA|MEMO|DATM}

Selects the display trace type. (`DISPLAY: DATA`, `MEMORY`, `DATA and MEMORY` under (Display))

Parameter	Description
DATA	Current data trace
MEMO	Memory trace
DATM	Current data and memory traces

■ Query Response

{DATA|MEMO|DATM} <new line><^END>

DMKR␣{ON|FIX|TRAC|OFF}

Displays the Δmarker (ON, FIX, TRAC) at the point of the marker and the marker mode changes to the Δmode. Erases (OFF) the Δmarker and the Δmode is turned off. (`ΔMKR`, `FIXED ΔMKR`, `TRACKING ΔMKR`, `ΔMODE OFF` under (Marker))

Parameter	Description
ON	Puts the Δmarker on a current position of the marker.
FIX	Sets a user-specified fixed reference marker.
TRAC	Puts a Δmarker at the present active marker position and turns on the tracking Δmarker.
OFF	Turns off the Δmode.

■ Query Response

{ON|FIX|TRAC|OFF} <new line><^END>

DMKRAUV␣<*numeric*>

Sets the auxiliary amplitude value of the fixed Δmarker. This command is used with a polar, Smith, or admittance chart. (Network and impedance analyzers only) (`ΔMKR AUX VALUE` under (Marker))

Parameter	Range	Unit
<*numeric*>	−1000000000 to 1000000000	

■ Query Response

{*numeric*} <new line><^END>

DMKRPRM␣<*numeric*>[HZ|DBM]

Sets the sweep parameter value of the Δmarker. (ΔMKR SWP PRM under (Marker))

Parameter	Range	Unit
<*numeric*>	Start value to stop value	Hz (frequency) dBm (power)

■ Query Response

{*numeric*} <new line><^END>

DMKRVAL␣<*numeric*>

Sets the amplitude value of the fixed Δmarker. (ΔMKR VALUE under (Marker))

Parameter	Range	Format
<*numeric*>	−1000000000 to 1000000000	

■ Query Response

{*numeric*} <new line><^END>

DONE

Completes the measurement of the selected response/isolation calibration. (Network and impedance analyzers only) (DONE: RESPONSE under (Cal). When Type-N calkits or user calkit,DONE: OPEN DONE: SHORT under (Cal).No query)

DOUT

Sets the port D, a 24-bit I/O port, as the output port.

DSKEY

Disables the front panel keys and the rotary knob. To enable the keys and knob again, send the ENKEY command. (No query)

DUAC␣{OFF|ON|0|1}

Selects the display of both measurement channels or the active channel only.
(DUAL CHAN ON off under (Display))

Parameter	Description
OFF or 0	Active channel only
ON or 1	Both channels

■ Query Response

{0|1} <new line><^END>

E

Commands in Entry Block E (∗E included)

EDITDONE

Completes editing the sweep list.(LIST DONE under Sweep; No query)

EDITLIML

Begins editing the limit line table. (EDIT LIMIT LINE under System; No query)

EDITLIST

Begins editing the frequency sweep list. (EDIT LIST under Sweep; No query)

ELED␣<*numeric*>[S|MS|US|NS|PS|FS]

Adjusts the electrical delay to balance the phase of the DUT. (Network analyzer only) (ELECTRICAL DELAY under Scale Ref)

Parameter	Range	Unit
<*numeric*>	-0.01 to 0.01 (1×10^{-12} resolution)	sec

- Query Response

  {*numeric*} <new line><^END>

ENKEY

Re-enables the front panel keys and the rotary knob that have been disabled by the DSKEY command. (No query)

EQUC␣CIR{A|B|C|D|E}

Selects the equivalent circuit. (SELECT EQV CKT [] under Display. Impedance analyzer only)

Parameter	Description
CIRA	For coils with high core loss.
CIRB	For coils and resistance.
CIRC	For high-value resistors.
CIRD	For capacitors.
CIRE	For resonators.

- Query Response

CIR{A|B|C|D|E}<new line><^END>

ESB?

Outputs the Event Status register B (Instrument Event Status register) value. (Query only)

- Query Response

{*numeric*} <new line><^END>

*ESE␣<*numeric*>

Sets the enable bits of the Standard Event Status Register.

Parameter	Description
<*numeric*>	0 to 255 (decimal expression of enable bits of the operation status register)

- Query Response

{*numeric*} <new line><^END>

ESNB␣<*numeric*>

Enables the bits of Event Status register B (Instrument Event Status register).

Parameter	Range	Unit
<*numeric*>	Decimal expression of the contents of the register, 0 to 65535 (=$2^{16}-1$)	

- Query Response

{*numeric*} <new line><^END>

*ESR?

Returns the contents of the Standard Event Status Register. (Query only)

- Query Response

{*numeric*} <new line><^END>

EXPP␣{OFF|ON|0|1}

Turns on and off the expanded phase display (displaying phase in -180° to 180° without wrapping around). (EXP PHASE on OFF under (Format); Impedance analyzer only)

Parameter	Description
OFF or 0	Turns off the expanded phase display.
ON or 1	Turns on the expanded phase display.

- Query Response

{0|1} <new line><^END>

F

Commands in Entry Block F

FILC␣<*string1*>,<*string2*>,<*string3*>,<*string4*>

Copies files. (COPY FILE under Save; No query)

Parameter	Description
<*string1*>	Source file name
<*string2*>	Source device name ("DISK" or "MEMORY")[1]
<*string3*>	Destination file name
<*string4*>	Destination device name ("DISK" or "MEMORY")

1 "DISK" for the built-in flexible disk drive; "MEMORY" for the RAM disk memory.

FIXE␣<*numeric*>

Sets the electrical length of the fixture. (DEFINE EXTENSION under Meas; Impedance analyzer only.)

Parameter	Range	Unit
<*numeric*>	−0.01 to 0.01 (1×10^{-12} resolution)	m

- Query Response

  {*numeric*} <new line><^END>

FIXKDONE

Terminates the user fixture setting. (DONE under Meas; No query; Impedance analyzer only.)

FIXT␣{NONE|HP16191|HP16192|HP16193|HP16194|USED}

Specifies the fixture in use in order to select which electrical length (recorded in the analyzer) is to be used. (FIXTURE:NONE , HP16191 , HP16192 , HP16193 , HP16194 , USED under (Meas) SELECT FIXTURE ; Impedance analyzer only.)

- Query Response

  {NONE|HP16191|HP16192|HP16193|HP16194|USED}<new line><^END>

FMT␣<*parameter*>

Selects the display format. (FORMAT: LOG MAG , PHASE , DELAY , SMITH [Re Im] , POLAR [Re Im] , LIN MAG , SWR , FORMAT: REAL , IMAGINARY , EXPANDED PHASE , ADMITTANCE [Re Im] , FORMAT: SPECTRUM , NOISE , LIN Y-AXIS , LOG Y-AXIS , COPLEX PLANE under (Format))

Parameter	Description
LOGM	Log magnitude format (Network analyzer only)
PHAS	Phase format (Network analyzer only)
DELA	Delay format (Network analyzer only)
LINM	Linear magnitude format (Network analyzer only)
SWR	SWR format (Network analyzer only)
REAL	Real format (Network analyzer only)
IMAG	Imaginary format (Network analyzer only)
SMITH	Smith chart format (Network and impedance analyzers only)
POLA	Polar chart format (Network and impedance analyzers only)
ADMIT	Admittance Smith chart (Network and impedance analyzers only)
SPECT	Spectrum measurement (Spectrum analyzer only)
NOISE	Noise level measurement (Spectrum analyzer only)
LINY	Linear Y-axis measurement (Impedance analyzer only)
LOGY	Log Y-axis measurement (Impedance analyzer only)
COMP	Complex plane measurement (Impedance analyzer only)
EXPP	Expanded phase format (Network analyzer only)

- Query Response

  {LOGM|PHAS|DELA|LINM|SWR|REAL|IMAG|SMITH|POLA|EXPP|ADMIT|SPECT|NOISE|LINY|
  <new line><^END>

FNAME?␣<*numeric*>

Returns the file name corresponding to a specified number in the current directory. To each file, a number is assigned from 1 to "the number of the files" in alphabetical order. Use the `FNUM?` command to verify the number of the files in the current directory. (Query only)

Parameter	Description	Range
<*numeric*>	Specified file No.	1 to "the number of the files in the current directory"

- Query Response

  {*string*} <new line><^END>

FNUM?

Returns the number of the files in the current directory. (Query only)

- Query Response

  {*numeric*} <new line><^END>

FORM2

Sets the IEEE 32-bit floating point format to transfer trace data via GPIB. (No query)

FORM3

Sets the IEEE 64-bit floating point format to transfer the trace data via GPIB. (No query)

FORM4

Sets the ASCII transfer format to transfer the trace data via GPIB. (No query)

FORM5

Sets MS-DOS format to transfer the trace data via GPIB. (No query)

FREO

Blanks the displayed frequency notation for security purposes. Frequency notation cannot be restored except by sending the `:SYSTem:PRESet` or `*RST` command, or by turning the power OFF and ON. (`FREQUENCY BLANK` under Display)

- Query Response

  {0|1} <new line><^END>

FSIZE?␣<*string*>

Returns the size of a specified file in bytes. If the file does not exist, this command returns -1. (Query only)

Parameter	Description
<*string*>	File name of up to 12 characters including its extension (for the LIF format, up to 10 characters)

- Query Response

  {*numeric*} <new line><^END>

FULS

Sets the SPAN to the maximum range. This command is valid for all sweeping modes except list sweep. (`FULL SPAN` under Span; No query)

FWDI

Measures S_{21} isolation. (Network analyzer only) (FWD ISOL'N ISOL'N STD under Cal; No query)

FWDM

Measures S_{11} load match. (Network analyzer only) (FWD. MATCH THRU under Cal; No query)

FWDT

Measures S_{21} frequency response. (Network analyzer only) (FWD. TRANS. THRU under Cal; No query)

G

Commands in Entry Block G

GATCTL␣{LEV|EDG}

Specifies the gate trigger mode. (Spectrum analyzer only) (Option 1D6 only) (GATE CTL: LEVEL, EDGE under (Trigger))

Parameter	Description
LEV	Level gate trigger mode
EDG	Edge gate trigger mode

- Query Response

  {LEV|EDG} <new line><^END>

GATDLY␣<*numeric*>[S]

Sets the gate delay. (Spectrum analyzer only) (Option 1D6 only) (GATE DELAY under (Trigger))

Parameter	Range	Unit
<*numeric*>	0.0000008 (=0.8μ) to 3.2	sec

- Query Response

  {*numeric*} <new line><^END>

GATLEN␣<*numeric*>[S]

Sets the gate length. (Spectrum analyzer only) (Option 1D6 only) (GATE LENGTH under Trigger)

Parameter	Range	Unit
<*numeric*>	0.000002 (=2 μ) to 3.2	sec

- Query Response

{*numeric*} <new line><^END>

GCLEAR

Erases the image of data trace displayed using DATOVE command. (CLEAR GRAPHICS under Display; No query)

GRODAPER␣<*numeric*>[PCT]

Sets the aperture for the group delay measurement as a percentage of the span. (Network analyzer only) (GROUP DELY APERTURE under Bw/Avg)

Parameter	Range	Unit
<*numeric*>	0.25 to 20 (of span) (simple command)	%

- Query Response

{*numeric*} <new line><^END>

H

Commands in Entry Block H

HOLD

Freezes the data trace on the display. the analyzer stops sweeping and taking data. (SWEEP: HOLD under Trigger)

- Query Response

{0|1} <new line><^END>

Parameter	Description
0	Sweeping (not hold mode)
1	Hold mode

Commands in Entry Block I (∗I included)

∗IDN?

Returns the analyzer's ID.

- Query Response

  {*Agilent Technologies*} {*4395A*} {*serial no.*} {*firmware rev.*} <new line><^END>

INID

Initializes the disk in the flexible disk drive or the RAM disk memory. (INITIALIZE under Save; No query)

Floppy disks can be initialized in the 2HD format only.

INP8IO?

Inputs data from the 4-bit parallel input to the analyzer, and outputs the data to a controller. (Query only)

- Query Response

  {*numeric*} <new line><^END>

INPT?

Returns the pulse input status for the INPUT1, a 24-bit I/O port. (Query only)

- Query Response

  {0|1} <new line><^END>

Parameter	Description
0	No pulse input
1	Pulse input[1]

1 Once 1 is returned, the value is reset, 0 will be returned until there is another pulse input.

INPUCALC{**1-12**}␣<*numeric (1)*>,<*numeric (2)*>, ... ,<*numeric (n)*>

Stores the measurement calibration error coefficient set of real/imaginary pairs input via GPIB into the analyzer's memory. The command definition changes to INPUCALC{1-3} when used in the impedance analyzer. (Network and impedance analyzers only; No query)

Parameter	Description
<*numeric*>	Complex number (Data format: real, imaginary)

INPUCALK␣<*block*>

Stores the calibration kit data transmitted by the `OUTPCALK?` command. (Network and impedance analyzers only) (No query)

Parameter	Description
<*block*>	Block data (Data format: 4395A internal format (714 bytes of binary data))

INPUCOMC{**1**|**2**|**3**}␣␣<*numeric (1)*>,<*numeric (2)*>, ... ,<*numeric (n)*>

Inputs data into the fixture compensation coefficient arrays. (No query; Impedance analyzer only.)

The analyzer handles a reflection coefficient data for the intermediate processing. Thus, the fixture compensation is performed for the reflection coefficient as follows:

$$\Gamma = \frac{\Gamma_M - A}{B \times (\Gamma_M - A) + C}$$

Where,

A, B, and C Fixture compensation coefficients. (complex)

Γ_M Measured reflection data. (converted from V and I.)

Γ Corrected reflection data.

By using this command, you can change the contents of the fixture compensation coefficient arrays.

Parameter	Description
1	coefficient A
2	coefficient B
3	coefficient C
<*numeric*>	Complex number (Data format: real, imaginary)

INPUDATA␣<*numeric (1)*>,<*numeric (2)*>, ... ,<*numeric (n)*>

Inputs the error corrected data. (No query)

Parameter	Description
<*numeric*>	Complex number (Data format: real, imaginary) for the Network analyzer Real number for the Spectrum analyzer

INPUDTRC␣<*numeric (1)*>,<*numeric (2)*>, ... ,<*numeric (n)*>

Inputs data to DATA TRACE memory. (No query)

Parameter	Description
<*numeric*>	Complex number (Data format: real, imaginary) for the Network analyzer Real number for the Spectrum analyzer

INPURAW{1-4}␣<*numeric (1)*>,<*numeric (2)*>, ... ,<*numeric (n)*>

Inputs raw data. The command definition changes to INPURAW{1} when used in the impedance analyzer and spectrum analyzer. (No query)

Parameter	Description
<*numeric*>	Complex number (Data format: real, imaginary) for the Network analyzer Real number for the Spectrum analyzer

INTE␣<*numeric*>[PCT]

Sets the display intensity as a percent of the brightest setting. (INTENSITY under Display)

Parameter	Range	Unit
<*value*>	0 to 100 (simple command)	%

■ Query Response

{*numeric*} <new line><^END>

ISOD

Completes isolation calibration. The error coefficients are calculated and stored. (Network analyzer only) (ISOLATION DONE under (Cal); No query)

ISOL

Starts the isolation calibration. (Network analyzer only) (ISOLATION under (Cal); No query)

K

Commands in Entry Block K

KEY␣<*numeric*>

Sends the key code for a key or a softkey on the front panel. This is equivalent to actually pressing a key. See Figure K-1 for key codes.

Parameter	Description
<*numeric*>	0 to 52

- Query Response

{*numeric*} <new line><^END>

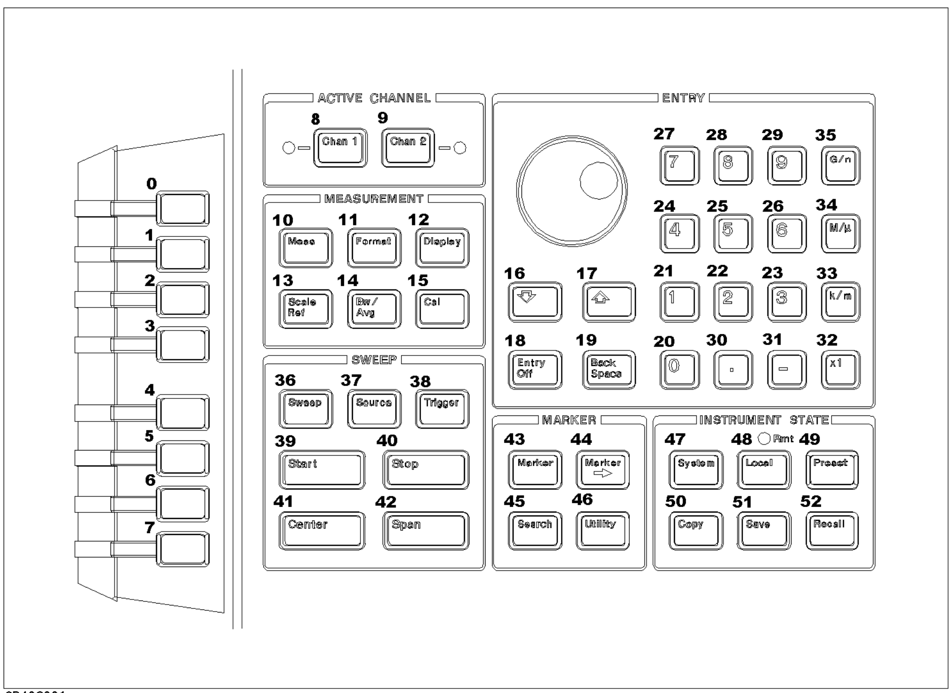


Figure K-1. Key Codes

KITD

Completes the procedure to define user cal kit. (Network analyzer only)
(KIT DONE (MODIFIED) under [Cal]; No query)

L

Commands in Entry Block L

LABECOMK␣<*string*>

Modifies the label of user defined fixture compensation kit. (Impedance analyzer only) (LABEL KIT under Cal; No query)

Parameter	Description
<*string*>	Up to 8 characters.

- Query Response

{*string*} <new line><^END>

LABEFIX␣<*string*>

Modifies the label of user defined test fixture. (LABEL FIXTURE under Meas FIXTURE [] ; Impedance analyzer only.)

Parameter	Description
<*string*>	Up to ten characters.

- Query Response

<*string*><new line><^END>

LABEFWD{T|M}␣<*string*>

Defines the label for the forward transmission (THRU) or the forward match (THRU) calibration. (Network analyzer only) (`LABEL: FWD. TRANS.`, `FWD. MATCH` under (Cal); No query)

Parameter	Description
<*string*>	Up to eight characters.

- Query Response

  {*string*} <new line><^END>

LABEIMP{A|B|C}␣<*string*>

Defines the label for the first class, second class, or the third class required for an impedance measurement calibration. (Impedance analyzer only) (`LABEL CLASS` under (Cal); No query)

Parameter	Description
<*string*>	Up to eight characters.

- Query Response

  {*string*} <new line><^END>

LABERES{P|I}␣<*string*>

Defines the label for the response, or the response and isolation calibration. (Network analyzer only) (`RESPONSE`, `RESPONSE & ISOL'N` under (Cal); No query)

Parameter	Description
<*string*>	Up to eight characters.

- Query Response

  {*string*} <new line><^END>

LABEREV{T|M}␣<*string*>

Defines the label for reverse transmission (THRU) or the reverse match (THRU) calibration. (Network analyzer only) (`REV.TRANS.`, `REV.MATCH` under (Cal); No query)

Parameter	Description
<*string*>	Up to eight characters.

- Query Response

{*string*} <new line><^END>

LABES11{A|B|C}␣<*string*>

Defines the label for the first class, the second class, or the third class required for an S_{11} 1-port calibration. (Network analyzer only) (`LABEL: S11A`, `S11B`, `S11C` under (Cal); No query)

Parameter	Description
<*string*>	Up to eight characters.

- Query Response

{*string*} <new line><^END>

LABES22{A|B|C}␣<*string*>

Defines the label for the first class, the second class, or the third class required for an S_{22} 1-port calibration. (Network analyzer only) (`LABEL: S22A`, `S22B`, `S22C` under (Cal); No query)

Parameter	Description
<*string*>	Up to eight characters.

- Query Response

{*string*} <new line><^END>

LABK␣<*string*>

Defines a label for a new calibration kit. (Network and impedance analyzers only) (LABEL KIT under Cal; No query)

Parameter	Description
<*string*>	Up to eight characters.

- Query Response

{*string*} <new line><^END>

LABS␣<*string*>

Defines a label for the standard. (Network and impedance analyzers only) (LABEL STD under Cal; No query)

Parameter	Description
<*string*>	Up to ten characters.

- Query Response

{*string*} <new line><^END>

LIMCLEL

Clears all segments in the limit line. (CLEAR LIST YES under System; No query)

LIMD␣<*numeric*>

Sets the limits an equal amount above and below a specified middle value, instead of setting upper and lower limits separately. (DELTA LIMITS under System)

This command is valid while editing a segment in a limit line table.

Parameter	Range	Unit
<*numeric*>	0 to 2000000000	

- Query Response

{*numeric*} <new line><^END>

LIMEDONE

Completes editing the limit table. (`DONE` under (System); No query)

LIMIAMPO␣<*numeric*>

Adds or subtracts an offset in amplitude value. (`AMPLITUDE OFFSET` under (System))

Parameter	Range	Unit
<*numeric*>	−1000000000 to 1000000000	

- Query Response

{*numeric*} <new line><^END>

LIMILINE␣{OFF|ON|0|1}

Sets limit lines ON or OFF. (`LIMIT LINE ON off` under (System))

Parameter	Description
`OFF` or `0`	Limit lines OFF
`ON` or `1`	Limit lines ON

- Query Response

{0|1} <new line><^END>

LIMIPRMO␣<*numeric*>

Adds or subtracts an offset from the sweep parameter value. (`SWP PARAM OFFSET` under (System))

Parameter	Range	Unit
<*numeric*>	-12×10^{9} (=−1G) to $1.\times10^{9}$ (=1G) (0.001 resolution)	

- Query Response

{*numeric*} <new line><^END>

LIMITEST␣{OFF|ON|0|1}

Sets the limit testing ON or OFF. (LIMIT TEST ON off under System)

This command uses the limit line currently set in the 4395A, regardless it is displayed or not.

Parameter	Description
OFF or 0	Limit testing OFF
ON or 1	Limit testing ON

■ Query Response

{*numeric*} <new line><^END>

LIML␣<*numeric*>

Sets the lower limit value for the segment. (LOWER LIMIT under System)

This command is valid while editing a segment in a limit line table.

Parameter	Range	Unit
<*numeric*>	−1000000000 to 1000000000	

■ Query Response

{*numeric*} <new line><^END>

LIMM␣<*numeric*>

Sets the midpoint for delta limits. (MIDDLE VALUE under System)

This command is valid while editing a segment in a limit line table.

Parameter	Range	Unit
<*numeric*>	−1000000000 to 1000000000	

■ Query Response

{*numeric*} <new line><^END>

LIMPRM␣<*numeric*>[HZ|DBM]

Sets the starting sweep parameter value of a segment, using entry block controls. (SWP PARAM under System)

This command is valid while editing a segment in a limit line table.

Parameter	Range	Unit
<*numeric*>	-12×10^9 (=−1G) to $1. \times 10^9$ (=1G) (0.001 resolution)	

- Query Response

{*numeric*} <new line><^END>

LIMSADD

Adds a new segment to the end of the limit list. (ADD under System; No query)

This command initiates the editing a segment in a limit line table.

LIMSDEL

Deletes a limit testing segment. (DELETE under System; No query)

LIMSDON

Terminates a limit line segment definition. (DONE under System; No query)

All the segments in a limit line are sorted based on the sweep parameters then stored in the build-in memory in the 4395A. Presetting the 4395A by PRES will not clear the limit line in the memory.

LIMSEDI␣[<*numeric*>]

Specifies which limit segment in the table to edit. When you want to define or modify the values of the specified segment, you do not have to enter <*numeric*> (the segment number). (SEGMENT, EDIT under System)

This command initiates the editing a segment in a limit line table.

Parameter	Description
<*numeric*>	Segment number, 1 to 18.

- Query Response

{*numeric*} <new line><^END>

LIMU␣<*numeric*>

Sets the upper limit value for a limit testing segment. (UPPER LIMIT under System)

This command is valid while editing a segment in a limit line table.

Parameter	Range	Unit
<*numeric*>	−1000000000 to 1000000000	

- Query Response

{*numeric*} <new line><^END>

LISDFBASE

Displays contents of the trace sorting based on the sweep parameter, when the sweeping mode is set to the frequency list sweep. This commands works exclusively with the command `LISDOBASE`.

Parameter	Description
`OFF` or `0`	Displays contents of the trace at an even intervals sorting based on the order of acquisition.
`ON` or `1`	Displays contents of the trace sorting based on the sweep parameter.

- Query Response

{0|1} <new line><^END>

LISDOBASE

Displays contents of the trace at an even intervals sorting based on the order of acquisition, when the sweeping mode is set to the frequency list sweep. This commands works exclusively with the command `LISDFBASE`.

Parameter	Description
`OFF` or `0`	Displays contents of the trace sorting based on the sweep parameter.
`ON` or `1`	Displays contents of the trace at an even intervals sorting based on the order of acquisition.

- Query Response

  {0|1} <new line><^END>

LISV

Displays a tabular listing of all the measured data points and their current values. (`LIST VALUES` under (Copy); No query)

LVCDT[A|B|R]␣<*numeric*>[DB]

Sets the level calibration data for the port R, A, or B (adds an offset value to the measured value). (Spectrum analyzer only) (`LVL CAL DATA` under (Cal))

Parameter	Range	Unit
<*numeric*>	−10 to 10 (0.1 resolution)	dB

M

Commands in Entry Block M

MATH␣{DATA|DDVM|DMNM|DPLM}

Sets the trace math operation. (DATA MATH: DATA , DATA-MEM , DATA+MEM , DATA/MEM under DATA MATH [ ] under Display)

Parameter	Description
DATA	Turns OFF all data math functions.
DMNM	Subtracts the memory from the data.
DPLM	Adds the memory to the data.
DDVM	Divides the data by the memory.

- Query Response

{DATA|DMNM|DPLM|DDVM} <new line><^END>

MAXDCI␣<*numeric*>[A]

Sets the upper limit of the current for the DC OUT port when it is in the voltage control mode. This command also defines the upper limit value for altered polarity for the specified value is regarded as an absolute value.

Parameter	Range	Unit
<*numeric*>	0.00001 (=100μ) to 0.1 (0 V<$\|V_{DC}\|$<25 V)	A
	0.00001 (=100μ) to 0.02 (25 V<$\|V_{DC}\|$<40 V)	
	(20μA resolution)	

$|V_{DC}|$ denotes the output voltage of the DC OUT port.

MAXDCV␣<*numeric*>[V]

Sets the upper limit of the voltage for the DC OUT port when it is in the current control mode. This command also defines the upper limit value for altered polarity for the specified value is regarded as an absolute value.

Parameter	Range	Unit
<*numeric*>	0.01 to 40 (0 A<$\lvert I_{DC}\rvert$<20 mA) 0.01 to 25 (20 mA<$\lvert I_{DC}\rvert$<100 mA) (1mV resolution)	A

$\lvert I_{DC}\rvert$ denotes the output current of the DC OUT port.

MEAS␣<*parameter*>

Selects the parameters or inputs to be measured. (NETWORK: A/R, B/R, R, A, B, Refl: FWD S11 [A/R], Trans:FWD S21 [B/R], Trans:REV S12 [A/R], Refl: REV S22 [B/R], SPECTRUM: S, R, A, B, IMPEDANCE: MAG(|Z|), PHASE(θ_Z), RESIST(R), REACT(X), ADMITTNCE:MAG(|Y|), PHASE(θ_Y), CONDUCT(G), SUSCEPT(B), REFL.COEF:MAX(|Γ|), PHASE(θ_Γ)}}, REAL(Γ_X), IMAG(Γ_Y), CAPCITNCE:PRL(Cp), SER(Cs), INDUCTNCE:PRL(Lp) SER(Ls), RESISTNCE:PRL(Rp), SER(Rs), D FACTOR(D), Q FACTOR(Q) under (Meas))

Parameter	Description
AR	A/R measurement (Network analyzer only)
BR	B/R measurement (Network analyzer only)
R	R measurement (Both Network and Spectrum analyzers)
A	A measurement (Both Network and Spectrum analyzers)
B	B measurement (Both Network and Spectrum analyzers)
S11	S11 measurement (Network analyzer only)
S12	S12 measurement (Network analyzer only)
S21	S21 measurement (Network analyzer only)
S22	S22 measurement (Network analyzer only)
S	S measurement (Spectrum analyzer only)
IMAG	$\|Z\|$ measurement (Impedance analyzer only)
IPH	θ_z (Impedance analyzer only)
IRE	R (Impedance analyzer only)
IIM	X (Impedance analyzer only)
AMAG	$\|Y\|$ (Impedance analyzer only)
APH	θ_y (Impedance analyzer only)
ARE	G (Impedance analyzer only)
AIM	B (Impedance analyzer only)
RCM	$\|\Gamma\|$ (Impedance analyzer only)
RCPH	θ_Γ (Impedance analyzer only)
RCR	Γ_x (Impedance analyzer only)
RCIM	Γ_y (Impedance analyzer only)
CP	Parallel Capacitance, C_p (Impedance analyzer only)
CS	Series Capacitance, C_s (Impedance analyzer only)
LP	Parallel Inductance, L_p (Impedance analyzer only)
LS	Series Inductance, L_s (Impedance analyzer only)
D	Dissipation Factor, D (Impedance analyzer only)
Q	Quality Factor, Q (Impedance analyzer only)
RP	Parallel Resistance, R_p (Impedance analyzer only)
RS	Series Resistance, R_s (Impedance analyzer only)

■ Query Response

{AR|RB|R|A|B|S11|S12|S21|S22|IMAG|IPH|IRE|IIM|AMAG|APH|ARE|AIM|RCM|RCPH|RCM|RCPH|RCR|RCIM| CP|CS|LP|LS|D|Q|RP|RS} <new line><^END>

MEASTAT␣{OFF|ON|0|1}

Calculates the mean, standard deviation, and peak-to-peak values in the portion of the displayed trace that is in the search range. (STATICS ON off under (Utility))

Parameter	Description
OFF or 0	Does not display the statistical values
ON or 1	Displays the statistical values

■ Query Response

{0|1} <new line><^END>

MKR␣{OFF|ON|0|1}

Sets the marker to active (ON) or inactive (OFF). When the MKR is turned off, the marker, sub-marker, and Δmarker are tuned to be off. ((MKR))

Parameter	Description
OFF or 0	Turns off the marker function.
ON or 1	Turns on the marker function.

■ Query Response

{0|1} <new line><^END>

MKRAMPO

Moves the limits so that they are centered an equal amount above and below the marker at the sweep parameter value. (MAKER→AMP.OFS. under System; No query)

MKRAUV?

Outputs the auxiliary amplitude value (value 2) of the measurement value at the marker position. See "Marker Readout" in Chapter 8 for the auxiliary amplitude value of each display format. (Query only)

- Query Response

  {*numeric*} <new line><^END>

MKRCENT

Sets the sweep parameter center value of the destination channel to the sweep parameter value of the marker and centers the new span about that value. (MKR→CENTER under Marker→; No query)

MKRCONT␣{OFF|ON|0|1}

Sets the continuous or discontinuous marker mode. (MKR [ ] under Marker)

Parameter	Description
OFF or 0	Discontinuous marker mode.
ON or 1	Continuous marker mode.

- Query Response

  {0|1} <new line><^END>

MKRCOUP␣{OFF|ON|0|1}

Sets the coupled or uncoupled marker mode. (MKR [ ] under (Marker))

Parameter	Description
OFF or 0	Uncoupled marker mode
ON or 1	Coupled marker mode

■ Query Response

{0|1} <new line><^END>

MKRDELA

Sets the group delay at the marker point of a fixed frequency aperture, 20% of the span, to the electrical delay to balance the phase of the DUT. (Network analyzer only) (MKR→DELAY under (Scale Ref); No query)

MKRL␣{OFF|ON|0|1}

Sets the maker list function ON or OFF. (MKR LIST ON off under (Utility))

Parameter	Description
OFF or 0	Marker list function OFF
ON or 1	Marker list function ON

■ Query Response

{0|1} <new line><^END>

MKRMIDD

Sets the midpoint the LIMD command using the marker to set the middle amplitude value of a limit segment. (MKR→MIDDLE under (System); No query)

This command is valid while editing a segment in a limit line table.

MKRNOI␣{OFF|ON|0|1}

Sets the noise format of the marker ON or OFF. This marker reads out the average noise level at the marker position (referenced to a 1 Hz noise power bandwidth). (Spectrum analyzer only) (NOISE FORM ON off under (Utility))

This command is valid when the display format is set to "spectrum" by FMT SPECT command.

- Query Response

  {0|1} <new line><^END>

MKRO␣{DATA|MEMO}

Sets a trace from data or memory to be applied for the marker values. (MKR ON [ ] under (Marker))

Parameter	Description
DATA	DATA TRACE
MEMO	MEMORY TRACE

- Query Response

  {DATA|MEMO} <new line><^END>

MKROFS

Sets the marker's amplitude value into the offset value. (MKR→OFFSET under (Display); No query)

MKRP␣<*numeric*>

Moves the marker to the specified data point number.

Parameter	Description
<*numeric*>	1 to Number of Points

- Query Response

{*numeric*} <new line><^END>

MKRPKD

Sets the peak delta value to the smaller value of the difference of amplitude values between the present marker position and both side display points of the marker. (Network and impedance analyzers only) (MKR→PEAK DELTA under (Search); No query)

MKRPRM␣<*numeric*>[HZ|DBM]

Sets the marker at the point of the specified sweep parameter, when the marker is ON.

Parameter	Range	Unit
<*numeric*>	start value to stop value	Hz (frequency) dBm (power)

- Query Response

{*numeric*} <new line><^END>

MKRREF

Makes the reference value of the destination channel equal to the marker's absolute value (regardless of the Δmarker value). (MKR→REFERENCE under (Scale Ref) and (Marker→); No query)

MKRSTAR

Sets the start value sweep parameter of the destination channel to the sweep parameter value of the marker. When editing a segment in a list sweep table, sets the start value sweep parameter of the segment to the sweep parameter value of the marker. (SEGMENT: MKR→START under (Sweep), or MKR→START under (Marker→); No query)

MKRSTOP

Sets the sweep parameter start value of the destination channel to the sweep parameter value of the marker. (SEGMENT: MKR→START under (Sweep), or MKR→START under (Marker→); No query)

MKRSWPRM

Sets the segment sweep parameter value to the present marker sweep parameter value. This command is used when editing a limit line table. (MKR→SWP PARAM under (System); No query)

MKRTHRE

Sets the threshold value to the amplitude value of the present marker position. (MKR→THRESHOLD under (Search); No query)

MKRTIME␣{OFF|ON|0|1}

Sets the x-axis units to time, (the start point is zero and the stop point is the value of the sweep time). (MKR TIME ON off under (Utility))

Parameter	Description
OFF or 0	Sets the x-axis to the sweep parameter
ON or 1	Sets the x-axis to time

- Query Response

{0|1} <new line><^END>

MKRUNIT␣{DBM|DBV|DBUV|W|V}

Selects the unit of the marker readout on the active channel when operating in the spectrum analyzer mode. (Spectrum analyzer only) (`UNIT: dBm`, `dBV`, `dBμV`, `WATT`, `VOLT` under (Utility))

Parameter	Description
DBM	dBm
DBV	dBV
DBUV	dBμV
W	Watt
V	Volt

MKRVAL?

Outputs the amplitude value of the measurement value at the marker position. See "Marker Readout" in Chapter 8 for the amplitude value of each display format. (Query only)

- Query Response

  {*numeric*} <new line><^END>

MKRZM

Sets the sweep parameter center value of the destination channel to the sweep parameter value of the marker, and changes the sweep parameter span value of the destination channel to "sweep parameter span × zooming aperture." (`MKR ZOOM` under (Marker→); No query)

MODI1

Leads to the modify calibration kit menu, where a calibration kit can be user-modified. (Network and impedance analyzers only) (`MODIFY [ ]` under (Cal); No query)

MODICOMK

Leads to the modify fixture compensation kit menu. (MODIFY [ ] under (Cal) COMPEN KIT[] ; No query; Impedance analyzer only.)

MODIFIX

Leads to the modify user fixture menu. (MODIFY [ ] under (Meas) FIXTURE[] ; No query; Impedance analyzer only.)

MONDYEAR

Changes the displayed date to the “month:day:year” format. (DATE MODE: MonDayYear under (System))

- Query Response

  {0|1} <new line><^END>

Parameter	Description
0	“day:month:year” format
1	“month:day:year” format

N

Commands in Entry Block N

NA

Selects the network analyzer as the analyzer type. (NETWORK ANALYZER under (Meas))

- Query Response

  {0|1} <new line><^END>

Parameter	Description
0	Network analyzer is not selected.
1	Network analyzer is selected.

NEGL

Sets the I/O signal from the 24-bit I/O port as negative logic.

NEXP

Displays the next page of information in a tabular listing. (NEXT PAGE under (Copy); No query)

NUMG␣<*numeric*>

Triggers a user-specified number of sweeps and returns to the HOLD mode. (NUMBER OF GROUPS under (Trigger); No query)

Parameter	Description
<*numeric*>	1 to 999 (if <*numeric*> is 0 or less than 0, it is set to 1.)

- Query Response

  {*numeric*} <new line><^END>

O

Commands in Entry Block O (*O included)

OFSD␣<*numeric*>[S]

Specifies the one-way electrical delay from the measurement (reference) plane to the standard. (Network and impedance analyzers only) (OFFSET DELAY under Cal; No query)

Parameter	Range	Unit
<*numeric*>	−0.01 to 0.01 (1.0×$^{-9}$ resolution)	sec

- Query Response

{*numeric*} <new line><^END>

OFSL␣<*numeric*>

Specifies energy loss, due to skin effect, along a one-way length of coaxial cable offset. (Network and impedance analyzers only) (OFFSET LOSS under Cal; No query)

Parameter	Range	Unit
<*numeric*>	0 to 1×10^{15}	Ω/s or GΩ/s

- Query Response

{*numeric*} <new line><^END>

OFSZ␣<*numeric*>[OHM]

Specifies the characteristic impedance of the coaxial cable offset. (Network and impedance analyzers only) (`OFFSET Z0` under (Cal); No query)

Parameter	Range	Unit
<*numeric*>	0.001 to 5000000 (=5M)	Ω

- Query Response

{*numeric*} <new line><^END>

OMII

Omits correction for isolation of a 2-port calibration. (Network analyzer only) (`OMIT ISOLATION` under (Cal); No query)

*OPC

Tells the analyzer to set bit 0 (Operation Complete bit) in the Standard Event Status Register when it completes all pending operations.

`*OPC?` query places an ASCII character 1 into the analyzer's output queue when all pending operations have been completed.

- Query Response

{1} <new line><^END>

OPEP

Provides a tabular listing on the display of the key parameters for both channels. (`OPERATING PARAMETERS` under (Copy); No query)

*OPT?

Queries the options installed. (Query only)

- Query Response

  {parameter} <new line><^END>

Parameter	Description
(Null)	None
`1C2`	HP Instrument BASIC
`1D6`	Time-gated spectrum analysis

OSE␣<*numeric*>

Enables the operational status register.

Parameter	Description
<*numeric*>	Decimal expression of the contents of the register, 0 to 65535 ($=2^{16}-1$)

- Query Response

  {*numeric*} <new line><^END>

OSER?

Outputs the current value in the event register of an operational status register. (Query only)

- Query Response

  {*numeric*} <new line><^END>

OSNT␣<*numeric*>

Sets the negative transition filter of an operational status register.

Parameter	Description
<*numeric*>	Decimal expression of the contents of the register, 0 to 65535 ($=2^{16}-1$)

■ Query Response

{*numeric*} <new line><^END>

OSPT␣<*numeric*>

Sets the positive transition filter of an operational status register.

Parameter	Description
<*numeric*>	Decimal expression of the contents of the register, 0 to 65535 ($=2^{16}-1$)

■ Query Response

{*numeric*} <new line><^END>

OSR?

Outputs the operational status register value. (Query only)

■ Query Response

{*numeric*} <new line><^END>

OUT1ENV{H|L}

Sets the 24-bit I/O port, OUTPUT1, to become HIGH (or LOW) when INPUT1 detects pulse input.

OUT1{H|L}

Sets the 24-bit I/O port, OUTPUT1, to HIGH (or LOW).

OUT2ENV{H|L}

Sets the 24-bit I/O port, OUTPUT2, to become HIGH (or LOW)when INPUT1 detects pulse input.

Note MODIFY JAPANESE TEXT

OUT2{H|L}

Sets the 24-bit I/O port, OUTPUT2, to HIGH (or LOW).

Note MODIFY JAPANESE TEXT

OUT8IO␣<*numeric*>

Outputs the data to the 8-bit parallel output port. (No query)

Parameter	Description
<*numeric*>	0 to 255

OUTAIO␣$<numeric>$

Outputs 8-bit wide data to the 24-bit I/O port, the port A. (No Query)

Parameter	Description
$<numeric>$	0 to 255

OUTBIO␣$<numeric>$

Outputs 8-bit wide data to the 24-bit I/O port, the port B. (No Query)

Parameter	Description
$<numeric>$	0 to 255

OUTCIO␣$<numeric>$

Outputs 4-bit wide data to the 24-bit I/O port, the port C. (No Query)

You must first use `COUT` to set the port C as the output port before using this command.

Parameter	Description
$<numeric>$	0 to 15

OUTDIO␣$<numeric>$

Outputs 4-bit wide data to the 24-bit I/O port, the port D. (No Query)

You must first use `COUT` to set the port D as the output port before using this command.

Parameter	Description
$<numeric>$	0 to 15

OUTEIO␣<*numeric*>

Outputs 8-bit wide data to the 24-bit I/O port, the port E. (No Query)

You must set the port C, D as the output ports with `COUT` and `DOUT` before using this command.

Parameter	Description
<*numeric*>	0 to 255

OUTFIO␣<*numeric*>

Outputs 16-bit wide data to the 24-bit I/O port, the port F. (No Query)

Parameter	Description
<*numeric*>	0 to 65535

OUTGIO␣<*numeric*>

Outputs 20-bit wide data to the 24-bit I/O port, the port G. (No Query)

You must first use `COUT` to set the port C as the output port before using this command.

Parameter	Description
<*numeric*>	0 to 1048575

OUTHIO␣<*numeric*>

Outputs 24-bit wide data to the 24-bit I/O port, the port H. (No Query)

You must set the port C, D as the output ports with `COUT` and `DOUT` before using this command.

Parameter	Description
<*numeric*>	0 to 16777215

OUTPCALC{1-12}?

Outputs the active calibration set array of the active channel. (Network and impedance analyzers only) (Query only)

- Query Response

  {*numeric (1)*} {*numeric (2)*} ... {*numeric (n)*} <new line><^END> (n is the number of points.)

  numeric is a complex number (data format: real, imaginary).

OUTPCALK?

Outputs the active calibration kit. (Network and impedance analyzers only) (Query only)

- Query Response

  {block data (714 bytes of binary data)} <new line><^END>

OUTPCOMC{1|2|3}?

Outputs data of the fixture compensation arrays. See "INPUCOMC{1|2|3}␣␣<*numeric (1)*>,<*numeric (2)*>, ... ,<*numeric (n)*>" in Appendix I for details about the fixture compensation arrays. (Impedance analyzer only)

- Query Response

  {*numeric (1)*} {*numeric (2)*} ... {*numeric (n)*} <new line><^END>

  (n is the number of points.)

  numeric is a complex number. (data format: real, imaginary)

OUTPCOMK?

Outputs data of the fixture compensation kit that is currently defined.

OUTPINPCIO?

Loads data through port C of the 24-bit I/O port and returns the value to GPIB (Query only).

Use `CIN` to specify port C as an input port before using this command.

- Query Response

  {*numeric*} <new line><^END>

OUTPDATA?

Outputs the error corrected data. (Query only)

- Query Response

  {*numeric (1)*} {*numeric (2)*} ... {*numeric (n)*} <new line><^END>

  (n is the number of points.)

  numeric is a complex number (data format: real, imaginary) for the Network analyzer, or a real number for the Spectrum analyzer.

OUTPDATAP?␣<*numeric*>

Outputs the error corrected data at the specified point. (Query only)

Parameter	Description
<*numeric*>	1 to "number of points" (If <*numeric*> is 0 or less than 0, it is set to 1. If <*numeric*> is greater than "number of points," it is set to "number of points.")

- Query Response

  {*numeric (real)*} {*numeric (imaginary)*} <new line><^END> (Network analyzer)

  {*numeric (val)*} <new line><^END> (Spectrum analyzer)

OUTPDMKR?

Outputs sweep parameter and measurement value at the Δmarker position. (Query only)

- Query Response

  {*numeric (val1)*} {*numeric (val2)*} {*numeric (stimulus)*} <new line><^END>

  (Val1: Amplitude value, Val2: Auxiliary amplitude value.)

OUTPDTRC?

Outputs DATA TRACE data. (Query only)

- Query Response

  {*numeric (1:val1)*} {*numeric (1:val2)*} {*numeric (2:val1)*} {*numeric (2:val2)*} ...
  {*numeric (n:val2)*} {*numeric (n:val2)*} <new line><^END> (Network analyzer)

  {*numeric (1)*} {*numeric (2)*} ... {*numeric (n)*} <new line><^END> (Spectrum analyzer)

  (n is the number of points.) (Val1: Amplitude value, Val2: Auxiliary amplitude value.)

OUTPDTRCP?␣<*numeric*>

Outputs DATA TRACE data at the specified point. (Query only)

Parameter	Description
<*numeric*>	1 to "number of points" (If <*numeric*> is 0 or less than 0, it is set to 1. If <*numeric*> is greater than "number of points," it is set to "number of points.")

- Query Response

  {*numeric (val1)*} {*numeric (val2)*} <new line><^END> (Network analyzer)

  {*numeric (val)*} <new line><^END> (Spectrum analyzer)

  (Val1: Amplitude value, Val2: Auxiliary amplitude value.)

OUTPERRO?

Outputs the error message in the error queue.

- Query Response

  {*numeric (Error number)*} {*string (Error message)*} <new line><^END>

  When no message is in the queue: 0,“No error” <new line><^END>

OUTPFAIP?

Outputs number of the failed point of the limit test. (Query only)

- Query Response

  {*numeric*} <new line><^END>

OUTPINPCIO?

Note OUTPINP[C|D|E]IO? DOUBLED?

Loads data through port C of the 24-bit I/O port and returns the value to GPIB (Query only).

Use `CIN` to specify port C as an input port before using this command.

- Query Response

  {*numeric*} <new line><^END>

OUTPINPDIO?

Loads data through port D of the 24-bit I/O port and returns the value to GPIB (Query only).

Use `DIN` to specify port D as an input port before using this command.

- Query Response

  {*numeric*} <new line><^END>

OUTPINPEIO?

Loads data through port E of the 24-bit I/O port and returns the value to GPIB (Query only).

Use `CIN` and `DIN` to specify port C and D as an input port before using this command.

- Query Response

  {*numeric*} <new line><^END>

OUTPLIMF?

Outputs the limit test results only for the failed points. (Query only)

- Query Response

  {*numeric (stimulus 1)*} {0} {*numeric (upper_limit 1)*} {*numeric (lower_limit 1)*}
  {*numeric (stimulus 2)*} {0} {*numeric (upper_limit 2)*} {*numeric (lower_limit 2)*}
  ⋮
  {*numeric (stimulus n)*} {0} {*numeric (upper_limit n)*} {*numeric (lower_limit n)*} <new line><^END> (Form 4)

  (n is the number of failed points.)

  {0} <new line><^END> (for no failed points.)

OUTPLIML?

Outputs the limit test results for each point. (Query only)

- Query Response

  {*numeric (stimulus 1)*} {*numeric (result 1)*} {*numeric (upper_limit 1)*} {*numeric (lower_limit 1)*}

  {*numeric (stimulus 2)*} {*numeric (result 2)*} {*numeric (upper_limit 2)*} {*numeric (lower_limit 2)*}

  ⋮

  {*numeric (stimulus n)*} {*numeric (result n)*} {*numeric (upper_limit n)*} {*numeric (lower_limit n)*} <new line><^END> (Form 4)

  (n is the number of points.) (*result* is 1 for pass, 0 for fail, or −1 for no test.)

OUTPLIMM?

Outputs the limit test result for the marker position. (Query only)

- Query Response

  {*numeric (stimulus)*} {*numeric (result)*} {*numeric (upper_limit)*} {*numeric (lower_limit)*} <new line><^END>

  (*result* is 1 for pass, 0 for fail, or −1 for no test)

OUTPMEMO?

Outputs the memory data from the active channel. (Query only)

- Query Response

  {*numeric (1)*} {*numeric (2)*} ... {*numeric (n)*} <new line><^END>

  (n is the number of points.)

  numeric is a complex number (data format: real, imaginary) for the Network analyzer, or a real number for the Spectrum analyzer.

OUTPMEMOP?␣<*numeric*>

Outputs the memory data from the active channel at a specified point. (Query only)

Parameter	Description
<*value*>	1 to "number of points" (If <*numeric*> is 0 or less than 0, it is set to 1. If <*numeric*> is greater than "number of points," it is set to "number of points.")

- Query Response

  {real} {imaginary} <new line><^END> (Network analyzer)

  {numeric} <new line><^END> (Spectrum analyzer)

OUTPMKR?

Outputs the sweep parameter and measurement values at the marker position. (Query only)

- Query Response

  {*numeric (val1)*} {*numeric (val2)*} {*numeric (stimulus)*} <new line><^END>

  (Val1: Amplitude value, Val2: Auxiliary amplitude value.)

OUTPMSTA?

Outputs the marker statistics. (STATISTICS ON off under Utility; Query only)

- Query Response

  {*numeric (mean)*} {*numeric (standard deviation)*} {*numeric (peak to peak)*}
  <new line><^END>

OUTPMTRC?

Outputs the MEMORY TRACE data. (Query only)

- Query Response

  {*numeric (1:val1)*} {*numeric (1:val2)*} {*numeric (2:val1)*} {*numeric (2:val)*} ...
  {*numeric (n:val1)*} {*numeric (n:val2)*} <new line><^END> (Network analyzer)

  {*numeric (1)*} {*numeric (2)*} ... {*numeric (n)*} <new line><^END> (Spectrum analyzer)

  (n is the number of points.) (Val1: Amplitude value, Val2: Auxiliary amplitude value.)

OUTPMTRCP?␣<*numeric*>

Outputs the MEMORY TRACE data at the specified point. (Query only)

Parameter	Description
<*numeric*>	1 to "number of points" (If <*numeric*> is 0 or less than 0, it is set to 1. If <*numeric*> is greater than "number of points," it is set to "number of points.")

- Query Response

  {*numeric (val1)*} {*numeric (val2)*} <new line><^END> (Network analyzer)

  {*numeric (val)*} <new line><^END> (Spectrum analyzer)

  (Val1: Amplitude value, Val2: Auxiliary amplitude value.)

OUTPMWID?

Outputs the results of the bandwidth search. (Network and impedance analyzers only) (WIDTHS ON off under Search; Query only)

- Query Response

{*numeric (bandwidth)*} {*numeric (center)*} {*numeric (Q)*} <new line><^END>

OUTPRAW{1-4}?

Outputs the uncorrected data arrays for the active channel. (Query only)

- Query Response

{*numeric (1)*} {*numeric (2)*} ... {*numeric (n)*} <new line><^END>

(n is the number of points.)

numeric is a complex number (data format: real, imaginary) for the Network analyzer, or a real number for the Spectrum analyzer.

OUTPSMKR{1-7}?

Outputs the measurement values and sweep parameter at the sub-marker position. (Query only)

- Query Response

{*numeric (val1)*} {*numeric (val2)*} {*numeric (stimulus)*} <new line><^END>

(Val1: Amplitude value, Val2: Auxiliary amplitude value.)

OUTPSWPRM?

Outputs the sweep parameter data. (Query only)

- Query Response

{*numeric 1*} {*numeric 2*} ... {*numeric n*} <new line><^END>

(n is the number of points.)

OUTPSWPRMP?␣<*numeric*>

Outputs the sweep parameter data at a specified point. (Query only)

Parameter	Description
<*numeric*>	1 to "number of points" (If <*numeric*> is 0 or less than 0, it is set to 1. If <*numeric*> is greater than "number of points," it is set to "number of points.")

- Query Response

  {*numeric*} <new line><^END>

P

Commands in Entry Block P (∗P included)

PARS␣{OFF|ON|0|1}

Sets the partial search of the marker search function ON or OFF. (PART SRCH ON off under Search)

Parameter	Description
OFF or 0	Partial search OFF
ON or 1	Partial search ON

- Query Response

  {0|1} <new line><^END>

∗PCB␣<*numeric*>

Specifies the address of a controller that is temporarily passing GPIB control to the analyzer. (Option 1C2 only; No query)

Parameter	Description
<*numeric*>	0 to 30

PEAKCENT

Searches for a peak using the marker and then changes the CENTER of the destination channel to the sweep parameter value of that peak. (PEAK→CENTER under Center or Marker→; No query)

PEAKREF

Searches for a peak using the marker and applies a sweep parameter at the marker to the reference value of the sweep parameters for the destination channel. The sweep parameter specified is an absolute value;not a difference even if a Δmarker is used. (Spectrum analyzer only) (PEAK→REFERENCE under (Scale Ref); No query)

PEN␣{1-6}

Specifies the pen to be used in displaying a data trace by DATOVE command. (SELECT PEN COLOR under (Display); No query)

The color for each pen can be specified using COLO command.

Parameter	Description
1	PEN 1
2	PEN 2
3	PEN 3
4	PEN 4
5	PEN 5
6	PEN 6

PHAO␣<*numeric*>[DEG]

Adds or subtracts a phase offset that is constant with frequency. (Network analyzer only) (PHASE OFFSET under (Scale Ref))

Parameter	Range	Unit
<*numeric*>	−360 to +360	°

- Query Response

{*numeric*} <new line><^END>

PHAU {RAD|DEG}

Selects the unit of phase format. (`PHASE UNIT []` under (Format); Impedance analyzer only.)

Parameter	Description
DEG	Degree.
RAD	Radian.

- Query Response

{DEG|RAD}<new line><^END>

PKDLTX␣<*numeric*>[HZ|DBM]

Sets the peak ΔX value that is used to define the peak. (Network and impedance analyzers only) (`PEAK DEF: ΔX` under (Search))

Parameter	Range	Unit
<*numeric*>	1×10^{-9} to 1×10^{9}	Hz (frequency)
		dBm (power)

- Query Response

{*numeric*} <new line><^END>

PKDLTY␣<*numeric*>

Sets the peak ΔY value that is used to define the peak. (`PEAK DEF: ΔY` under (Search))

Parameter	Range	Format
<*numeric*>	-1×10^{9} to 1×10^{9}	????

- Query Response

{*numeric*} <new line><^END>

PKPOL␣{POS|NEG}

Sets the peak polarity for the marker search functions. (Network and impedance Analyzers only) (PEAK PLRTY pos neg under Search)

Parameter	Description
POS	Positive peak
NEG	Negative peak

■ Query Response

{POS|NEG} <new line><^END>

PKTHRE␣{OFF|ON|0|1}

Sets the threshold ON or OFF. (THRESHOLD ON off under Search)

Parameter	Description
OFF or 0	Threshold OFF
ON or 1	Threshold ON

■ Query Response

{0|1} <new line><^END>

PKTHVAL␣<*numeric*>

Sets the threshold values. (THRESHOLD VALUE under Search)

Parameter	Range	Unit
<*numeric*>	-1×10^9 to 1×10^9	????

■ Query Response

{*numeric*} <new line><^END>

POIN␣<*numeric*>

Sets the number of points for the segment, or sets the number of points for the list sweep table. (In the spectrum analyzer mode, this command can set the number of points for zero span measurement only;can be used to query in the other measurement types.)
(`NUMBER OF POINTS` under (Sweep))

Parameter	Description
<*numeric*>	2 to 801.

- Query Response

{*numeric*} <new line><^END>

PORE␣{OFF|ON|0|1}

Sets the reference plane extension mode ON or OFF. (Network and impedance analyzers only)
(`EXTENSIONS ON off` under (Cal))

Parameter	Description
`OFF` or `0`	Reference plane extension mode OFF
`ON` or `1`	Reference plane extension mode ON

- Query Response

{0|1} <new line><^END>

PORT1␣<*numeric*>[S|MS|US|NS|PS]

Extends the reference plane for measurement of S_{11}, S_{21}, and S_{12}. (Network analyzer only)
(`EXTENSION PORT 1` under (Cal))

Parameter	Range	Unit
<*numeric*>	-0.01 to 0.01 (1×10^{-12} resolution)	sec

- Query Response

{*numeric*} <new line><^END>

PORT2␣<*numeric*>[S]

Extends the reference plane for measurement of S_{22}, S_{12}, and S_{21}. (Network analyzer only) (EXTENSION PORT 2 under (Cal))

Parameter	Range	Unit
<*numeric*>	−0.01 to 0.01 (1×10^{-12} resolution)	sec

- Query Response

  {*numeric*} <new line><^END>

PORTA␣<*numeric*>[S]

Adds electrical delay to the input A reference plan for all A input measurements (including S-parameters). (Network analyzer only) (EXTENSION INPUT A under (Cal))

Parameter	Range	Unit
<*numeric*>	−0.01 to 0.01 (1×10^{-12} resolution)	sec

- Query Response

  {*numeric*} <new line><^END>

PORTB␣<*numeric*>[S]

Adds electrical delay to the input B reference plane for all B input measurements (including S-parameters). (Network analyzer only) (EXTENSION INPUT B under (Cal))

Parameter	Range	Unit
<*numeric*>	−0.01 to 0.01 (1×10^{-12} resolution)	sec

- Query Response

  {*numeric*} <new line><^END>

PORTR␣<*numeric*>[S]

Adds electrical delay to extend the reference plane at input R to the end of cable. The compensation takes effects in all the measurement which use the port R, including S parameter measurement. (Network analyzer only) (EXTENSION INPUT R under (Cal))

Parameter	Range	Unit
<*numeric*>	−0.01 to 0.01 (1×10^{-12} resolution)	sec

- Query Response

{*numeric*} <new line><^END>

PORTZ␣<*numeric*>

Sets the port extension value. (EXTENSION VALUE under (Cal); Impedance analyzer only.)

Parameter	Range	Unit
<*numeric*>	−10 to 10	sec

- Query Response

<*numeric*><new line><^END>

POSL

Sets the I/O signal of 24-bit I/O port to positive logic.

POWE␣<*numeric*>[DBM]

Sets the power level segment by segment, or sets the power level for the list sweep table. (POWER under (Sweep))

This command is valid when the linear frequency or log frequency sweeping mode is selected in the network and impedance analyzer modes, or when measuring on zero span in the spectrum analyzer mode.

Parameter	Range	Unit
<*numeric*>	−50 to 15 (0.1 resolution)	dBm

- Query Response

{*numeric*} <new line><^END>

PREP

Displays the previous page of information in a tabular listing. (PREV PAGE under Copy; No query)

PRES

Presets the ANALYZER to the preset default values. See the *Operation Manual* for the default values. The PRES command does *not* preset the HP Instrument BASIC. (PRESET; No query)

PRIC

Sets the print command to the color printing. (COLOR under Copy)

- Query Response

  {0|1} <new line><^END>

Parameter	Description
0	Single-color printing
1	Color printing

PRICFIXE

Sets the default colors for printing a hard copy. (PRINT COLOR [FIXED] under Copy)

- Query Response

  {0|1} <new line><^END>

Parameter	Description
0	Variable colors (colors similar to the display)
1	Fixed colors (default colors)

PRICVARI

Sets the colors used for printing a hard copy as close as possible to the display colors. Refer to "System Accessory Printer" in Chapter 9 of *FuncRef* for the printers which support the variable color printing. (`PRINT COLOR [VARIABLE]` under (Copy))

- Query Response

{0|1} <new line><^END>

Parameter	Description
0	Fixed colors (default colors)
1	Variable colors (colors similar to the display)

PRINALL

Causes an extra copy of the display to be printed. (`PRINT [ ]` under (Copy); No query)

PRIS

Sets the print command to the single color printing. (`PRINT: STANDARD` under (Copy))

- Query Response

{0|1} <new line><^END>

Parameter	Description
0	Color printing
1	Single color printing

PRSMKRS

Turns off all markers and cancels all settings of the marker functions. (`PRESET MKRS` under (Marker); No query)

PRSOFT␣{OFF|ON|0|1}

Sets printing the softkeys displayed in the screen ON or OFF. (COPY SKEY under Copy)

Parameter	Description
OFF or 0	Does not print the soft keys
ON or 1	Print the soft keys

- Query Response

  {0|1} <new line><^END>

PURG␣<*string*>

Removes the file. (PURGE FILE under SAVE; No query)

Parameter	Description
<*string*>	File name, up to 10 characters including the extension

R

Commands in Entry Block R (∗R included)

RAID

Completes the response and isolation calibration. Computes and stores the error coefficients. (Network analyzer only) (`DONE RESP ISOL'N CAL` under (Cal); No query)

RAIISOL

Selects the isolation class for the response and isolation calibration. (Network analyzer only) (`ISOL'N STD` under (Cal); No query)

RAIRESP

Selects the response class for the response and isolation calibration. (Network analyzer only) (`RESPONSE` under (Cal); No query)

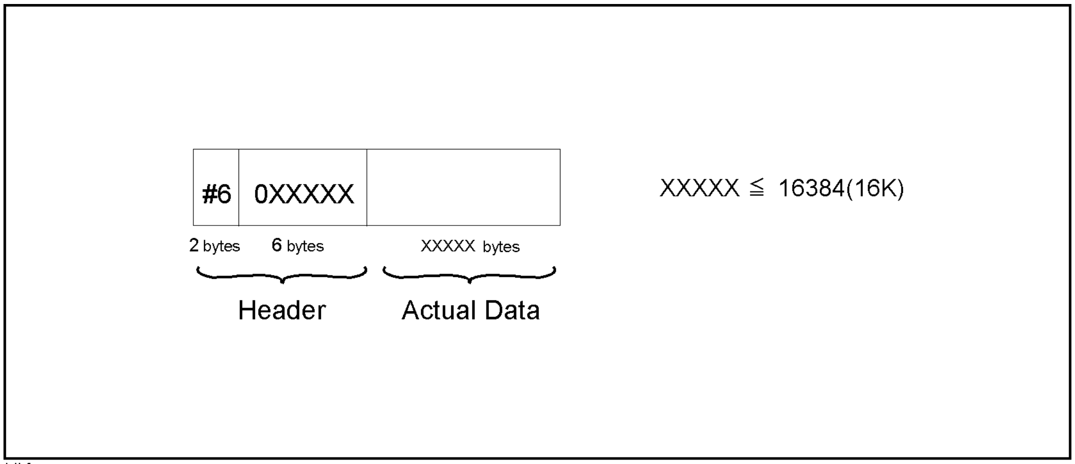
READ?

Reads data from a file that has been read-enabled using the `ROPEN` command. The returned data is in the fixed length block format defined in IEEE488.2. The fixed length block format, as shown in Figure R-1, consists of a header part indicating the data size and an actual data part. In the case of the 4395A, the number of digits to indicate the data size is 6 and the maximum length of the actual data part is 16 Kbytes. If a file contains data greater than 16 Kbytes, execute this command repeatedly to read it. Note that acceptable file formats for this command are the DOS format and the LIF format BDAT type.

Generally, this command is used in combination with the `ROPEN` command and the `CLOSE` command, as shown in Figure R-2. (Query only)

- Query Response

  {*block*} <new line><^END>



blkform

Figure R-1. Fixed length block format

RECC

Recalls the previously saved version of the color set from the non-volatile memory. (RECALL COLORS under Display; No query)

RECD␣<*string*>

Loads the instrument states or data. (file name under Recall; No query)

Parameter	Description
<*string*>	File name, Up to 10 characters including the extension

REFD

Completes with the reflection part of the full 2-port or one-path 2-port calibration. (Network analyzer only) (REFLECT'N DONE under Cal; No query)

REFL

Begins the reflection part of the full 2-port or one-path 2-port calibration. (Network analyzer only) (REFLECT'N under (Cal); No query)

REFP␣<*numeric*>

Sets the position of the reference line on the graticule of a Cartesian display. (Network and impedance analyzers only) (REFERENCE POSITION under (Scale Ref))

Parameter	Range	Unit
<*numeric*>	0 to 10	Div

REFV␣<*numeric*>

Sets the value of the reference line, moving the measurement trace correspondingly. (REFERENCE VALUE under (Scale Ref))

Parameter	Range	Unit
<*numeric*>	-1×10^9 to 1×10^9	????

- Query Response

{*numeric*} <new line><^END>

REFX␣<*numeric*>

Sets the value of the x-axis reference line in complex plane format, moving the measurement trace correspondingly, when the measurement format is set to the complex plane. (REFERENCE X VALUE under (Scale Ref); Impedance analyzer only.)

Parameter	Range	Unit
<*numeric*>	-1×10^9 to 1×10^9	U

- Query Response

<*numeric*><new line><^END>

REFY␣<*numeric*>

Sets the value of the y-axis reference line in complex plane format, moving the measurement trace correspondingly, when the measurement format is set to the complex plane. (REFERENCE Y VALUE under Scale Ref)

Parameter	Range	Unit
<*numeric*>	-1×10^9 to 1×10^9	U

- Query Response

  <*numeric*><new line><^END>

RESAVD␣<*string*>

Updates a file that is already saved. (RE-SAVE FILE under Save; No query)

Parameter	Description
<*string*>	File name up to 10 characters including the extension

RESC

Eliminates the need to restart a calibration sequence that was interrupted to access some other menu. (Network and impedance analyzers only) (RESUME CAL SEQUENCE under Cal; No query)

RESCOM

Resume the last measured compensation sequence. (RESUME COMP SEQ under Cal; No query; Impedance analyzer only.)

RESD

Turns off the tabular listing and returns the measurement display to the screen. (RESTORE DISPLAY under Copy; No query)

RESPDONE

Completes the response calibration. Computes and stores the error coefficients. This command also set ??the error compensation function?? on. (Network analyzer only) (DONE: RESPONSE under Cal; No query)

REST

Aborts the sweep in progress and then restarts the measurement. (MEASURE RESTART under Trigger; No query)

Measurement will restart on the active channel when dual channel display is disabled (DUAC OFF). When dual channel display is enabled (DUAC ON), measurement will restart on both channels;first on the channel 1 then on the channel 2.

If the sweep trigger is in the HOLD mode, this command executes a single sweep.

RESTMDISK␣{2}

Recalls the contents of the built-in RAM disk memory from the built-in flush memory, allowing to use with the parameter to specify the flush memory from which the contents is recalled. (No query)

Parameter	Description
None	Recalls from the backup memory.
2	Recalls from the memory for service/demo/sample.

REVI

Measures S_{12} isolation for the full 2-port calibration. (Network analyzer only) (REV ISOL'N ISOL'N STD under (Cal); No query)

REVM

Measures S_{22} load match for the full 2-port calibration. (Network analyzer only) (REV. MATCH THRU under (Cal); No query)

REVT

Measures S_{12} frequency response for the full 2-port calibration. (Network analyzer only) (REV. TRANS. THRU under (Cal); No query)

RFO␣{OFF|ON|0|1}

Sets the signal output on the RF OUT port ON or OFF. (RF OUT ON off under (Source))

Parameter	Description
OFF or 0	RF OUT port OFF
ON or 1	RF OUT port ON

■ Query Response

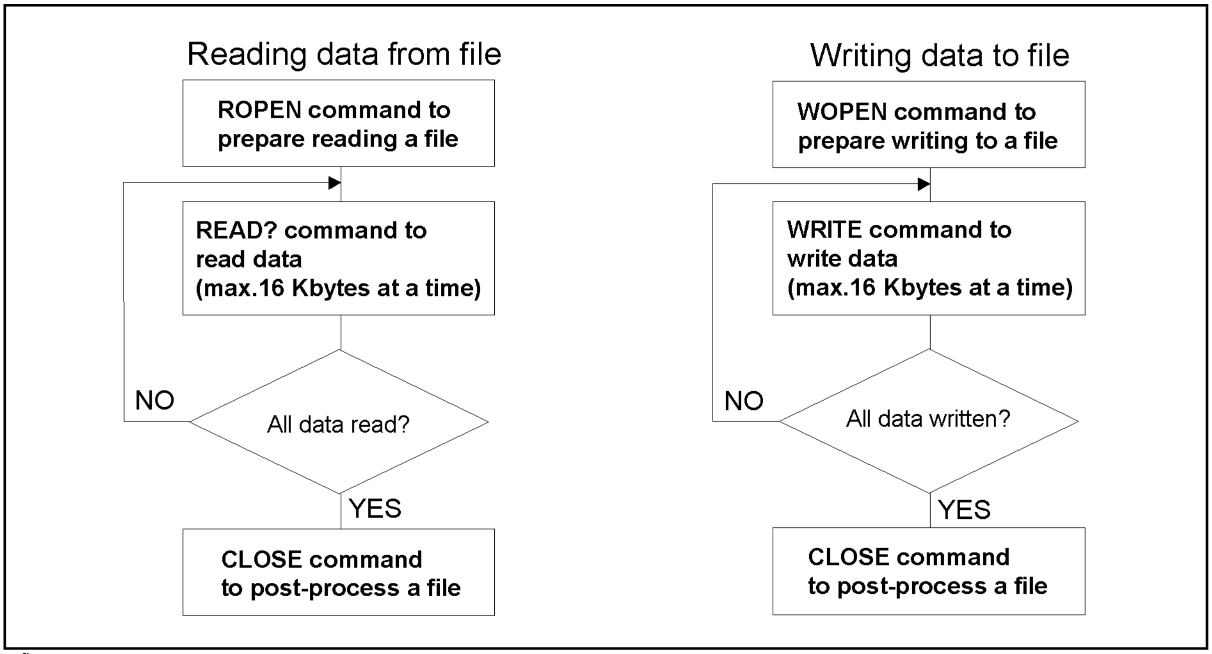
{0|1} <new line><^END>

ROPEN␣<*string*>

Makes a specified file read-enabled. If the file does not exist, an error occurs.

Generally, this command is used in combination with the READ? command and the CLOSE command, as shown in Figure R-2. (No query)

Parameter	Description
<*string*>	File name of up to 12 characters including its extension (for the LIF format, up to 10 characters)



rwflow

Figure R-2. Procedure of executing commands to read/write data

RSCO

Resets the color being modified to the default color. (RESET COLOR under (Display); No query)

*RST

Resets the analyzer to its default values (No query):

- Initializes the instrument settings.
- Sets the trigger mode to HOLD.
- Resets HP Instrument BASIC (only if executed on the external controller)

See *Operation Manual* for information on the default values.

S

Commands in Entry Block S (∗S included)

SA

Selects the spectrum analyzer as the analyzer type. (SPECTRUM ANALYZER under Meas)

- Query Response

{0|1} <new line><^END>

Parameter	Description
0	Spectrum analyzer is not selected.
1	Spectrum analyzer is selected.

SADD␣<*numeric*>

Adds a new segment to a list sweep table. (ADD under Sweep; No query)

Parameter	Range	Unit
<*numeric*>	1 to 51 (Network and impedance analyzers)	
	1 to 15 (Spectrum analyzer)	

SAUNIT␣{DBM|DBV|DBUV|W|V}

Selects the unit of the measurement data on the active channel when operating in the spectrum analyzer mode . (Spectrum analyzer only) (UNIT: dBm, dBV, dBμV, WATT, VOLT under Format)

Parameter	Description
DBM	dBm
DBV	dBV
DBUV	dBμV
W	Watt
V	Volt

- Query Response

{DBM|DBV|DBUV|W|V|WLOGY|VLOGY} <new line><^END>

SAV1

Completes the S_{11} or S_{22} 1-port calibration. The error coefficients are computed and stored. (Network analyzer only) (DONE: 1-PORT CAL under Cal; No query)

SAV2

Completes the full or one-path 2-port calibration. The error coefficients are computed and stored. (Network analyzer only) (DONE: 2-PORT CAL under Cal; No query)

SAVC

Initializes and performs error compensation on a raw data array based on the error coefficients array and stores the resulting data on the data trace array. This command then redraws a trace using the current error coefficient array data. (Network and impedance analyzers only) (No query)

This command should be executed after the error coefficients are transferred using INPUCALC{1-12} command.

SAVCAL␣{OFF|ON|0|1}

Selects whether or not to save the calibration coefficients arrays. (CAL ON off under Save; No query for the SCPI command)

Parameter	Description
OFF or 0	Does not save the calibration coefficients arrays.
ON or 1	Saves the calibration coefficients arrays.

- Query Response

  {0|1} <new line><^END>

SAVCOM

Calculates the fixture compensation coefficients and store them. This command also enables one of fixture compensation functions (`COMDAT{A|B|C}`) which applies. (`DONE: COMPEN` under (Cal); No query; Impedance analyzer only)

SAVDASC␣<*string*>

Specifies saving the internal data arrays as an ASCII file. (`SAVE ASCII` under (Save); No query)

Parameter	Description
<*string*>	File name, up to 8 characters

SAVDAT␣{OFF|ON|0|1}

Selects whether or not to save the data arrays. (`DATA ON off` under (Save); No query for the SCPI command)

Parameter	Description
`OFF` or `0`	Does not save the data arrays.
`ON` or `1`	Saves the data arrays.

- Query Response

  {0|1} <new line><^END>

SAVDDAT␣<*string*>

Specifies saving the internal data arrays which are defined by the `SAVRAW`, `SAVCAL`, `SAVDAT`, `SAVMEM`, `SAVTDAT`, and `SAVTMEM` commands. (`SAVE BINARY` under (Save); No query)

Parameter	Description
<*string*>	File name up to 8 characters

SAVDTIF␣<*string*>

Specifies the file format for saving the screen currently displayed as the TIFF format. (GRAPHICS under Save/Recall; No query)

Parameter	Description
<*string*>	File name contains up to eight characters.

SAVDSTA␣<*string*>

Specifies saving only the instrument states and the calibration coefficients. Also saved are the raw data array, the ??TrcMem?? array, the ??DatMem?? array, and the ??Hld?? array. (STATE under Save; No query)

Parameter	Description
<*string*>	File name up to 8 characters

SAVDTRC␣{OFF|ON|0|1}

Sets whether or not to save the trace arrays. (DATA TRACE ON off under Save; No query for the SCPI command)

Parameter	Description
OFF or 0	Does not save the trace arrays.
ON or 1	Saves the trace arrays.

- Query Response

{0|1} <new line><^END>

SAVEUSEK

Stores the user-modified or user-defined calibration kit into memory. (Network and impedance analyzers only) (SAVE USER KIT under Cal; No query)

SAVIMP

Calculates the error-correction coefficients from the calibration data and stores the coefficients. (DONE:CAL under CAL; No query; Impedance analyzer only)

SAVMEM␣{OFF|ON|0|1}

Specifies whether or not to save the memory arrays. (MEM ON off under Save; No query for the SCPI command)

Parameter	Description
OFF or 0	Does not save the memory arrays.
ON or 1	Saves the memory arrays.

- Query Response

{0|1} <new line><^END>

SAVMTRC␣{OFF|ON|0|1}

Specifies whether or not to save the memory trace arrays. (MEM TRACE ON off under Save; No query for the SCPI command)

Parameter	Description
OFF or 0	Does not save the memory trace arrays.
ON or 1	Saves the memory trace arrays.

- Query Response

{0|1} <new line><^END>

SAVRAW␣{OFF|ON|0|1}

Specifies whether or not to save the raw data arrays. (RAW ON off under (Save); No query for the SCPI command)

Parameter	Description
OFF or 0	Does not save the raw data arrays.
ON or 1	Saves the raw data arrays.

- Query Response

  {0|1} <new line><^END>

SAVUCOMK

Stores the user-modified compensation kit into memory. (SAVE COMPEN KIT under (Cal); No query; Impedance analyzer only)

SAVUFIXT

Saves the settings of user defined fixture. (SAVE USER FXTR KIT under (Meas) FIXTURE [] ; No query; Impedance analyzer only)

SCAC␣{OFF|ON|0|1}

Couples or uncouples the "DATA" and "MEMORY" traces to be scaled. (D&M SCALE [ ] under (Scale Ref);)

Parameter	Description
OFF or 0	Uncouples the "DATA" and "MEMORY" traces.
ON or 1	Couples the "DATA" and "MEMORY" traces.

- Query Response

  {0|1} <new line><^END>

SCAF␣{DATA|MEMO}

Selects one of the "DATA" or "MEMORY" traces to be scaled. (`SCALE FOR [ ]` under (Scale Ref); No equivalent SCPI command)

- Query Response

  {DATA|MEMO} <new line><^END>

SCAL␣<*numeric*>

Sets the response value scale per graticule trace. (`SCALE/DIV` under (Scale Ref))

Parameter	Range	Format
<*numeric*>	1f to 100M	????

- Query Response

  {*numeric*} <new line><^END>

SCRN␣{OFF|ON|0|1}

Controls whether the LCD display is visible or not.

Parameter	Description
`OFF` or `0`	Invisible (only softkey labels are displayed.)
`ON` or `1`	Visible

- Query Response

  {0|1} <new line><^END>

SDEL

Deletes a segment from a list sweep table. (DELETE under Sweep; No query)

Parameter	Range	Unit
<*numeric*>	1 to 51 (Network and impedance analyzers)	
	1 to 15 (Spectrum analyzer)	

SDON

Saves the modified segment of a list sweep table and exit the editing. (SEGMENT DONE under Sweep; No query)

SEAL

Searches the trace for the next occurrence of the target value to the left of the marker. (Network and impedance analyzers only) (SEARCH LEFT under Search; No query)

SEAM␣{PEAK|MAX|MIN|TARG|PKSA|PKSR|PKSL|OFF}

Selects the marker search function. (SEARCH: PEAK, MAX, MIN, TARGET, SEARCH: PEAKS ALL, PEAKS RIGHT, PEAKS LEFT under Search; No query for the SCPI command)

Parameter	Description
PEAK	Peak search
MAX	Maximum search
MIN	Minimum search
TARG	Target search (Network and impedance analyzers only)
PKSA	Peak all
PKSR	Peak right all
PKSL	Peak left all
OFF	Marker search function OFF

- Query Response

  {PEAK|MAX|MIN|TARG|PKSA|PKSR|PKSL|OFF} <new line><^END>

SEANPK

Moves the marker to the next peak. (NEXT PEAK under Search; No query)

SEANPKL

Moves the marker to the peak to the left of the present marker position. (NEXT PEAK LEFT under Search; No query)

SEANPKR

Moves the marker to the peak to the right of the present marker position. (NEXT PEAK RIGHT under Search; No query)

SEAR

Searches the trace for the next occurrence of the target value to the right of the marker. (Network and impedance analyzers only) (SEARCH RIGHT under Search; No Query)

SEARSTR

Sets the partial search range to the range between the marker and the Δmarker. (MKRΔ→SEARCH RNG under Search; No query)

SEARSTRL

Sets the left (lower) border of the partial search range at the current position of the marker. (MKR→LEFT RNG under Search; No query)

SEARSTRR

Sets the right (higher) border of the partial search range at the current position of the marker. (MKR→RIGHT RNG under Search; No query)

SEATARG␣<*numeric*>[DB|DEG|S|OHM]

Makes the target value to the active function to enter a value and moves the marker to a specified target point on the trace. (Network and impedance analyzers only) (TARGET under Search)

In the Δmarker mode, specify a relative value to the Δ for the target value.

Parameter	Range	Unit
<*numeric*>	-1×10^{9} to 1×10^{9}	

- Query Response

{*numeric*} <new line><^END>

SEDI␣<*numeric*>

Determines the segment of the list sweep table to be modified. (EDIT under Sweep; No query for the SCPI command)

Parameter	Range	Unit
<*numeric*>	1 to 51 (Network and impedance analyzers) 1 to 15 (Spectrum analyzer)	

- Query Response

{*numeric*} <new line><^END>

SETCDATE␣<*numeric (year)*>,<*numeric (month)*>,<*numeric (day)*>

Sets the date of the internal clock. (DATE MM/DD/YY under (System))

Parameter	Description
<*numeric (year)*>	1900 to 2099
<*numeric (month)*>	1 to 12
<*numeric (day)*>	1 to 31

- Query Response

{*numeric (year)*} {*numeric (month)*} {*numeric (day)*} <new line><^END>

SETCTIME␣<*numeric (hour)*>,<*numeric (minute)*>,<*numeric (second)*>

Sets the time of the internal clock. (SETCTIME under (System))

Parameter	Description
<*numeric (hour)*>	0 to 23
<*numeric (minute)*>	0 to 59
<*numeric (second)*>	0 to 59

- Query Response

{*numeric (hour)*} {*numeric (minute)*} {*numeric (second)*} <new line><^END>

SETZ␣<*numeric*>[OHM]

Sets the characteristic impedance of the coaxial cable offset. (Network analyzer only) (SET Z0 under (Cal))

Parameter	Range	Unit
<*numeric*>	0.001 to 5000000 (=5M) (Network analyzer)	Ω
	50,75 (Spectrum analyzer)	

- Query Response

{*numeric*} <new line><^END>

SGTRK␣{OFF|ON|0|1}

Sets the signal tracking function ON or OFF. (Spectrum analyzer only) (SGNL TRACK ON off under Search)

Parameter	Description
OFF or 0	Signal tracking OFF
ON or 1	Signal tracking ON

- Query Response

{0|1} <new line><^END>

SIMFCHAR

Simulates frequency response of the equivalent circuit. (SIMULTE F-CHAR under Display; No query; Impedance analyzer only)

SING

Makes one sweep of the data and returns to the hold mode. (Instrument BASIC EXECUTE executable; SINGLE under Trigger; No query;)

When you execute this command by EXECUTE command of the instrument BASIC, the analyzer sweeps once and then back the control to the analyzer. The program waits the completion of sweep. You can use this method instead of detecting the sweep end by monitoring the status register to synchronize the program with the analyzer.

SMKR{1-7}␣{OFF|ON|0|1}

Displays the specified sub-marker at the point of the marker (ON), or erases the sub-marker (OFF). (SUB MKR {1-7 } under Marker)

Parameter	Description
OFF or 0	Sub-marker ON
ON or 1	Sub-marker OFF

- Query Response

{0|1} <new line><^END>

SMKRAUV{1-7}?

Outputs the auxiliary amplitude value of the measurement value at the sub-marker position. See "Marker Readout" in Chapter 8 for the auxiliary amplitude value of each display format. (SUB MKR {1-7} under (Marker); Query only)

- Query Response

{*numeric*} <new line><^END>

SMKRP{1-7}␣<*numeric*>

Moves the sub-marker to the specified data point number.

Parameter	Description
<*numeric*>	1 to "number of points" (If <*numeric*> is 0 or less than 0, it is set to 1. If <*numeric*> is greater than "number of points," it is set to "number of points.")

- Query Response

{*numeric*} <new line><^END>

SMKRPRM{1-7}␣<*numeric*>[HZ|DBM]

Moves the sub-marker to the specified sweep parameter value. (SUB MKR {1-7} under (Marker))

Parameter	Range	Unit
<*numeric*>	start value to stop value (0.001 resolution)	Hz (frequency) dBm (power)

- Query Response

{*numeric*} <new line><^END>

SMKRVAL{1-7}?

Outputs the primary part of the measurement value at the sub-marker position. (SUB MKR {1-7} under (Marker); Query only)

- Query Response

{*numeric*} <new line><^END>

SPAN␣<*numeric*>[HZ|DBM]

Sets the span of the sweep parameters. This command is not valid when the list sweeping mode is selected. ((Span))

When editing a list sweep table, the command sets the span of a segment. (SPAN under (Sweep))

Parameter	Range	Unit
<*numeric*>	0 to 510M (Network and impedance analyzers)	Hz (frequency)
<*numeric*>	0 to 510M, varies depending on the resolution bandwidth (when setting the sweep span in the spectrum analyzer mode)	
<*numeric*>	Varies depending on the resolution bandwidth (when editing a segment in the spectrum analyzer mode)	
	0 to 20 (Network and impedance analyzers)	dBm (power)

- Query Response

{*numeric*} <new line><^END>

SPECFWD{M|T}␣<*numeric (1)*>[,<*numeric (2)*>[, ... [,<*numeric (7)*>]

Enters the standard numbers for the forward match (THRU) or forward transmission (THRU) calibration. (Network analyzer only) (FWD.MATCH , FWD.TRANS. under (Cal); No query)

Parameter	Description
<*numeric*>	1 to 8

SPECIMP{**A**|**B**|**C**}⊔<*numeric 1*>**[**,<*numeric 2*>**[**, . . . **[**,<*numeric 7*>**]**

Enters the standard numbers for the first, second, or third standard class required for an impedance calibration. (`SPECIFY CLASS` under (Cal); No query. Impedance analyzer only.)

Parameter	Description
<*numeric*>	1 to 8

SPECRES{**I**|**P**}⊔<*numeric (1)*>**[**,<*numeric (2)*>**[**, . . . **[**,<*numeric (7)*>**]**

Enters the standard numbers for a response and isolation, or a response calibration. (Network analyzer only) (`RESPONSE & ISOL'N`, `RESPONSE` under (Cal); No query)

Parameter	Description
<*numeric*>	1 to 8

SPECREV{**M**|**T**}⊔<*numeric (1)*>**[**,<*numeric (2)*>**[**, . . . **[**,<*numeric (7)*>**]**

Enters the standard numbers for the reverse match (THRU) or reverse transmission (THRU) calibration. (Network analyzer only) (`REV.MATCH`, `REV.TRANS.` under (Cal); No query)

Parameter	Description
<*numeric*>	1 to 8

SPECS11{**A**|**B**|**C**}⊔<*numeric (1)*>**[**,<*numeric (2)*>**[**, . . . **[**,<*numeric (7)*>**]**

Enters the standard numbers for the first, second, or third standard class required for an S_{11} 1-port calibration. (Network analyzer only) (`SPECIFY: S11A`, `S11B`, `S11C` under (Cal); No query)

Parameter	Description
<*numeric*>	1 to 8

SPECS22{A|B|C}␣<*numeric (1)*>[,<*numeric (2)*>[, ... [,<*numeric (7)*>]

Enters the standard numbers for the first, second, or third standard class required for an S_{22} 1-port calibration. (Network analyzer only) (`SPECIFY: S22A`, `S22B`, `S22C` under (Cal); No query)

Parameter	Description
<*numeric*>	1 to 8

SPLD␣{OFF|ON|0|1}

Sets the dual channel display mode. (`SPLIT DISP ON off` under (Display))

Parameter	Description
`OFF` or `0`	Full-screen single graticule display
`ON` or `1`	Split display with two half-screen graticules

- Query Response

  {0|1} <new line><^END>

SQUI

Terminates editing a segment of the list sweep table. (`SEGMENT QUIT` under (Sweep); No query)

*SRE␣<*numeric*>

Sets the enable bits of the Status Byte Register.

Parameter	Description
<*numeric*>	0 to 255 (decimal expression of enable bits of the status byte register)

- Query Response

  {*numeric*} <new line><^END>

STAN{A-G}

Measures the calibration standard in the current standard class. (Network analyzer only)
(OPEN, SHORT, THRU, OPEN [ ], SHORT [ ], defined std {1-7} under Cal; No query)

STAR␣<*numeric*>[HZ|DBM]

Sets the start value of the sweep parameters. This command is not valid when the list sweeping mode is selected. (Start)

When editing a list sweep table, the command sets the start value of a segment.
(SEGMENT: START under Sweep)

Parameter	Range	Unit
<*numeric*>	0 to 510M (Spectrum analyzer)	Hz (frequency)
	10 to 510M (Network and impedance analyzers)	
	−50 to 15 (Network and impedance analyzers)	dBm (power)

- Query Response

  {*numeric*} <new line><^END>

∗STB?

Reads the Status Byte Register by reading the master summary status bit. (Query only)

- Query Response

  {*numeric*} <new line><^END>

STDD

Terminates the standard definition. (Network and impedance analyzers only)
(STD DONE (DEFINED) under Cal; No query)

STDT␣{OPEN|SHOR|LOAD|DELA|ARBI}

Defines the standard type. (Network and impedance analyzers only) (STD TYPE: OPEN, SHORT, LOAD, DELAY/THRU, ARBITRARY IMPEDANCE under Cal)

Parameter	Description
OPEN	OPEN
SHOR	SHORT
LOAD	LOAD
DELA	Transmission line of specified length
ARBI	LOAD with an arbitrary impedance

- Query Response

  {OPEN|SHOR|LOAD|DELA|ARBI} <new line><^END>

STOD{DISK|MEMO}

Sets the storage device. (STOR DEV[ ] under Save; No query; No equivalent SCPI command)

Parameter	Description
STODDISK	Flexible disk drive
STODMEMO	RAM disk memory

STOP␣<*numeric*>[HZ|DBM]

Sets the stop value of the sweep parameters. This command is not valid when the list sweeping mode is selected. (Stop)

When editing a list sweep table, the command sets the stop value of a segment. (SEGMENT: STOP under Sweep)

Parameter	Range	Unit
<*numeric*>	0 to 510M (Spectrum analyzer)	Hz (frequency)
	10 to 510M (Network and impedance analyzers)	
	−50 to 15 (Network and impedance analyzers)	dBm (power)

- Query Response

  {*numeric*} <new line><^END>

STORMDISK

Stores the contents of the RAM disk memory in the backup memory. (No query)

SVCO

Saves the modified version of the color set to the non-volatile memory. (SAVE COLORS under (Display); No query)

SWAI␣<*numeric*>

Specifies the time to keep the analyzer waiting for measurement start until the setting for all the segments except the first one is completed. This command is valid if frequency list sweeping is selected. (No query)

Parameter	Range	Unit
<*numeric*>	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	sec

SWET␣<*numeric*>[S]

Disables the automatic sweep time setting function and sets the sweep time with a specified value. (SWEEP TIME under (Sweep); Query only in the spectrum analyzer)

Parameter	Range	Unit
<*numeric*>	0(minimum measurement time) to 359999[1]	sec

1 The effective upper limit is n×400 sec, where n denotes the number of points. For n=801, it gives 320400 sec.

- Query Response

{*numeric*} <new line><^END>

SWETAUTO␣{OFF|ON|0|1}

Sets the automatic or manual sweep time. (SWEEP TIME AUTO man under (Sweep))

This command is not valid in the spectrum analyzer mode, in which sweep time is automatically set.

Parameter	Description
OFF or 0	Manual sweep time
ON or 1	Automatic sweep time

■ Query Response

{0|1} <new line><^END>

SWPT␣{LINF|LOGF|LIST|POWE}

Selects the sweep type. (SWEEP TYPE:LIN FREQ, LOG FREQ, LIST FREQ, POWER SWEEP under (Sweep))

Parameter	Description
LINF	Linear frequency
LOGF	Log frequency (Network and impedance analyzers only)
LIST	Frequency list
POWE	Power (Network and impedance analyzers only)

■ Query Response

{LINF|LOGF|LIST|POWE} <new line><^END>

T

Commands in Entry Block T (*T included)

TERI␣<*numeric*>[OHM]

Specifies the (arbitrary) impedance of the standard. (Network and impedance analyzers only) (TERMINAL IMPEDANCE under Cal; No query)

Parameter	Range	Unit
<*numeric*>	0 to 10000 (=10 k)	Ω

TESS?

Outputs the test set identifier. (Network analyzer only) (Query only)

- Query Response

{0|1} <new line><^END>

Parameter	Description
0	None
1	S-parameter test set

TINT␣<*numeric*>

Adjusts the hue of the specified display element. (TINT under Display; No equivalent SCPI command)

Parameter	Range	Unit
<*numeric*>	0 to 100	%

- Query Response

{*numeric*} <new line><^END>

TITL␣<*string*>

Sends the string to the title area on the display. (TITLE under Display)

Parameter	Description
<*string*>	up to 53 characters

- Query Response

{*string*} <new line><^END>

TMARG␣<*numeric*>

Specify the value for the top margin of printed paper. (TOP MARGIN under Copy)

Parameter	Range	Unit
<*numeric*>	0 to 5	inch

- Query Response

{*numeric*} <new line><^END>

TOPV␣<*numeric*>

Defines the top border of the display and adjusts the scale value. (TOP VALUE under Scale Ref; Network and impedance analyzers only.)

Parameter	Range	Unit
<*numeric*>	-1×10^9 to 1×10^9	y-axis unit

- Query Response

<*numeric*><new line><^END>

TRACK⊔{OFF|ON|0|1}

Sets the search tracking function ON or OFF. (`SRCH TRACK ON off` under (Search))

- Query Response

  {0|1} <new line><^END>

TRAD

Completes the transmission calibration of the full or one-path 2-port calibration. (Network analyzer only) (`TRANS. DONE` under (Cal); No query)

TRAN

Starts the transmission part of the full or one-path 2-port calibration. (Network analyzer only) (`TRANSMISSION` under (Cal); No query)

*TRG

Triggers the analyzer when the trigger mode is set to BUS trigger. (No query)

TRGEVE⊔{SWE|POIN}

Selects the trigger event mode. (Network and impedance analyzers only) (`TRIG EVENT [ ]` under (Trigger))

Parameter	Description
SWE	Trigger event on sweep
POIN	Trigger event on point[1]

1 Available only when the trigger source is the GPIB, Manual, or External trigger.

- Query Response

  {SWE|POIN} <new line><^END>

TRGP␣{POS|NEG}

Sets the trigger signal polarity of an external signal connected to the rear panel EXT TRIGGER input. (`TRIG PLRTY pos neg` under (Trigger))

Parameter	Description
POS	Positive trigger (low-to-high transition)
NEG	Negative trigger (high-to-low transition)

- Query Response

{POS|NEG} <new line><^END>

TRGS␣{INT|EXT|BUS|MAN|GAT}

Selects the trigger source, which is common to both channels. (`TRIGGER: [ ]` under (Trigger))

Parameter	Description
INT	Internal trigger
EXT	External trigger input from BNC on the rear panel
BUS	GPIB trigger
VID	Video trigger (Spectrum analyzer only)
MAN	Manual trigger
GAT	External gate trigger (Spectrum analyzer and option 1D6 only)

- Query Response

{INT|EXT|BUS|MAN|GAT} <new line><^END>

*TST?

Executes an internal self-test and returns the test result. (Query only)

- Query Response

{*numeric*} <new line><^END>

Parameter	Description
0	Pass
1	Fail

Commands in Entry Block U

USKEY

Show ON KEY LABELS of instrument BASIC. (No query)

The USKEY command is equivalent to executing the program shown below;

```
OUTPUT @Hp4396;"KEY 47"
OUTPUT @Hp4396;"KEY 0"
OUTPUT @Hp4396;"KEY 6"
```

Note Only instrument BASIC is available USKEY command.

V

Commands in Entry Block V

VBW␣<*numeric*>

Sets the bandwidth of the video bandwidth filter. (Spectrum analyzer only) (VIDEO BW under Bw/Avg)

Parameter	Description
<*numeric*>	RBW/1, RBW/3, RBW/10, RBW/30, RBW/100, RBW/300 (RBW: the current RBW setting)

- Query Response

{*numeric*} <new line><^END>

VBWT␣{LIN|LOG}

Selects either the linear or logarithmic video filter. (VBW TYPE [LIN] or [LOG] under Bw/Avg)

- Query Response

{LIN|LOG}<new line><^END>

VELOFACT␣<*numeric*>

Enters the velocity factor used by the analyzer to calculate the equivalent electrical length. (Network and impedance analyzers only) (VELOCITY FACTOR under Cal)

Parameter	Range	Unit
<*numeric*>	0 to 10	

- Query Response

{*numeric*} <new line><^END>

W

Commands in Entry Block W (∗W included)

∗WAI

Makes the analyzer wait until all previously sent commands are completed. (No query)

WIDSIN

Searches for the cutoff point on the trace within the current cutoff points. (Network and impedance analyzers only; SEARCH IN under (Search); No query)

WIDSOUT

Searches for the cutoff point on the trace outside of the current cutoff points. (Network and impedance analyzers only; SEARCH OUT under (Search); No query)

WIDT␣{OFF|ON|0|1}

Sets the bandwidth search feature ON or OFF. (Network and impedance analyzers only) (WIDTHS ON off under (Search))

Parameter	Description
OFF or 0	Bandwidth search feature OFF
ON or 1	Bandwidth search feature ON (calculates the center stimulus value, bandwidth, Q, insertion loss, and cutoff point deviation from the center of a bandpass or band reject shape on the trace.)

- Query Response

  {0|1} <new line><^END>

WIDV␣<*numeric*>

Sets an amplitude parameter that defines the start and stop points for a bandwidth search. (Network and impedance analyzers only) (WIDTH VALUE under Search)

Parameter	Range	Format
<*numeric*>	-1×10^{9} to 1×10^{9}	????

- Query Response

{*numeric*} <new line><^END>

WIDVTYPE␣{DIVS2|MULS2|DIV2|FIXed}

Select Maker Width Value Type. When you use `FIXed`, you must specify the bandwidth value by using `WIDV`. (MKRVAL/($\sqrt{}$2), MKRVAL*($\sqrt{}$2), MKRVAL/2, or FIXED VALUE under Search WIDTH [] WIDHT VALUE. Impedance analyzer only.)

- Query Response

`{DIVS2|MULS2|DIV2|FIX}`<new line><^END>

WOPEN␣<*string*>[,<*numeric*>]

If the specified file exists, this command makes it write-enabled; otherwise, creates a new file and makes it write-enabled. This command takes its arguments in a different way, depending on the file format. For a DOS format file you do not have to specify its file size, for a LIF format file you must. Specify the file size, 0 or greater, so that the file can contain the maximum number of bytes used. Note that only the BDAT type is available as the LIF file format.

The format and size of an existing file cannot be changed. Therefore, if you want to change them, delete the file itself using the `PURG` command and then create a new file using this command.

This command is used in combination with the `WRITE` command and the `CLOSE` commands, as shown in Figure R-2. (No query)

Parameter	Description
<*string*>	File name of up to 12 characters including its extention (for the LIF format, up to 10 characters)
<*numeric*>	File size (required only for the LIF format)

WRITE␣<*block*>

Writes data in a file that has been write-enabled using the `WOPEN` command. Written data must take the fixed length block format (see Figure R-1) defined in IEEE488.2. The maximum length of data is 16 Kbytes. If data is greater than 16 Kbytes, execute this command repeatedly to write it. (No query)

Generally, this command is used in combination with the `WOPEN` command and the `CLOSE` command, as shown in Figure R-2. (No query)

Parameter	Description
<*block*>	Data in the fixed length block format

X

Commands in Entry Block X

XMKRCENT

Applies a sweep parameter at the marker to the center value of the sweep parameters for the channel that is not active. (`XCH_MKR→CENTER` under [Marker→]; No query)

Note

MODIFY JAPANESE TEXT

XMKRSTAR

Applies a sweep parameter at the marker to the start value of the sweep parameters for the channel that is not active. (`SEGMENT: MKR→START` under [Sweep], or `XCH_MKR→START` under [Marker→]; No query)

XMKRSTOP

Applies a sweep parameter at the marker to the stop value of the sweep parameters for the channel that is not active. (`SEGMENT: MKR→STOP` under [Sweep], or `XCH_MKR→STOP` under [Marker→]; No query)

XMKRZM

Applies a sweep parameter at the marker to the center value of the sweep parameters for the channel that is not active, and changes the sweep parameter span value of the channel to "sweep parameter span × zooming aperture." (`XCH_MKR ZOOM` under [Marker→]; No query)

XPEAKCENT

Searches for a peak using the marker and applies a sweep parameter at the marker to the center value of the sweep parameters for the channel that is not active. (`PEAK→CENTER` under [Center], or `XCH_PEAK→CENTER` under [Marker→]; No query)

Z

Commands in Entry Block Z (Other commands included)

ZA

Selects the impedance analyzer mode. (IMPEDANCE ANALYZER under (Meas))

- Query Response

{0|1}<new line><^END>

Parameter	Description
0	Impedance analyzer mode is not selected.
1	Impedance analyzer mode is selected.

ZMAPER␣<*numeric*>

Sets the zooming aperture value as a percentage of the span. (ZOOMING APERTURE under (Marker→))

Parameter	Range	Unit
<*numeric*>	0 to 100 (relative to span. 0.01 resolution)	%

- Query Response

{*numeric*} <new line><^END>

Other Commands

Note The commands in the `PROGram` subsystem are related to HP Instrument BASIC. This command can be used from an external controller only.

:PROGram[:SELected]:DEFine␣<*block*>

Creates and downloads programs. The `DEFine` query uploads programs.

Parameter	Description
<*block*>	program

The <*block*> must be arbitrary block program data containing the lines of program code. The first line of <*block*> must be a header, which shows the program size. There are two formats for the header as follows:

`#0` Allows the `OUTPUT` statement to send program line until END is specified in the `OUTPUT` statement.

`#NMM.... M` Specifies the program size.

`N` specifies the number of digits that define the program size

`M.... M` is program size in byte (`N` digits)

Each line of the program must be separated by <CR> or <CR> <LF>. When the size of the <*block*> exceeds the amount of available memory in the instrument, the program lines are saved up to the point of memory overflow.

In the response to the `DEFine` query, the selected program and its size are returned. The selected program must be in either the paused or stopped state for the program to be uploaded. The <*block*> is uploaded as definite length arbitrary block response data. The program size is returned in the first line as the header, then program lines are returned.

- Query Response

  {*block*} <new line><^END>

:PROGram[:SELected]:DELete[:SELected]

Deletes the program in the BASIC editor of the analyzer. (No query)

:PROGram[:SELected]:DELete:ALL

Deletes the program in the BASIC editor of the analyzer. (No query)

:PROGram[:SELected]:EXECute␣<*string*>

Executes the program command. The program must be in either paused or stopped before the `EXECute` command is allowed. (No query)

Parameter	Description
<*string*>	Legal program command

:PROGram[:SELected]:MALLocate␣{<*numeric*>|DEFault}

Performs no function in the analzyer's HP Instrument BASIC.

:PROGram[:SELected]:NAME␣<*string*>

Performs no function in the analyzer's HP Instrument BASIC.

:PROGram[:SELected]:NUMBer␣<*string*>,<*numeric (1)*>[,<*numeric (2)*>[, ... [,<*numeric (n)*>]

Sets or queries the contents of numeric program variables and arrays in the program on the BASIC editor of the analyzer.

Parameter	Description
<*string*>	Name of an existing variable in the selected program (either character data or string data)
<*numeric*>	Value to be set the variable (use a comma to separate multiple entries)

- Query Response

  {*numeric 1*} [{*numeric 2*} [... [{*numeric n*}] ...]] <new line><^END> (n:the size of the array.)

:PROGram[:SELected]:STATe␣{RUN|PAUSe|STOP|CONTinue}

Sets or queries the state of the program in the BASIC editor of the analyzer. The table below defines the affect of setting the state to the specified state from each of the possible current states.

Desired State	**Current State**		
	RUN	**PAUSE**	**STOP**
RUN	error (−221)	RUN	RUN
CONT	error (−221)	RUN	error (−221)
PAUSE	PAUSE	PAUSE	STOP
STOP	STOP	STOP	STOP

- Query Response

{"RUN"|"PAUS"|"STOP"|"CONT"} <new line><^END>

:PROGram[:SELected]:STRing␣<*string (varname)*>,<*string (value 1)*>[,<*string (value 2)*> [, . . . [,<*string (value n)*>]

Sets or queries the contents of string program variables and arrays in the program in the BASIC editor of the analyzer. If a string value is too long it is truncated when stored in the program's variable.

Parameter	Description
<*string (varname)*>	Name of an existing variable in the selected program (either character data or string data).
<*string (value)*>	Value to be set the variable (use a comma to separate multiple entries)

- Query Response

{*string 1*} [{*string 2*} [. . . [{*string n*] . . .]] (n:the size of an array) <new line><^END>

:PROGram[:SELected]:WAIT

Causes no further commands or queries to be executed until the specified program exits from the RUN state. That is, the program is either stopped or paused. When used as a query command, it returns the status of the program.

- Query Response

  {1} <new line><^END>

  `1` is returned when the program is either stopped or paused.

Note

The following commands under the `EXPLicit` node perform the specified functions in the same manner as the corresponding commands under the `SELected` node. The `EXPLicit` commands are included in the analyzer's GPIB commands to maintain compatibility with other SCPI instruments. Therefore, you can use either the `EXPLicit` or the `SELected` commands for the analyzer. However, you should select one set and use it consistently to avoid confusion.

:PROGram:EXPLicit:DEFine⊔"PROG",<*string*>

See ":PROGram[:SELected]:DEFine⊔<*block*>".

:PROGram:EXPLicit:DELete⊔"PROG"

See ":PROGram[:SELected]:DELete[:SELected]".

:PROGram:EXPLicit:EXECute⊔"PROG",<*string*>

See ":PROGram[:SELected]:EXECute⊔<*string*>".

:PROGram:EXPLicit:MALLocate␣"PROG",{<*numeric*>|DEFault}

See ":PROGram[:SELected]:MALLocate␣{<*numeric*>|DEFault}".

:PROGram:EXPLicit:NAME␣"PROG",<*string*>

See ":PROGram[:SELected]:NAME␣<*string*>".

:PROGram:EXPLicit:NUMBer␣"PROG",<*string*> [,<*numeric*>]

See ":PROGram[:SELected]:NUMBer␣<*string*>,<*numeric (1)*>[,<*numeric (2)*>[, . . . [,<*numeric (n)*>]".

:PROGram:EXPLicit:STATe␣"PROG",{RUN|PAUSe|STOP|CONTinue}

See ":PROGram[:SELected]:STATe␣{RUN|PAUSe|STOP|CONTinue}".

:PROGram:EXPLicit:STRing␣"PROG",<*varname*>[,<*string*>]

See ":PROGram[:SELected]:STRing␣<*string (varname)*>,<*string (value 1)*>[,<*string (value 2)*> [, . . . [,<*string (value n)*>]".

:PROGram:EXPLicit:WAIT "PROG"

See ":PROGram[:SELected]:WAIT".

Status Notations and Error Messages

Status Notations

Displays the current status of various functions for the active channel. The following notations are used:

*	Sweep parameters changed: measured data in doubt until a complete fresh sweep has been taken.
`P`	RF output is ON (zero span in spectrum analyzer mode only).
`Cor`	Error correction is ON (network analyzer mode and impedance analyzer mode). Level correction is ON (spectrum analyzer mode only).
`C2`	Two-port error correction is ON (network analyzer mode only).
`Cmp`	Fixture compensation is ON(impedance analyzer mode only).
`C?`	Sweep parameters have changed[1] and interpolated error correction is ON (network analyzer mode and impedance analyzer mode).
`C2?`	Sweep parameters have changed[1] and interpolated two-port correction is ON (network analyzer mode only).
`Cm?`	Sweep parameters have changed[1] and interpolated fixture compensation is ON (impedance analyzer mode only).
`C!`	Sweep parameters have changed[2] and extrapolated error correction is ON (network analyzer mode and impedance analyzer mode).
`C2!`	Sweep parameters have changed[2] and extrapolated two-port correction is ON (network analyzer mode only).
`Cm!`	Sweep parameters have changed[2] and extrapolated fixture compensation is ON (impedance analyzer mode only).
`Cm*`	Fixture compensation is ON when error correction is `C?` or `C!` (impedance analyzer mode only).
`Del`	Electrical delay, port extension, or phase offset has been added or subtracted (network analyzer mode and impedance analyzer mode).
`Neg`	Negative peak detection is ON (spectrum analyzer mode only).
`Smp`	Sample detection is ON (spectrum analyzer mode only).
`Avg`	Sweep-by-sweep averaging is ON. The averaging count is shown below.
`Max`	Maximum hold is ON.
`Min`	Minimum hold is ON.
`G*`	Data math Gain is ON.
`-O`	Data math Offset is ON.
`G&O`	Data math Gain is ON and data math Offset is ON.
`D-M`	Data math **(Data Trace - Memory Trace)** is ON.
`D+M`	Data math **(Data Trace + Memory Trace)** is ON.
`D/M`	Data math **(Data Trace / Memory Trace)** is ON.
`Hld`	Hold sweep.
↑	Sweep indicator. (When sweep time is longer than 2 seconds, it appears on the trace).
`ext`	Waiting for external trigger (BNC in rear panel).
`man`	Waiting for manual trigger.
`bus`	Waiting for GPIB trigger.
`Svc`	A service mode is turned on. If this notation is shown, the measurement data will be out of specifications. (See *Service Manual*.)

1 Frequency span reduced, etc.

2 Frequency span expanded, etc.

Note No status notation is displayed when Gate trigger is used.

This section lists the error messages that are displayed on the analyzer display or transmitted by the instrument over GPIB. Each error message is accompanied by an explanation, and suggestions are provided to help in solving the problem. Where applicable, references are provided to the related chapter of the appropriate manual.

When displayed, error messages are preceded with the word "CAUTION:." That part of the error message has been omitted here for the sake or brevity. Some messages without the "CAUTION:" are for information only, and do not indicate an error condition. The messages are listed first in alphabetical order because the displayed messages do not contain the message number. The messages are then listed in numerical order to make them easier to find if they are read over the GPIB.

Error Messages in Numerical Order

+0 **No error**

The error queue is empty. Every error in the queue has been read (OUTPERRO? query) or the queue was cleared by power-on or the *CLS command.

1 **CAN'T SET RBW AUTO IN ZERO SPAN**

The RBW AUTO mode cannot be selected in the zero span. The RBW must be specified manually in the zero span. (spectrum analyzer mode only).

10 **ADDITIONAL STANDARDS NEEDED**

Error correction for the selected calibration class cannot be computed until all the necessary standards have been measured.

11 **CALIBRATION REQUIRED**

No valid calibration coefficients were found when you attempted to turn calibration ON.

12 **NO CALIBRATION CURRENTLY IN PROGRESS**

The RESUME CAL SEQUENCE softkey is not valid unless a calibration is in progress. Start a new calibration.

13 **CALIBRATION ABORTED**

The calibration in progress was terminated due to a change of the active channel or stimulus parameters.

14 **NOT VALID FOR PRESENT TEST SET**

The calibration requested is inconsistent with the test set present. This message occurs in the following situations:

- A full 2-port calibration is requested with a test set other than an S-parameter test set.
- A one-path 2-port calibration is requested with an S-parameter test set (this procedure is typically used with a transmission/reflection test set).

15 **EXCEEDED 7 STANDARDS PER CLASS**

A maximum of seven standards can be defined for any class. See "Modifying Calibration Kits" in the *Function Reference*.

16 **CURRENT PARAMETER NOT IN CAL SET**

GPIB only. Correction is not valid for the selected measurement parameter.

17 **BACKUP DATA LOST**

Data checksum error on the battery backup memory has occurred. The battery is recharged for approximately 10 minutes after power was turned ON.

19 **UNEXPECTED DATA DETECTED: CAL ABORTED**

The signal measured for the level cal is not adequate for the calibration signal. (spectrum analyzer mode only.)

26 **PRINTER:not on, not connect, wrong address**

The printer does not respond to control. Check the supply to the printer, online status, sheets, and so on.

34 **NO VALID MEMORY TRACE**

If a memory trace is to be displayed or otherwise used, a data trace must first be stored to memory.

37 **DISPLAY BUFFER IS FULL**

The display buffer is filled with the overlay traces or traces drawn by IBASIC DRAW/MOVE commands, etc.

44 **OVERLOAD ON INPUT B**

The power level at one of the four receiver inputs exceeds a certain level greater than the maximum input level.

45 **OVERLOAD ON INPUT A**

The power level at one of the four receiver inputs exceeds a certain level greater than the maximum input level.

46 **OVERLOAD ON INPUT R**

The power level at one of the four receiver inputs exceeds a certain level greater than the maximum input level.

48 **PHASE LOCK LOOP UNLOCKED**

EXT REF Input of 10 MHz is not proper, or the instrument is needed to adjust or repair. Check the external reference signal first. Contact your nearest Agilent Technologies office for adjustment or repair.

50 **CONT POWER CHANGE >30dB MAY DAMAGE MECH SW**

RF output power switch is switching sweep by sweep, because RF power level or the input attenuator setting is different between two channels and the dual channel is turn on. To avoid premature wearing out of the output power switch and input attenuator switch, change trigger type to HOLD, SINGLE, or NUMBER of GROUPS to hold sweep after measurement required. Or turn off the dual channel, or set the power level and the input attenuator of both channels to the same setting.

51 **MEASUREMENT INVALID AT f<=(5*IFBW)**

This message will displayed when whole frequency measured is less than or equal to 1 MHz and IFBW is set to 10 kHz or 40 kHz because the network measurement performance is not warranted at frequency $\leq$1 MHz with 10 kHz or 40 kHz IFBW.

52 **CONT SWITCHING MAY DAMAGE RCVR ATTEN**

Input attenuator switch at input R, A, or B is switching sweep by sweep, because the attenuator setting of one of the inputs is different between two channels and the dual channel is turn on. To avoid premature wearing out of the input attenuator switch, change trigger type to HOLD, SINGLE, or NUMBER of GROUPS to hold sweep after measurement required. Or turn off the dual channel, or set the input attenuator of both channels to the same setting.

54 **TOO MUCH DATA**

Either there is too much binary data to send to the analyzer when the data transfer format is FORM 2, FORM 3 or FORM 5, or the amount of data is greater than the number of points.

55 **NOT ENOUGH DATA**

The amount of data sent to the analyzer is less than that expected (*GPIB only*).

56 **OPTION NOT INSTALLED**

This error occurs when an GPIB command which is optional command is sent and the analyzer is not installed the option (*GPIB only*). Please confirm options installed to the analyzer using `*OPT?` command.

64 TOO MANY SEGMENTS

The maximum number of segments for the limit line table is 18.

74 CURRENT EDITING SEGMENT SCRATCHED

The current editing segment for the list table and the limit line is scratched when the following cases occur (*GPIB only*) :

- When `EDITLIST` (edit list table) command is received while editing a segment for the list table.
- When `EDITLIML` (edit limit line) command is received while editing a segment for the limit line.

Send `LIMSDON` (limit segment done) or `SDON` (segment done) to terminate editing segment.

75 COMMAND IGNORED - SEGMENT NOT DONE YET

The GPIB command the analyzer received is ignored, because the segment is editing (*GPIB only*). Send `LIMSDON` (limit segment done) or `SDON` (segment done) to terminate editing segment.

76 SEGMENT START/STOP OVERLAPPED

Segments are not allowed to be overlapped. Reenter appropriate value for start or stop value of segments to avoid that segment is not overlapped.

77 TOO MANY SEGMENTS OR POINTS

Frequency list mode is limited to 31 segments or 801 points.

78 TOO SMALL POINTS OR TOO LARGE STOP

STOP+SPAN/(NOP−1) is out of sweep range. Increase NOP or change STOP value to lower frequency to avoid this error.

82 CAN'T CHANGE- ANOTHER CONTROLLER ON BUS

The analyzer cannot assume the mode of system controller until the active controller is removed from the bus or relinquishes the bus.

83 FORMAT NOT VALID FOR MEASUREMENT

The conversion function except the 1/S and the multiple phase modes is not valid for the Smith, admittance, and SWR formats.

84 ANALYZER TYPE MISMATCH

The analyzer receives a command that is not available for the current analyzer type. Please confirm GPIB command or change analyzer type before sending the command.

93 **NO DATA TRACE**

The `MARKER ON [DATA]` is selected when the data trace is not displayed.

94 **NO MEMORY TRACE**

The `MARKER ON [MEMORY]` is selected when the memory trace is not displayed.

95 **NO MARKER DELTA - SPAN NOT SET**

The `MKRΔ→SPAN` softkey requires that delta marker mode be turned ON.

96 **NO MARKER DELTA - RANGE NOT SET**

The `MKRΔ→SEARCH RNG` softkey requires that delta marker is turned ON.

97 **CAN'T CHANGE WHILE DATA MATH ON**

The setting cannot be changed when the data math function is used.

98 **NO ACTIVE MARKER**

The marker→ command cannot be execute when no marker is displayed on the screen. Turn on the marker before executing the marker→ commands.

99 **CAN'T CHANGE WHILE DUAL CHAN OFF**

The Cross channel cannot be turned on when dual channel is off. Turn on the dual channel before the cross channel is turned on.

100 **NO FIXED DELTA MARKER**

The `FIXED ΔMKR VALUE` and `FIXED ΔMKR AUX VALUE` softkey requires that fixed delta marker is turned ON.

110 **SAVE ERROR**

A serious error, for example physically damaged disk surface, is detected on saving a file.

111 **RECALL ERROR: INSTR STATE PRESET**

A serious error, for example corrupted data, is detected on recalling a file, and this forced the analyzer to be PRESET.

112 **INVALID FILE NAME**

GPIB only. The file name for the RECALL, PURGE, or RE-SAVE function must have a "`_D`" or "`_S`" extension for LIF format.

113 **NO STATE/DATA FILES ON DISK**

There are no files on the flexible disk with extensions, "`_D`" or "`_S`" for LIF format, or "`STA`" or "`.DTA`" for DOS format.

114 CAN'T SAVE GRAPHICS WHEN COPY IN PROGRESS

If you attempt to save graphics when a print is in progress, this error message is displayed.

115 LIF-DOS COPY NOT ALLOWED

If you try to copy a file between the memory disk and the flexible disk when the format of the memory disk is different from the format of the flexible disk, this message is displayed.

116 NO STATE/DATA FILES ON MEMORY

There are no files on the memory disk with extensions, "_D" or "_S" for LIF format, or ".STA" or ".DTA" for DOS format.

117 DUPLICATE FILE EXTENSION

The extension name entered is already used for other file types. Use other extension name.

119 NO DATA TRACE DISPLAYED

The SCALE FOR [DATA] is selected when the data trace is not displayed.

120 NO MEMORY TRACE DISPLAYED

The SCALE FOR [MEMORY] is selected when the memory trace is not displayed.

124 LIST TABLE EMPTY OR INSUFFICIENT TABLE

The frequency list is empty. To implement the list frequency mode, add segments to the list table.

126 CAN'T CHANGE NUMBER OF POINTS

The number of points of the spectrum analyzer mode cannot be to change manually, except in zero span.

127 CAN'T SET SWEEP TIME AUTO IN ZERO SPAN

The automatic sweep time cannot be in zero span of the spectrum analyzer mode. (The network analyzer mode allows that the automatic sweep time is turned on.)

128 SPAN = 0 ONLY

The setup must be zero span and spectrum analyzer mode when turning on the RF OUTPUT.

131 FREQUENCY SWEEP ONLY

The sweep type must be frequency sweep when the center step size is set.

133 CAN'T CHANGE ON LIST SWEEP

When list sweep is selected, the following parameters are not allowed to be changed:

- CENTER, SPAN, START, STOP
- NOP
- IFBW or RBW
- POWER
- DC SOURCE

Modify the list table to change these parameters in the list sweep.

134 CAN'T COUPLE IN CURRENT INPUTS

When one channel measures a ratio measurement, and the other one measures an absolute measurement (for example: A/R and B), `COUPLED CH` can not be turned on.

135 COUPLED CHAN - BETWEEN NA&NA OR ZA&ZA

The analyzer types of both channels must be the network analyzer mode or impedance analyzer mode when the coupled channel is turned on. It is not possible to turn the coupled channel on in spectrum analyzer mode.

136 DC SOURCE OVERLOAD

The DC SOURCE output is overloded.

137 DC CURRENT LIMIT OCCURED

The output current at DC SOURCE port is reached to an upper limit and the output voltage is reduced so that the current does not exceed the upper limit. This message appears when the DC SOURCE port is used in voltage control mode.

138 DC VOLTAGE LIMIT OCCURED

The output voltage at DC SOURCE port is reached to an upper limit and the output current is reduced so that the voltage does not exceed the upper limit. This message appears when the DC SOURCE port is used in current control mode.

141 INSUFFICIENT MEMORY

If a lot of tasks is executed at same time, memory might be insufficient for a while. (For example, running HP Instrument BASIC program, printing a screen, and sending or receiving data array by GPIB are required at same time.) Please wait until finishing some tasks then execute the next task.

146 ON POINT NOT ALLOWD FOR THE CURRENT TRIG

The trigger event mode cannot be changed to the ON POINT mode because the current trigger source setting does not allow the ON POINT mode. The ON POINT mode is available for only MANUAL, EXTERNAL, and BUS trigger sources of the network analyzer mode.

154 INVALID DATE

The date entered to set the real time clock is invalid. Reenter correct date.

173 ACTIVE/SYSTEM CONTROLLER REQUIRED

When the 4395A and its peripherals are controlled using the Instrument BASIC, you must set the GPIB system's control mode to the system controller mode.

184 NOT ALLOWED IN SVC MODE

The operation is not allowed in service mode.

193 POWER ON TEST FAILED

Power on test failed. Contact your nearest Agilent Technologies office.

267 COMPENSTATION REQUIRED

Compensation is required. Perform compansation to obtain compensation data.

268 NO COMPENSATION CURRENTLY IN PROGRESS

No compensation is currently in progress.

269 COMPENSATION ABORTED

Compensation data acquisition process is aborted.

270 COMPENSATION STD LIST UNDEFINED

Compensation standard list is undefined.

−100 Command error

This is a generic syntax error that the analyzer cannot detect more specific errors. This code indicates only that a command error, as defined in IEEE 488.2, 11.5.1.1.4, has occurred.

−101 Invalid character

A syntax element contains a character that is invalid for that type. For example, a header containing an ampersand (`SENSE&`).

−102 Syntax error

An unrecognized command or data type was encountered. For example, a string was received when the analyzer was not expecting to receive a string.

−103 Invalid separator

The parser was expecting a separator and encountered an illegal character. For example, the semicolon was omitted after a program message unit, `*RST:TRIG`.

−104 Data type error

The parser recognized an unallowed data element. For example, numeric or string data was expected but block data was encountered.

−105 GET not allowed

A Group Execute Trigger (GET) was received within a program message (see IEEE 488.2, 7.7).

−108 Parameter not allowed

More parameters were received than expected for the header. For example, the `*SRE` command only accepts one parameter, so receiving `*SRE 4,16` is not allowed.

−109 Missing parameter

Fewer parameters were received than required for the header. For example, the `*SRE` command requires one parameter, so receiving only `*SRE` is not allowed.

−110 Command header error

An error was detected in the header. This error message is used when the analyzer cannot detect the more specific errors described for errors −111 through −119.

−111 Header separator error

A character that is not a legal header separator was encountered while parsing the header. For example, no white space followed the header, thus `*SRE4` is an error.

−112 Program mnemonic too long

The header contains more than twelve characters (see IEEE 488.2, 7.6.1.4.1).

−113 Undefined header

The header is syntactically correct, but it is undefined for the analyzer. For example, `*XYZ` is not defined for the analyzer.

−114 Header Suffix out of range

The value of a numeric suffix attached to a program mnemonic makes the header invalid.

−120 Numeric data error

This error, as well as errors −121 through −129, are generated when parsing a data element that appears to be numeric, including the nondecimal numeric types. This particular error message is used if the analyzer cannot detect a more specific error.

−121 Invalid character in number

An invalid character for the data type being parsed was encountered. For example, an alpha character in a decimal numeric or a “9” in octal data.

−123 **Exponent too large**

The magnitude of the exponent was larger than 32000 (see IEEE 488.2, 7.7.2.4.1).

−124 **Too many digits**

The mantissa of a decimal numeric data element contains more than 255 digits excluding leading zeros (see IEEE 488.2, 7.7.2.4.1).

−128 **Numeric data not allowed**

A legal numeric data element was received, but the analyzer does not accept it in this position for a header.

−130 **Suffix error**

This error, as well as errors −131 through −139, are generated when parsing a suffix. This particular error message is used if the analyzer cannot detect a more specific error.

−131 **Invalid suffix**

The suffix does not follow the syntax described in IEEE 488.2, 7.7.3.2, or the suffix is inappropriate for the analyzer.

−134 **Suffix too long**

The suffix contained more than 12 characters (see IEEE 488.2, 7.7.3.4).

−138 **Suffix not allowed**

A suffix was encountered after a numeric element that does not allow suffixes.

−140 **Character data error**

This error, as well as errors −141 through −148, are generated when analyzing the syntax of a character data element. This particular error message is used if the analyzer cannot detect a more specific error.

−141 **Invalid character data**

Either the character data element contains an invalid character or the particular element received is not valid for the header.

−144 **Character data too long**

The character data element contains more than twelve characters (see IEEE 488.2, 7.7.1.4).

−148 **Character data not allowed**

A legal character data element was encountered where prohibited by the analyzer.

−150 **String data error**

This error, as well as errors −151 and −158, are generated when analyzing the syntax of a string data element. This particular error message is used if the analyzer cannot detect a more specific error.

−151 **Invalid string data**

A string data element was expected, but was invalid for some reason (see IEEE 488.2, 7.7.5.2). For example, an `END` message was received before the terminal quote character.

−158 **String data not allowed**

A string data element was encountered but was not allowed by the analyzer at this point in parsing.

−160 **Block data error**

This error, as well as errors −161 and −168, are generated when analyzing the syntax of a block data element. This particular error message is used if the analyzer cannot detect a more specific error.

−161 **Invalid block data**

A block data element was expected, but was invalid for some reason (see IEEE 488.2, 7.7.6.2). For example, an `END` message was received before the length was satisfied.

−168 **Block data not allowed**

A legal block data element was encountered but was not allowed by the analyzer at this point in parsing.

−200 **Execution error**

This is the generic syntax error that the analyzer cannot detect more specific errors. This code indicates only that an execution error as defined in IEEE 488.2, 11.5.1.1.5 has occurred.

−210 **Trigger error**

A trigger related error occurred. This error message is used when the analyzer cannot detect the more specific errors described for errors −211 through −219.

−211 **Trigger ignored**

A GET, `*TRG`, or triggering signal was received and recognized by the analyzer but was ignored because of analyzer timing considerations. For example, the analyzer was not ready to respond.

−213 **Init ignored**

A request for a measurement initiation was ignored as another measurement was already in progress.

−220 **Parameter error**

Indicates that a program data element related error occurred. This error message is used when the analyzer cannot detect the more specific errors described for errors −221 through −229.

−221 **Settings conflict**

A legal program data element was parsed but could not be executed due to the current device state (see IEEE 488.2, 6.4.5.3 and 11.5.1.1.5).

−222 **Data out of range**

A legal program data element was parsed but could not be executed because the interpreted value was outside the legal range as defined by the analyzer (see IEEE 488.2, 11.5.1.1.5).

−223 **Too much data**

A legal program data element of block, expression, or string type was received that contained more data than the analyzer could handle due to memory or related device-specific requirements.

−224 **Illegal parameter value**

Used where exact value, from a list of possibilities, was expected.

−225 **Data out of memory**

The analyzer has insufficient memory to perform the requested operation.

−230 **Data corrupt or stale**

Possibly invalid data. New reading started but not completed since last access.

−231 **Data questionable**

Indicates that measurement accuracy is suspect.

−240 **Hardware error**

Indicates that a legal program command or query could not be executed because of a hardware problem in the analyzer. Definition of what constitutes a hard ware problem is completely device-specific. This error message is used when the analyzer cannot detect the more specific errors described for errors −241 through −249.

−241 **Hardware missing**

A legal program command or query could not be executed because of missing analyzer hardware. For example, an option was not installed.

−250 **Mass storage error**

Indicates that a mass storage error occurred. This error message is used when the analyzer cannot detect the more specific errors described for errors −257.

−256 File name not found

A legal program command could not be executed because the file name on the device media was not found: for example, an attempt was made to read or copy a nonexistent file.

−257 File name error

Indicates that a legal program command or query could not be executed because the file name on the device media was in error. For example, an attempt was made to copy to a duplicate file name. The definition of what constitutes a file name error is device-specific.

−280 Program error

Indicates that a downloaded program-related execution error occurred. This error message is used when the analyzer cannot detect the more specific errors described for errors −281 through −289.

−281 Cannot create program

Indicates that an attempt to create a program was unsuccessful. A reason for the failure might include not enough memory.

−282 Illegal program name

The name used to reference a program was invalid. For example, redefining an existing program, deleting a nonexistent program, or in general, referencing a nonexistent program.

−283 Illegal variable name

An attempt was made to reference a nonexistent variable in a program.

−284 Program currently running

Certain operations dealing with programs may be illegal while the program is running. For example, deleting a running program might not be possible.

−285 Program syntax error

Indicates that a syntax error appears in a downloaded program. The syntax used when parsing the downloaded program is device-specific.

−286 Program runtime error

A program runtime error of the HP Instrument BASIC has occurred. To get a more specific error information, use the `ERRM$` or `ERRN` command of the HP Instrument BASIC.

−310 System error

Some error, termed "system error" by the analyzer, has occurred.

−311 Memory error

An error was detected in the analyzer's memory.

−330 **Self-test failed**

A self-test failed. Contact your nearest Agilent Technologies office or see the *Service Manual* for troubleshooting.

−350 **Queue overflow**

A specific code entered into the queue in lieu of the code that caused the error. This code indicates that there is no room in the queue and an error occurred but was not recorded.

−400 **Query errors**

This is the generic query error that the analyzer cannot detect more specific errors. This code indicates only that a query error as defined in IEEE 488.2, 11.5.1.1.7 and 6.3 has occurred.

−410 **Query INTERRUPTED**

A condition causing an interrupted query error occurred (see IEEE 488.2, 6.3.2.3). For example, a query followed by `DAB` or `GET` before a response was completely sent.

−420 **Query UNTERMINATED**

A condition causing an unterminated query error occurred (see IEEE 488.2, 6.3.2.2). For example, the analyzer was addressed to talk and an incomplete program message was received by the controller.

143 **FLOATING POINT ERROR OCCURED**

Indicate that a floating point error occured in the analyzer. Data processing may not be correct. This error message is used when an internal application was executed for illegal data sent from an external device, or when an internal software bug was detected. Contact your nearest Agilent Technologies office.

Index

E

F

G

H

I

K

L

REGIONAL SALES AND SUPPORT OFFICES

For more information about Agilent Technologies test and measurement products, applications, services, and for a current sales office listing, visit our web site: http://www.agilent.com/find/tmdir. You can also contact one of the following centers and ask for a test and measurement sales representative. *11/29/99*

United States:
Agilent Technologies
Test and Measurement Call Center
P.O.Box 4026
Englewood, CO 80155-4026
(tel) 1 800 452 4844

Canada:
Agilent Technologies Canada Inc.
5150 Spectrum Way
Mississauga, Ontario
L4W 5G1
(tel) 1 877 894 4414

Europe:
Agilent Technologies
Test & Measurement
European Marketing Organization
P.O.Box 999
1180 AZ Amstelveen
The Netherlands
(tel) (31 20) 547 9999

Japan:
Agilent Technologies Japan Ltd.
Call Center
9-1, Takakura-Cho, Hachioji-Shi,
Tokyo 192-8510, Japan
(tel) (81) 426 56 7832
(fax) (81) 426 56 7840

Latin America:
Agilent Technologies
Latin American Region Headquarters
5200 Blue Lagoon Drive, Suite #950
Miami, Florida 33126
U.S.A.
(tel) (305) 267 4245
(fax) (305) 267 4286

Australia/New Zealand:
Agilent Technologies Australia Pty Ltd
347 Burwood Highway
Forest Hill, Victoria 3131
(tel) 1-800 629 485 (Australia)
(fax) (61 3) 9272 0749
(tel) 0 800 738 378 (New Zealand)
(fax) (64 4) 802 6881

Asia Pacific:
Agilent Technologies
24/F, Cityplaza One, 1111 King's Road,
Taikoo Shing, Hong Kong
(tel) (852)-3197-7777
(fax) (852)-2506-9284