

# **HP-IB Command Reference**

**For the HP 8711 RF Network Analyzer**



**HP Part No. 08711-90110  
Printed in USA July 1993**

**Microwave Instrument Division**

© Copyright 1993, Hewlett-Packard Company

## HP-IB Command Reference

---

This document is a guide to HP-IB control of the HP 8711 RF Network Analyzer. Its purpose is to provide concise information about the operation of the instrument under HP-IB control. The reader should already be familiar with making measurements with the HP 8711 and with the general operation of HP-IB.

Related information can be found in the following references:

- Information on programming the HP 8711 is available in the *HP-IB Programming Guide* (or Chapter 9a of the *HP 8711 Operating Manual*).
- Information on making measurements with the HP 8711 is available in the *HP 8711 Operating Manual*.
- Information on using the HP-IB is available in the *Tutorial Description of the Hewlett-Packard Interface Bus* (HP literature no. 5952-0156).
- Information on the SCPI programming language is available in *A Beginner's Guide to SCPI* (HP literature no. 5010-7166).

### Table of Contents

Introduction to SCPI	9b-2
The Command Tree	9b-2
Sending Multiple Commands	9b-5
Command Abbreviation	9b-6
Implied Mnemonics	9b-7
Parameter Types	9b-8
Syntax Summary	9b-11
IEEE 488.2 Common Commands	9b-12
Menu Map with SCPI Commands	9b-14
Internal Measurement Arrays	9b-30
SCPI Command Summary	9b-33
SCPI Conformance Information	9b-49
SCPI Standard Commands	9b-49
Instrument Specific Commands	9b-52

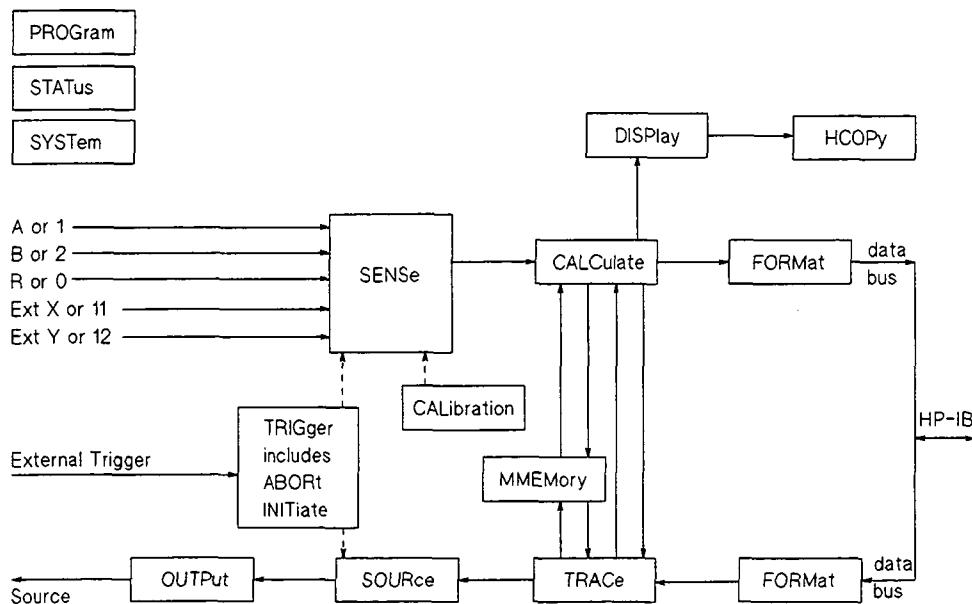
## Introduction to SCPI

Standard Commands for Programmable Instruments (SCPI) is a programming language designed specifically for controlling instruments by Hewlett-Packard and other industry leaders. SCPI provides commands that are common from one instrument to another. This elimination of “device specific” commands for common functions allows programs to be used on different instruments with very little modification.

SCPI was developed to conform to the IEEE 488.2 standard (replacing IEEE 728-1982). The IEEE 488.2 standard defines the syntax and data formats used to send data between devices, the structure of status registers, and the commands used for common tasks. For more information, refer to the IEEE standard itself. SCPI defines the commands used to control device-specific functions, the parameters accepted by these functions, and the values they return.

## The Command Tree

The SCPI standard organizes related instrument functions by grouping them together on a common branch of a command tree. Each branch is assigned a mnemonic to indicate the nature of the related functions. The HP 8711 has 16 major SCPI branches or **subsystems**. See Figure 9b-1 for a model of how these subsystems are organized to manage the measurement and data flow for the HP 8711.



**Figure 9b-1. Measurement and Data Flow of the HP 8711**

The HP 8711's major SCPI subsystems and their functions are described below.

<b>ABORt</b>	Aborts any sweep in progress.
<b>CALCulate</b>	Configures post-measurement processing of the measured data (such as marker and limit testing functions).
<b>CALibration</b>	Controls zeroing the broadband diode detectors.
<b>DISPlay</b>	Controls the display of measurement data, annotation and user graphics.
<b>FORMAT</b>	Controls the format of data transfers over the HP-IB (for more information about HP-IB data transfer refer to the <i>HP-IB Programming Guide</i> ).
<b>HCOPY</b>	Controls hardcopy (printer and plotter) output.
<b>INITiate</b>	Controls the triggering of sweeps.
<b>MMEMory</b>	Controls mass storage of instrument states and data (disk and internal memory interface functions).
<b>OUTPut</b>	Turns on/off the source output power (power to the device under test).
<b>PROGram</b>	Interfaces IBASIC programs and commands with an external controller (for more information on IBASIC programming refer to <i>Using HP Instrument BASIC with the HP 8711</i> — HP part no. 08711-90008).
<b>SENSe</b>	Configures parameters (such as the frequency and measurement parameters) related to the sweep and the measured signal (from the device under test). This subsystem also controls the narrowband calibration routines.
<b>SOURce</b>	Controls the RF output power level of the source (power to the device under test).
<b>STATus</b>	Contains the commands for using the SCPI status registers (for more information about using the status registers refer to the <i>HP-IB Programming Guide</i> ).
<b>SYSTem</b>	Contains miscellaneous system configuration commands (such as I/O port, clock and softkey control).
<b>TRACE</b>	Interfaces with the internal data arrays (functions such as data transfer and trace memory).
<b>TRIGger</b>	Controls the source of the sweep triggering.

When many functions are grouped together on a particular branch, additional branching is used to organize these functions into groups that are even more closely related. The branching process continues until each analyzer function is assigned to its own branch. For example, the function that turns on and off the marker tracking feature is assigned to the TRACKING branch of the FUNCTION branch of the MARKER branch of the CALCULATE subsystem. The command looks like this:

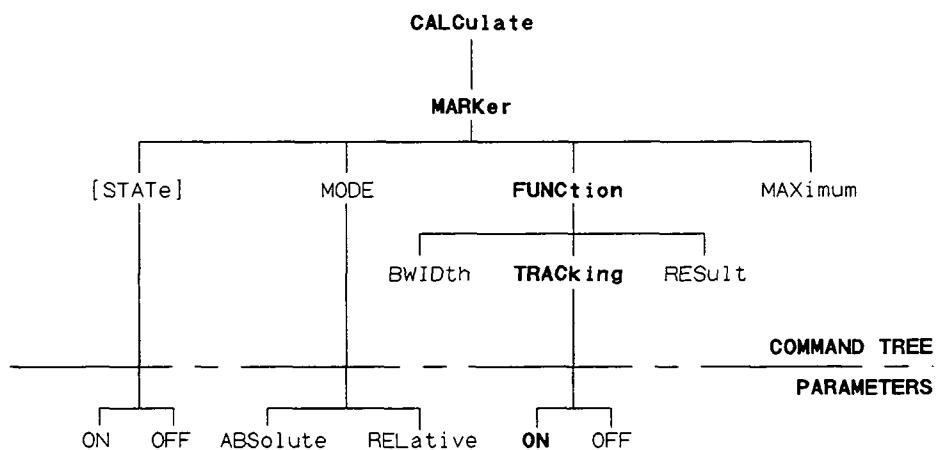
CALCULATE:MARKER:FUNCTION:TRACKING ON

---

**Note**

Colons are used to indicate branching points on the command tree. A parameter is separated from the rest of the command by a space.

---



**Figure 9b-2. Partial Diagram of the CALCULATE Subsystem Command Tree**

## Sending Multiple Commands

Multiple commands can be sent within a single program message by separating the commands with semicolons. For example, the following program message — sent within an HP BASIC OUTPUT statement — turns on the marker reference and moves the main marker to the highest peak on the trace:

```
OUTPUT 716;"CALCULATE:MARKER:MODE RELATIVE;:CALCULATE:MARKER:MAXIMUM"
```

One of the analyzer's command parser main functions is to keep track of a program message's position in the command tree. This allows the previous program message to be simplified. Taking advantage of this parser function, the simpler equivalent program message is:

```
OUTPUT 716;"CALCULATE:MARKER:MODE RELATIVE;MAXIMUM"
```

In the first version of the program message, the semicolon that separates the two commands is followed by a colon. Whenever this occurs, the command parser is reset to the base of the command tree. As a result, the next command is only valid if it includes the entire mnemonic path from the base of the tree.

In the second version of the program message, the semicolon that separates the two commands is not followed by a colon. Whenever this occurs, the command parser assumes that the mnemonics of the second command arise from the same branch of the tree as the final mnemonic of the preceding command. MODE, the final mnemonic of the first command, arises from the MARKER branch. So MAXIMUM, the first mnemonic of the second command is also assumed to arise from the MARKER branch.

The following is a longer series of commands — again sent within HP BASIC OUTPUT statements — that can be combined into a single program message:

```
OUTPUT 716;"CALCULATE:MARKER:STATE ON"  
OUTPUT 716;"CALCULATE:MARKER:MODE RELATIVE"  
OUTPUT 716;"CALCULATE:MARKER:MAXIMUM"  
OUTPUT 716;"CALCULATE:MARKER:FUNCTION:TRACKING ON"
```

The single program message is:

```
OUTPUT 716;"CALCULATE:MARKER:STATE ON;MODE RELATIVE;MAXIMUM;FUNCTION:TRACKING  
ON"
```

## Command Abbreviation

Each command mnemonic has a long form and a short form. The short forms of the mnemonics allow you to send abbreviated commands. Only the exact short form or the exact long form is accepted.

The short form mnemonics are created according to the following rules:

- If the long form mnemonic has four characters or less, the short form is the same as the long form. For example, DATA remains DATA.
- If the long form mnemonic has more than four characters and the fourth character is a consonant, the short form consists of the first four characters of the long form.  
For example, CALCULATE becomes CALC.
- If the long form mnemonic has more than four characters and the fourth character is a vowel, the short form consists of the first three characters of the long form. For example, LIMIT becomes LIM.

---

### Note

The short form of a particular mnemonic is indicated by the use of UPPER-CASE characters in this manual.



SCPI is not case sensitive so any mix of upper- and lower-case lettering can be used when sending commands to the analyzer.

---

If the rules listed in this section are applied to the last program message in the preceding section, the statement:

OUTPUT 716;"CALCULATE:MARKER:STATE ON;MODE RELATIVE;MAXIMUM;FUNCTION:TRACKING ON"

becomes:

OUTPUT 716;"CALC:MARK:STAT ON;MODE REL;MAX;FUNC:TRAC ON"

## Implied Mnemonics

Some mnemonics can be omitted from HP-IB commands without changing the effect of the command. These special mnemonics are called implied mnemonics, and they are used in many subsystems. In addition to entire mnemonics, variable parts of some mnemonics may also be implied. These are usually a number indicating a particular measurement channel, marker, or similar choice.

---

**Note**

When a number is not supplied for an implied variable, a default choice is assumed — this choice is always 1.



The INITIATE subsystem contains both the implied mnemonic IMMEDIATE at its first branching point and an implied variable for the measurement channel. The command to trigger a new sweep is shown in the "SCPI Command Summary" as:

OUTPUT 716;"INITiate[1|2] [:IMMEDIATE].

Any of the following forms of the command can be sent to the analyzer (using HP BASIC) to trigger a new sweep on channel 1:      OUTPUT 716;"INITIATE1:IMMEDIATE"

OUTPUT 716;"INITIATE:IMMEDIATE"

OUTPUT 716;"INITIATE1"

OUTPUT 716;"INITIATE"

If the sweep is to be triggered for measurement channel 2, the channel number MUST be specified.

OUTPUT 716;"INITIATE2:IMMEDIATE"

OUTPUT 716;"INITIATE2"

---

**Note**

Implied mnemonics are identified by square brackets [:MNEMONIC] in this manual. Implied variables are identified in a similar manner :MNEMONIC[1|2].



## Parameter Types

Parameters are used in many commands. The HP 8711 uses several types of parameters with different types of commands and queries. When a parameter is sent with a SCPI command it must be separated from the command by a space. If more than one parameter is sent they are separated from each other by commas.

### Numeric Parameters

Most subsystems use numeric parameters to specify physical quantities. Simple numeric parameters accept all commonly used decimal representations of numbers, including optional signs, decimal points, and scientific notation. If an instrument setting programmed with a numeric parameter can only assume a finite number of values, the instrument automatically rounds the parameter. In addition to numeric values, all numeric parameters accept MAXimum and MINimum as values (note that MAXimum and MINimum can be used to set or query values).

<num> is used in this document to denote a numeric parameter.

An example is the command to set the stop frequency for a measurement. Both of the following commands set the stop frequency to its maximum possible value of 1300 MHz.

```
OUTPUT 716;"SENSE1:FREQUENCY:STOP 1300 MHZ"  
OUTPUT 716;"SENSE1:FREQUENCY:STOP MAX"
```

**Query Response.** When a numeric parameter is queried the number is returned in one of the three numeric formats.

- NR1 Integers (such as +1, 0, -1, 123, -12345)
- NR2 Floating point number with an explicit decimal point (such as 12.3, +1.234, -0.12345)
- NR3 Floating point number in scientific notation (such as +1.23E+5, +123.4E-3, -456.789E+6)

An example is the response to a query of the stop frequency after executing the above commands (this response is of the NR3 type).

```
OUTPUT 716;"SENSE1:FREQUENCY:STOP?" returns the value 1.3E+9.
```

## Character Parameters

Character parameters (sometimes referred to as discrete parameters) consist of ASCII characters. They are typically used for program settings that have a finite number of values.

These parameters use mnemonics to represent each valid setting. They have a long and a short form which follow the same rules as command mnemonics.

<char> is used in this document to denote a character parameter.

An example of a command using a character parameter is the command that selects the format in which the measurement data is displayed.

```
OUTPUT 716;"CALCULATE1:FORMAT MLOGARITHMIC"
```

**Query Response.** When a character parameter is queried the response is always the short form of the mnemonic that represents the current setting. An example is the response to a query of the data format after executing the above command.

```
OUTPUT 716;"CALCULATE1:FORMAT?" returns the value MLOG.
```

## Boolean Parameters

Boolean parameters are used for program settings that can be represented by a single binary condition. Commands that use this type of parameter accept the values ON (or 1) and OFF (or 0).

<ON|OFF> is used in this document to denote a boolean parameter.

An example of a command that uses a boolean parameter is the command that makes the analyzer continuously trigger (or stop triggering) measurements.

```
OUTPUT 716;"INITIATE:CONTINUOUS ON"
```

A special group of commands uses boolean parameters to control automatic functions of the instrument, such as automatically selecting the fastest possible sweep speed. With these automatic functions an additional value is available for the parameter. This value ONCE causes the function to execute once before turning off.

**Query Response.** The response when a boolean parameter is queried is a single NR1 number indicating the state 1 for on or 0 for off. An example is the response to a query on the sweep trigger status after executing the above command.

```
OUTPUT 716;"INITIATE:CONTINUOUS?" returns the value 1.
```

## **String Parameters**

String parameters can contain virtually any set of ASCII characters. The string must begin with a single quote ( ' ) or a double quote ( " ) and end with the same character (called the delimiter). The delimiter can be included as a character (embedded) inside the string by typing it twice without any characters in between.

<string> is used in this document to denote a string parameter.

A command that uses a string parameter is the one used to configure the analyzer to measure a particular type of device and parameter.

```
OUTPUT 716;"CONFIGURE 'FILTER:TRANSMISSION'"
```

Some of the string parameters used by the HP 8711, like 'FILTER:TRANSMISSION' in the example above, follow the same rules that apply to mnemonics. They may have branching ('FILTER:REFLECTION' is a related command) and abbreviated versions.

**Query Response.** The response when a string parameter is queried is a string. The only difference is that the response string will only use double quotes as delimiters. Embedded double quotes may be present in string response data. When the string follows the "SCPI" mnemonic rules, the string returned in response to a query is in the abbreviated form. An example is the response to the configuration status of the analyzer (after executing the last command).

```
OUTPUT 716;"CONFIGURE?" returns the value "FILT:TRAN".
```

## **Block Parameters**

Block parameters are typically used to transfer large quantities of related data (like a data trace). Blocks can be sent as definite length blocks or indefinite length blocks — the instrument will accept either form. For more information on block data transfers refer to the "Transferring Data" section of the *HP-IB Programming Guide*.

<block> is used in this document to denote a block parameter.

## Syntax Summary

The following conventions are used throughout this manual whenever SCPI mnemonics are being described.

- **angle brackets** (< >) are used to enclose required parameters within a command or query. The definition of the variable is usually explained in the accompanying text.
- **square brackets** ([ ]) are used to enclose implied or optional parameters within a command or query.
- **UPPERlower case** characters are used to indicate the short form (upper-case) of a given mnemonic. The remaining (lower-case) letters are the rest of the long form mnemonic.

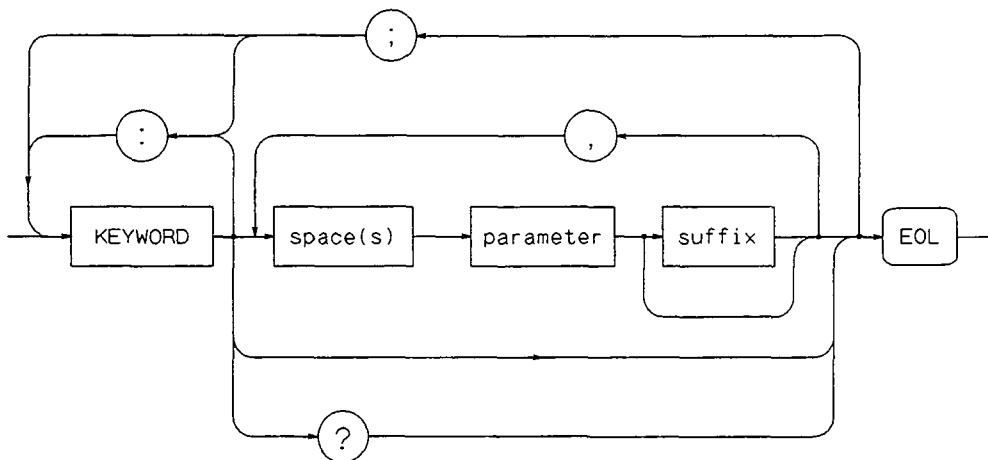


Figure 9b-3. SCPI Command Syntax

The following elements have special meanings within a SCPI program message (or combination or mnemonics).

- **colon (:**) — When a command or query contains a series of mnemonics, they are separated by colons. A colon immediately following a mnemonic tells the command parser that the program message is proceeding to the next level of the command tree. A colon immediately following a semicolon tells the command parser that the program message is returning to the base of the command tree.
- **semicolon (;**) — When a program message contains more than one command or query, a semicolon is used to separate them from each other.
- **comma (,**) — A comma separates the data sent with a command or returned with a response.
- **space ( )** — One space is required to separate a command or query from its data (or parameters). Spaces are not allowed inside a command or query.

---

## IEEE 488.2 Common Commands

IEEE 488.2 defines a set of common commands. All instruments are required to implement a subset of these commands, specifically those commands related to status reporting, synchronization and internal operations. The rest of the common commands are optional. The following list details which of these IEEE 488.2 common commands are implemented in the HP 8711 and the response of the analyzer when the command is received.

*CLS	Clears the instrument Status Byte by emptying the error queue and clearing all event registers, also cancels any preceding *OPC command or query (does not change the enable registers or transition filters).
*ESE <num>	Sets bits in the Standard Event Status Enable Register — current setting is saved in non-volatile memory.
*ESE?	Reads the current state of the Standard Event Status Enable Register.
*ESR?	Reads and clears the current state of the Standard Event Status Register.
*IDN?	Returns a string that uniquely identifies the analyzer. The string is of the form "HEWLETT-PACKARD,8711A,<serial number>,<software revision>"
*LRN?	This returns a string of device specific characters that, when sent back to the analyzer will restore the instrument state active when *LRN? was sent. Data formatting (ENTER USING "-K" in HP BASIC) or a similar technique should be used to ensure that the transfer does not terminate on a carriage return or line feed (both $C_R$ and $L_F$ are present in the learn string as part of the data).
*OPC	Operation complete command. The analyzer will generate the OPC message in the Standard Event Status Register when all pending overlapped operations have been completed (e.g. a sweep, or a preset). For more information about overlapped operations refer to the <i>HP-IB Programming Guide</i> .
*OPC?	Operation complete query. The analyzer will return an ASCII "1" when all pending overlapped operations have been completed.
*OPT?	Returns a string identifying the analyzer's option configuration. The string is of the form "1E1,1C2". The options are identified by the following: <ul style="list-style-type: none"><li>• 1EC 75 ohm</li><li>• 1E1 60 dB step attenuator</li><li>• 1C2 IBASIC</li></ul>
*PCB <num>	Sets the pass-control-back address (the address of the controller before a pass control is executed).
*PSC <num>	Sets the state of the Power-on Status Clear flag — flag is saved in non-volatile memory. This flag determines whether or not the Service Request enable register and the Event Status enable register are cleared at power-up.

- \*RST**      Executes a device reset and cancels any pending \*OPC command or query. The contents of the instrument's nonvolatile memory are *not* affected by this command.
- This command is different from the front panel **PRESET** function in the state of the commands (and their reset states) listed below. The preset instrument state is described in the Reference section of the *HP 8711 Operating Manual*.
- **INITiate:CONTinuous**      = OFF
  - **OUTPut[:STATE]**      = OFF
  - **CALibration:ZERO:AUTO**      = OFF
  - **SENSe:CORRection[:STATE]**      = OFF
  - **SENSe:SWEep:POINTS**      = MAX
  - **SOURce:POWer**      = MIN
- \*SRE <num>**      Sets bits in the Service Request Enable Register — current setting is saved in non-volatile memory.
- \*SRE?**      Reads the current state of the Service Request Enable Register.
- \*STB?**      Reads the value of the instrument Status Byte. This is a non-destructive read, the Status Byte is cleared by the \*CLS command.
- \*TST?**      Performs and returns the result of a complete self-test. An ASCII 0 indicates no failures found. Any other character indicates a specific self-test failure. See the *HP 8711 Service Manual* (HP part no. 08711-90003) for further information.
- \*WAI**      Prohibits the instrument from executing any new commands until all pending overlapped commands have been completed.

## Menu Map with SCPI Commands

This section contains a map of all the softkey menu choices in the HP 8711A. Each softkey is shown with its corresponding SCPI command (if one exists). Hardkeys are indicated with the **Hardkey** notation, softkeys are shown as **Softkeys**. SCPI commands are all shown in their short form.

Some commands (such as source settings) have mnemonics that specify the channel in use. These mnemonics are shown as SENS[1|2]: ... indicating that either channel could be used. The actual mnemonic entered would be SENS1: ... for setting channel 1 or SENS2: ... for channel 2. Mnemonics for keys that toggle between two states are shown as ... ON|OFF.

<num> and <string> refer to parameter types described in the "Parameter Types" section. <string> parameters are typically enclosed in single quotes ('the string data').

KEYSTROKES	SCPI COMMAND
<b>PRESET</b>	SYST:PRES;*WAI
<b>BEGIN</b>	
<b>Amplifier</b>	
<b>Transmissn</b>	CONF 'AMPL:TRAN';*WAI
<b>Reflection</b>	CONF 'AMPL:REFL';*WAI
<b>Power</b>	CONF 'AMPL:POW';*WAI
<b>Filter</b>	
<b>Transmissn</b>	CONF 'FILT:TRAN';*WAI
<b>Reflection</b>	CONF 'FILT:REFL';*WAI
<b>Broadband Passive</b>	
<b>Transmissn</b>	CONF 'BBAN:TRAN';*WAI
<b>Reflection</b>	CONF 'BBAN:REFL';*WAI
<b>Mixer</b>	
<b>Conversion Loss</b>	CONF 'MIX:CLOS';*WAI
<b>Reflection</b>	CONF 'MIX:REFL';*WAI

KEYSTROKES	SCPI COMMAND
CHAN 1   CHAN 2	SENS[1 2]:STAT ON;*WAI
Transmissn	SENS[1 2]:FUNC 'XFR:POW:RAT 2,0'; DET NBAN;*WAI
Reflection	SENS[1 2]:FUNC 'XFR:POW:RAT 1,0'; DET NBAN;*WAI
Power	SENS[1 2]:FUNC 'XFR:POW 2';DET BBAN;*WAI
Conversion Loss	SENS[1 2]:FUNC 'XFR:POW:RAT 2,0'; DET BBAN;*WAI
Detection Options	
Narrowband Internal	
A	SENS[1 2]:FUNC 'XFR:POW 1';DET NBAN;*WAI
B	SENS[1 2]:FUNC 'XFR:POW 2';DET NBAN;*WAI
R	SENS[1 2]:FUNC 'XFR:POW 0';DET NBAN;*WAI
A/R	SENS[1 2]:FUNC 'XFR:POW:RAT 1,0'; DET NBAN;*WAI
B/R	SENS[1 2]:FUNC 'XFR:POW:RAT 2,0'; DET NBAN;*WAI
Broadband Internal	
B*	SENS[1 2]:FUNC 'XFR:POW 2';DET BBAN;*WAI
R*	SENS[1 2]:FUNC 'XFR:POW 0';DET BBAN;*WAI
B*/R*	SENS[1 2]:FUNC 'XFR:POW:RAT 2,0'; DET BBAN;*WAI
Broadband External	
X	SENS[1 2]:FUNC 'XFR:POW 11';DET BBAN;*WAI
Y	SENS[1 2]:FUNC 'XFR:POW 12';DET BBAN;*WAI
X/Y	SENS[1 2]:FUNC 'XFR:POW:RAT 11,12'; DET BBAN;*WAI
Y/X	SENS[1 2]:FUNC 'XFR:POW:RAT 12,11'; DET BBAN;*WAI
Y/R*	SENS[1 2]:FUNC 'XFR:POW:RAT 12,0'; DET BBAN;*WAI

KEYSTROKES	SCPI COMMAND
<b>CHAN 1</b> <b>CHAN 2</b> continued <b>Detection Options</b> continued <b>AUX Input</b> <b>Chan OFF</b>	SENS[1 2]:FUNC 'XFR:VOLT';*WAI SENS[1 2]:STAT OFF;*WAI
<b>FREQ</b> <b>Start</b> <b>Number</b> <b>Units</b> <b>Stop</b> <b>Number</b> <b>Units</b> <b>Center</b> <b>Number</b> <b>Units</b> <b>Span</b> <b>Number</b> <b>Units</b> <b>CW</b> <b>Number</b> <b>Units</b> <b>Disp Freq Resolution</b> <b>MHz</b> <b>kHz</b> <b>Hz</b>	DISP:ANN:FREQ[1 2]:MODE SSTOP SENS[1 2]:FREQ:STAR <num> [MHZ KHZ HZ];*WAI DISP:ANN:FREQ[1 2]:MODE SSTOP SENS[1 2]:FREQ:STOP <num> [MHZ KHZ HZ];*WAI DISP:ANN:FREQ[1 2]:MODE CSPAN SENS[1 2]:FREQ:CENT <num> [MHZ KHZ HZ];*WAI DISP:ANN:FREQ[1 2]:MODE CSPAN SENS[1 2]:FREQ:SPAN <num> [MHZ KHZ HZ];*WAI DISP:ANN:FREQ[1 2]:MODE CW; :SENS[1 2]:FREQ:SPAN 0;*WAI SENS[1 2]:FREQ:CENT <num> [MHZ KHZ HZ];*WAI DISP:ANN:FREQ:RES MHZ DISP:ANN:FREQ:RES KHZ DISP:ANN:FREQ:RES HZ
<b>POWER</b> <b>Level</b> <b>Number</b> <b>ENTER</b> <b>RF ON off</b>	SOUR[1 2]:POW <num> [DBM];*WAI OUTP <ON OFF>;*WAI

KEYSTROKES	SCPI COMMAND
<b>SWEEP</b>	
Sweep Time <b>Number</b> <b>Units</b>	SENS[1 2]:SWE:TIME <num> [SEC];*WAI
Sweep Time AUTO man	SENS[1 2]:SWE:TIME:AUTO <ON OFF>;*WAI
Alt Sweep on OFF	SENS:COUP <NONE ALL>;*WAI
<b>MENU</b>	
<b>Trigger</b>	
Continuous	ABOR::INIT[1 2]:CONT ON;*WAI
Hold	ABOR::INIT[1 2]:CONT OFF;*WAI
Single	ABOR::INIT[1 2]:CONT OFF;:INIT[1 2];*WAI
<b>Trigger Source</b>	
Internal	TRIG:SOUR IMM;:SENS:SWE:TRIG:SOUR IMM;*WAI
External Sweep	TRIG:SOUR EXT;:SENS:SWE:TRIG:SOUR IMM;*WAI
External Point	TRIG:SOUR EXT;:SENS:SWE:TRIG:SOUR EXT;*WAI
Number of Points <b>Number</b> <b>ENTER</b>	SENS[1 2]:SWE:POIN <num>;*WAI
Ext Ref on OFF	SENS:ROSC:SOUR <EXT INT>;*WAI
<b>Spur Avoid</b>	
Spur Avoid on OFF	DIAG:SPUR:AVO <ON OFF>;*WAI
Dither on OFF	DIAG:DITH <ON OFF>;*WAI

KEYSTROKES	SCPI COMMAND
<b>SCALE</b>	
<b>Autoscale</b>	DISP:WIND[1 2]:TRAC:Y:AUTO ONCE
<b>Scale/Div</b> <b>Number</b> <b>ENTER</b>	DISP:WIND[1 2]:TRAC:Y:PDIV <num> [DB]
<b>Reference Level</b> <b>Number</b> <b>ENTER</b>	DISP:WIND[1 2]:TRAC:Y:RLEV <num> [DB]
<b>Reference Position</b> <b>Number</b> <b>ENTER</b>	DISP:WIND[1 2]:TRAC:Y:RPOS <num>
<b>MARKER</b>	
<b>1:</b> or <b>1&gt;</b> <b>Number</b> <b>Units</b>	CALC[1 2]:MARK1 ON CALC[1 2]:MARK1:X <num> [MHZ KHZ HZ]
<b>2:</b> or <b>2&gt;</b> <b>Number</b> <b>Units</b>	CALC[1 2]:MARK2 ON CALC[1 2]:MARK2:X <num> [MHZ KHZ HZ]
<b>3:</b> or <b>3&gt;</b> <b>Number</b> <b>Units</b>	CALC[1 2]:MARK3 ON CALC[1 2]:MARK3:X <num> [MHZ KHZ HZ]
<b>4:</b> or <b>4&gt;</b> <b>Number</b> <b>Units</b>	CALC[1 2]:MARK4 ON CALC[1 2]:MARK4:X <num> [MHZ KHZ HZ]
<b>More Markers</b>	
<b>5:</b> or <b>5&gt;</b> <b>Number</b> <b>Units</b>	CALC[1 2]:MARK5 ON CALC[1 2]:MARK5:X <num> [MHZ KHZ HZ]
<b>6:</b> or <b>6&gt;</b> <b>Number</b> <b>Units</b>	CALC[1 2]:MARK6 ON CALC[1 2]:MARK6:X <num> [MHZ KHZ HZ]
<b>7:</b> or <b>7&gt;</b> <b>Number</b> <b>Units</b>	CALC[1 2]:MARK7 ON CALC[1 2]:MARK7:X <num> [MHZ KHZ HZ]
<b>8:</b> or <b>8&gt;</b> <b>Number</b> <b>Units</b>	CALC[1 2]:MARK8 ON CALC[1 2]:MARK8:X <num> [MHZ KHZ HZ]
<b>Active Marker Off</b>	CALC[1 2]:MARK[1 2]...8 OFF

KEYSTROKES	SCPI COMMAND
<b>MARKER</b> continued	
All Off	CALC[1 2]:MARK:AOFF
Marker Functions	
Delta Mkr on OFF	CALC[1 2]:MARK:MODE <REL ABS>
MARKER → Center	SENS[1 2]:FREQ:CENT (CALC[1 2]:MARK[1 2] ... 8]:X?);*WAI
MARKER → Reference	DISP:WIND[1 2]:TRAC:4:RLEV (CALC[1 2]:MARK[1 2] ... 8]:Y?);*WAI
Marker Search	
Marker > Max	CALC[1 2]:MARK:FUNC MAX
Marker > Min	CALC[1 2]:MARK:FUNC MIN
Target Search	CALC[1 2]:MARK:FUNC TARG
Target Value <b>Number</b> <b>ENTER</b>	CALC[1 2]:MARK:TARG <LEFT RIGH>,<num> [DB]
Search Left <b>Number</b> <b>ENTER</b>	CALC[1 2]:MARK:TARG LEFT,<num> [DB]
Search Right <b>Number</b> <b>ENTER</b>	CALC[1 2]:MARK:TARG RIGH,<num> [DB]
Bandwidth	CALC[1 2]:MARK:FUNC BWID
<b>Number</b> <b>ENTER</b>	CALC[1 2]:MARK:BVID <num> [DB]
Search Off	CALC[1 2]:MARK:FUNC OFF
Tracking on OFF	CALC[1 2]:MARK:FUNC:TRAC <ON OFF>
DISPLAY	
Normalize	TRAC CH[1 2]SMEM,CH[1 2]SDATA; :CALC[1 2]:MATH (IMPL/CH[1 2]SMEM); :DISP:WIND[1 2]:TRAC1 ON;TRAC2 OFF
Data->Mem	TRAC CH[1 2]SMEM,CH[1 2]SDATA
Data	CALC[1 2]:MATH (IMPL); :DISP:WIND[1 2]:TRAC1 ON;TRAC2 OFF
Memory	DISP:WIND[1 2]:TRAC1 OFF;TRAC2 ON

KEYSTROKES	SCPI COMMAND
<b>DISPLAY</b> continued	
<b>Data/Mem</b>	CALC[1 2]:MATH (IMPL/CH[1 2]SMEM); :DISP:WIND[1 2]:TRAC1 ON;TRAC2 OFF
<b>Data and Memory</b>	CALC[1 2]:MATH (IMPL); :DISP:WIND[1 2]:TRAC1 ON;TRAC2 ON
<b>Limit Line</b>	CALC[1 2]:LIM:DISP ON
<b>Add Limit</b>	
<b>Add Max Line</b>	CALC[1 2]:LIM:SEGM[1 2 ...12]:TYPE LMAX; STAT ON
<b>Add Min Line</b>	CALC[1 2]:LIM:SEGM[1 2 ...12]:TYPE LMIN; STAT ON
<b>Add Max Point</b>	CALC[1 2]:LIM:SEGM[1 2 ...12]:TYPE PMAX; STAT ON
<b>Add Min Point</b>	CALC[1 2]:LIM:SEGM[1 2 ...12]:TYPE PMIN; STAT ON
<b>Delete Limit</b>	CALC[1 2]:LIM:SEGM[1 2 ...12]:STAT OFF
<b>Delete All Limits</b>	CALC[1 2]:LIM:SEGM:AOFF
<b>Edit Limit</b>	
<b>Begin Frequency</b> <b>Number</b> <b>ENTER</b>	CALC[1 2]:LIM:SEGM[1 2 ...12]:FREQ:STAR <num> [MHZ KHZ HZ]
<b>End Frequency</b> <b>Number</b> <b>ENTER</b>	CALC[1 2]:LIM:SEGM[1 2 ...12]:FREQ:STOP <num> [MHZ KHZ HZ]
<b>Begin Limit</b> <b>Number</b> <b>ENTER</b>	CALC[1 2]:LIM:SEGM[1 2 ...12]:AMPL:STAR <num> [DB]
<b>End Limit</b> <b>Number</b> <b>ENTER</b>	CALC[1 2]:LIM:SEGM[1 2 ...12]:AMPL:STOP <num> [DB]
<b>Marker</b>	
<b>Limit Line ON off</b>	CALC[1 2]:LIM:DISP <ON OFF>
<b>Limit Test on OFF</b>	CALC[1 2]:LIM:STAT <ON OFF>
<b>More Display</b>	
<b>Split Display</b>	
<b>Full</b>	DISP:FORM SING
<b>Split</b>	DISP:FORM ULOW

KEYSTROKES	SCPI COMMAND
<p><b>DISPLAY continued</b></p> <p><b>More Display continued</b></p> <p><b>Title and Clock</b></p> <p>Enter Line 1</p> <p>Enter Line 2</p> <p>Show Clock on Line 1</p> <p>Show Clock on Line 2</p> <p>Clock Off</p> <p>Title + Clk ON off</p> <p>Y-Axis Lbl ON off</p> <p>Y-Axis Lbl rel ABS</p> <p>Graticule ON off</p>	DISP:ANN:TITL1:DATA <string> DISP:ANN:TITL2:DATA <string> DISP:ANN:CLOC:MODE LINE1 DISP:ANN:CLOC:MODE LINE2 DISP:ANN:CLOC:MODE OFF DISP:ANN:TITL <ON OFF> DISP:ANN:YAX:<ON OFF> DISP:ANN:YAX:MODE <REL ABS> DISP:WIND[1 2]:TRAC:GRAT:GRID <ON OFF>
<p><b>FORMAT</b></p> <p>Log Mag</p> <p>Lin Mag</p> <p>SWR</p> <p>75-ohm Formats</p> <p>Mag dBuV</p> <p>Mag dBmV</p> <p>Mag dBV</p>	CALC[1 2]:FORM MLOG CALC[1 2]:FORM MLIN CALC[1 2]:FORM SWR CALC[1 2]:FORM DBUV CALC[1 2]:FORM DBMV CALC[1 2]:FORM DBV

KEYSTROKES	SCPI COMMAND
<b>CAL</b>	
<b>Normalize</b>	TRAC CH[1 2]SMEM,CH[1 2]SDATA; :CALC[1 2]:MATH (IMPL/CH[1 2]SMEM); :DISP:WIND[1 2]:TRAC1 ON;TRAC2 OFF
<b>Transmissn</b>	
<b>Restore Defaults</b>	SENS[1 2]:CORR:CSET DEF;*WAI
<b>Full Band</b>	SENS[1 2]:CORR:COLL:IST ON;METH TRAN1;*WAI
<b>Measure Standard</b>	SENS[1 2]:CORR:COLL STAN1;*WAI; :SENS[1 2]:CORR:COLL:SAVE;*WAI
<b>User Defined</b>	SENS[1 2]:CORR:COLL:IST OFF;METH TRAN1; *WAI
<b>Measure Standard</b>	SENS[1 2]:CORR:COLL STAN1;*WAI; :SENS[1 2]:CORR:COLL:SAVE;*WAI
<b>Reflection</b>	
<b>Restore Defaults</b>	SENS[1 2]:CORR:CSET DEF;*WAI
<b>Full Band</b>	SENS[1 2]:CORR:COLL:IST ON;METH REFL3;*WAI
<b>Measure Standard</b> — Open	SENS[1 2]:CORR:COLL STAN1;*WAI
<b>Measure Standard</b> — Short	SENS[1 2]:CORR:COLL STAN2;*WAI
<b>Measure Standard</b> — Load	SENS[1 2]:CORR:COLL STAN3;*WAI; :SENS[1 2]:CORR:COLL:SAVE;*WAI
<b>User Defined</b>	SENS[1 2]:CORR:COLL:IST OFF;METH REFL3; *WAI
<b>Measure Standard</b> — Open	SENS[1 2]:CORR:COLL STAN1;*WAI
<b>Measure Standard</b> — Short	SENS[1 2]:CORR:COLL STAN2;*WAI
<b>Measure Standard</b> — Load	SENS[1 2]:CORR:COLL STAN3;*WAI; :SENS[1 2]:CORR:COLL:SAVE;*WAI
<b>Cal Kit</b>	
<b>Default Type-N(f)</b>	SENS:CORR:COLL:CKIT 'COAX,7MM,TYPE-N,50,FEMALE' SENS:CORR:COLL:CKIT 'COAX,7MM,TYPE-N,75,FEMALE' (option 1EC)
<b>Type-N(m)</b>	SENS:CORR:COLL:CKIT 'COAX,7MM,TYPE-N,50,MALE' SENS:CORR:COLL:CKIT 'COAX,7MM,TYPE-N,75,MALE' (option 1EC)

KEYSTROKES	SCPI COMMAND
<b>CAL</b> continued	
<b>Cal Kit</b> continued	
<b>User Defined</b>	SENS:CORR:COLL:CKIT 'USER, IMPLIED, IMPLIED, IMPLIED, IMPLIED'
<b>3.5 mm</b>	SENS:CORR:COLL:CKIT 'COAX,3.5MM,APC-3.5,50,IMPLIED'
<b>Detector Zero</b>	
<b>Autozero</b>	CAL:ZERO:AUTO ON;*WAI
<b>Manual Zero</b>	CAL:ZERO:AUTO ONCE;*WAI
<b>AVG</b>	
<b>Average on OFF</b>	SENS[1 2]:AVER <ON OFF>;*WAI
<b>Restart Average</b>	SENS[1 2]:AVER:CLE;*WAI
<b>Average Factor</b> <b>Number</b> <b>ENTER</b>	SENS[1 2]:AVER:COUN <num>;*WAI
<b>System Bandwidth</b>	
<b>Wide</b>	SENS[1 2]:BWID 6500 HZ;*WAI
<b>Medium</b>	SENS[1 2]:BWID 750 HZ;*WAI
<b>Narrow</b>	SENS[1 2]:BWID 250 HZ;*WAI
<b>SAVE RECALL</b>	
<b>Save State</b>	MMEM:STOR:STAT 1,<file> <sup>1</sup>
<b>Re-Save State</b>	MMEM:STOR:STAT 1,<file> <sup>1</sup>

<sup>1</sup> <file> may include the mass storage device mnemonic **MEM:**, **INT:** or **EXT:** before the actual name of the file. If the mass storage device is not explicitly named the currently selected device is assumed. <file>, <file1> and <file2> are <string> parameters.

KEYSTROKES	SCPI COMMAND
<b>SAVE RECALL</b> continued	
<b>Define Save</b>	
Inst State ON off	MMEM:STOR:STAT:IST <ON OFF>
Cal on OFF	MMEM:STOR:STAT:CORR <ON OFF>
Data on OFF	MMEM:STOR:STAT:TRAC <ON OFF>
Save ASCII	
Save Chan 1	MMEM:STOR:TRAC CH1FDATA,<file> <sup>1</sup>
Save Chan 2	MMEM:STOR:TRAC CH2FDATA,<file> <sup>1</sup>
<b>Recall State</b>	MMEM:LOAD:STAT 1,<file> <sup>1</sup>
<b>Programs</b>	
Save Program	
Re-Save Program	
Recall Program	
Save AUTOST	
IBASIC <sup>2</sup>	
<b>Select Disk</b>	
Internal Memory	MMEM:MSIS 'MEM:'
Internal Disk	MMEM:MSIS 'INT:'
External Disk	MMEM:MSIS 'EXT:' <sup>3</sup>
<b>Configure Ext. Disk</b>	
Ext Disk Address Number ENTER	SYST:COMM:GPIB:MMEM:ADDR <num>
Ext Disk Unit Number ENTER	SYST:COMM:GPIB:MMEM:UNIT <num>
Ext Disk Volume Number ENTER	SYST:COMM:GPIB:MMEM:VOL <num>

2 The IBASIC menu is described under the **SYSTEM OPTIONS** key.

3 Active controller status must be passed to the instrument (from IBASIC or the external controller) for external disk access.

KEYSTROKES	SCPI COMMAND
<b>SAVE RECALL</b> continued	
<b>File Utilities</b>	
<b>Rename File</b>	MMEM:MOVE <file1>,<file2> <sup>1</sup>
<b>Delete File</b>	MMEM:DEL <file> <sup>1</sup>
<b>Delete All Files</b>	MMEM:DEL '*.*' <sup>1,4</sup>
<b>Copy File</b>	
<b>Copy to Int Memory</b>	MMEM:COPY <file1>,<'MEM:file2'> <sup>1</sup>
<b>Copy to Int Disk</b>	MMEM:COPY <file1>,<'INT:file2'> <sup>1</sup>
<b>Copy to Ext Disk</b>	MMEM:COPY <file1>,<'EXT:file2'> <sup>1,3</sup>
<b>Copy All Files</b>	
<b>Copy to Int Memory</b>	MMEM:COPY '*.*','MEM:' <sup>1,4</sup>
<b>Copy to Int Disk</b>	MMEM:COPY '*.*','INT:' <sup>1,4</sup>
<b>Copy to Ext Disk</b>	MMEM:COPY '*.*','EXT:' <sup>1,3,4</sup>
<b>Format Disk</b>	
<b>Format Int Memory</b>	MMEM:INIT 'MEM:',<DOS LIF> <sup>5</sup>
<b>Format Int Disk</b>	MMEM:INIT 'INT:',<DOS LIF> <sup>5</sup>
<b>Format Ext Disk</b>	MMEM:INIT 'EXT:',<DOS LIF> <sup>3,5</sup>
<b>Directory Utilities</b>	
<b>Change Directory</b>	MMEM:CDIR <directory> <sup>6</sup>
<b>Make Directory</b>	MMEM:MDIR <directory> <sup>6</sup>
<b>Remove Directory</b>	MMEM:RDIR <directory> <sup>6</sup>
<b>HARD COPY</b>	
<b>Start</b>	HCOP:*WAI
<b>Abort</b>	HCOP:ABOR

4 '\*.\*' is the form for "all files" with a DOS formatted disk — a LIF formatted disk uses '\*' with no extension.

5 When a disk is formatted using the front panel keys the DOS format is always used. The LIF format is available when the mnemonic is used.

6 For use with DOS formatted disks only — the HP 8711 does not support LIF disks that use HFS (hierarchical file structure). <directory> is a <string> parameter.

KEYSTROKES	SCPI COMMAND
<b>HARD COPY</b> continued	
<b>Select Copy Port</b>	
<b>Restore Defaults</b>	
<b>Select</b>	HCOP:DEV:LANG <PCL HPGL IBM>; PORT <CENT SER GPIB MMEM>
<b>Hardcopy Address</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/>	SYST:COMM:GPIB:HCOP:ADDR <num>
<b>Baud Rate</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/>	SYST:COMM:SER:TRAN:BAUD <num>
<b>Xon/Xoff</b>	SYST:COMM:SER:TRAN:HAND XON
<b>DTR/DSR</b>	SYST:COMM:SER:TRAN:HAND DTR
<b>Define Printer</b>	
<b>Restore Defaults</b>	
<b>Monochrome</b>	HCOP:DEV1:COL OFF
<b>Color</b>	HCOP:DEV1:COL ON
<b>Portrait</b>	HCOP:PAGE:ORI PORT
<b>Landscape</b>	HCOP:PAGE:ORI LAND
<b>Auto Feed</b> ON off	HCOP:ITEM1:FFE:STAT <ON OFF>
<b>More Printer</b>	
<b>Restore Defaults</b>	
<b>Printer Resolution</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/>	HCOP:DEV:RES <num>
<b>Top Margin</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/>	HCOP:PAGE:MARG:TOP <num>
<b>Left Margin</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/>	HCOP:PAGE:MARG:LEFT <num>
<b>Print Width</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/>	HCOP:PAGE:WIDT <num>
<b>Define Plotter</b>	
<b>Restore Defaults</b>	
<b>Monochrome</b>	HCOP:DEV2:COL OFF
<b>Color</b>	HCOP:DEV2:COL ON

KEYSTROKES	SCPI COMMAND
<b>HARD COPY</b> continued	
<b>Define Plotter</b> continued	
<b>Set Pen Numbers</b>	
<b>Monochrome Pen</b> <b>Number</b> <b>ENTER</b>	
<b>Default Pen Colors</b>	
<b>Trace 1</b> <b>Number</b> <b>ENTER</b>	
<b>Trace 2</b> <b>Number</b> <b>ENTER</b>	
<b>Memory 1</b> <b>Number</b> <b>ENTER</b>	
<b>Memory 2</b> <b>Number</b> <b>ENTER</b>	
<b>Graticule</b> <b>Number</b> <b>ENTER</b>	
<b>Graphics</b> <b>Number</b> <b>ENTER</b>	
<b>Auto Feed on OFF</b>	<b>HCOP:ITEM2:FFE:STAT &lt;ON OFF&gt;</b>
<b>Define Hardcopy</b>	
<b>Restore Defaults</b>	
<b>Graph and Mkr Table</b>	<b>HCOP:DEV:MODE GMAR</b>
<b>Graph Only</b>	<b>HCOP:DEV:MODE GRAP</b>
<b>Mkr Table Only</b>	<b>HCOP:DEV:MODE MARK</b>
<b>List Trace Values</b>	<b>HCOP:DEV:MODE TABL</b>
<b>Define Graph</b>	
<b>Restore Defaults</b>	
<b>Trace Data ON off</b>	<b>HCOP:ITEM:TRAC:STAT &lt;ON OFF&gt;</b>
<b>Graticule ON off</b>	<b>HCOP:ITEM:GRAT:STAT &lt;ON OFF&gt;</b>
<b>Annotation ON off</b>	<b>HCOP:ITEM:ANN:STAT &lt;ON OFF&gt;</b>
<b>Mkr Symbol ON off</b>	<b>HCOP:ITEM:MARK:STAT &lt;ON OFF&gt;</b>
<b>Title + Clk ON off</b>	<b>HCOP:ITEM:TITL:STAT &lt;ON OFF&gt;</b>

KEYSTROKES	SCPI COMMAND
<b>SYSTEM OPTIONS</b>	
<b>IBASIC</b>	
Run	PROG:STAT RUN
Continue	PROG:STAT CONT
Step	PROG:EXEC 'STEP'
Edit	
Key Record on OFF	
Clear Program	PROG:DEL
<b>IBASIC Display</b>	
None	DISP:PROG OFF
Full	DISP:PROG FULL
Upper	DISP:PROG UPP
Lower	DISP:PROG LOW
<b>HP-IB</b>	
HP8711 Address Number ENTER	SYST:COMM:GPIB:ADDR <num> <sup>7</sup>
Talker Listener	SYST:COMM:GPIB:CONT OFF <sup>8</sup>
System Controller	SYST:COMM:GPIB:CONT ON <sup>8</sup>
HP-IB Echo on OFF	SYST:COMM:GPIB:ECHO <ON OFF>
<b>Operating Parameters</b>	
Hardcopy All	HCOP:DEV:MODE ISET;:HCOP;*WAI
Abort	HCOP:ABOR

7 A 5 second delay is required before a command is sent to the new address.

8 For use with IBASIC running on the HP 8711's internal controller — this command cannot be executed from an external controller. Use \*OPC? and wait for a reply before sending any OUTPUT 7xx commands from IBASIC.

9 <year>,<month>,<day>,<hour>,<minute> and <second> are all <num> parameters.

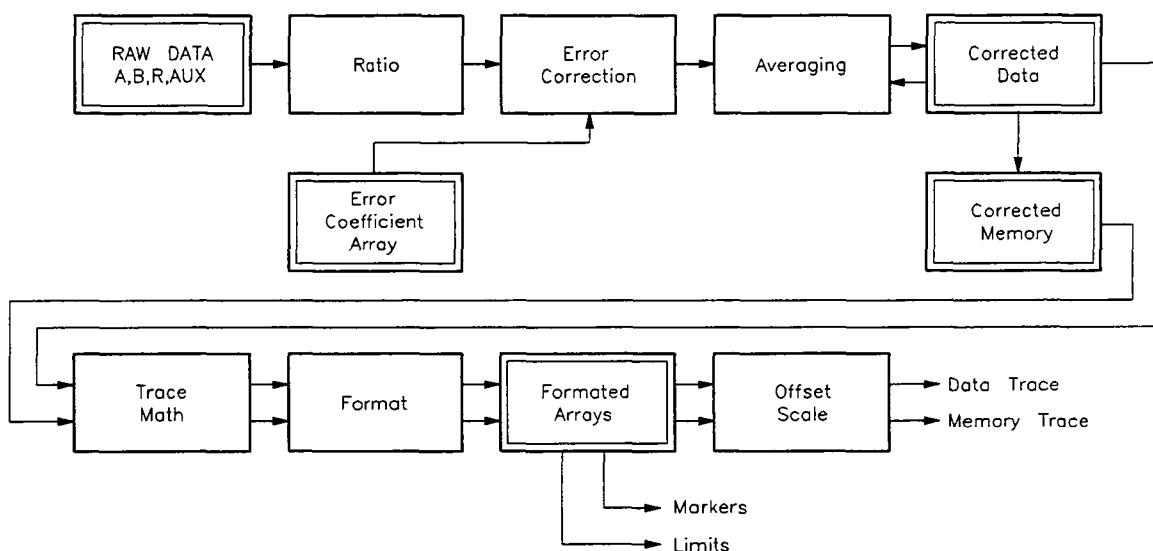
KEYSTROKES	SCPI COMMAND
<p><b>SYSTEM OPTIONS</b> continued</p> <p><b>System Config</b></p> <p><b>Set Clock</b></p> <p><b>Set Year</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/></p> <p><b>Set Month</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/></p> <p><b>Set Day</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/></p> <p><b>Set Hour</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/></p> <p><b>Set Minute</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/></p> <p><b>Round Seconds</b></p> <p><b>Clock Format</b></p> <p><b>YYYY-MM-DD HH:MM</b></p> <p><b>MM-DD-YYYY HH:MM</b></p> <p><b>DD-MM-YYYY HH:MM</b></p> <p><b>Numeric</b></p> <p><b>Alpha</b></p> <p><b>Seconds ON off</b></p> <p><b>Done</b></p> <p><b>Beeper Volume</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/></p> <p><b>External CRT Adjust</b></p> <p><b>Restore Defaults</b></p> <p><b>Vertical Back Porch</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/></p> <p><b>Vertical Frnt Porch</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/></p> <p><b>Horizontal Back Porch</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/></p> <p><b>Horizontal Frnt Porch</b> <input type="button" value="Number"/> <input type="button" value="ENTER"/></p> <p><b>Service</b> <sup>10</sup></p>	<p>SYST:DATE &lt;year&gt;,&lt;month&gt;,&lt;day&gt;<sup>9</sup></p> <p>SYST:DATE &lt;year&gt;,&lt;month&gt;,&lt;day&gt;<sup>9</sup></p> <p>SYST:DATE &lt;year&gt;,&lt;month&gt;,&lt;day&gt;<sup>9</sup></p> <p>SYST:TIME &lt;hour&gt;,&lt;minute&gt;,&lt;second&gt;<sup>9</sup></p> <p>SYST:TIME &lt;hour&gt;,&lt;minute&gt;,&lt;second&gt;<sup>9</sup></p> <p>SYST:TIME &lt;hour&gt;,&lt;minute&gt;,0<sup>9</sup></p> <p>DISP:ANN:CLOC:DATE:FORM YMD</p> <p>DISP:ANN:CLOC:DATE:FORM MDY</p> <p>DISP:ANN:CLOC:DATE:FORM DMY</p> <p>DISP:ANN:CLOC:DATE:MODE NUM</p> <p>DISP:ANN:CLOC:DATE:MODE ALPH</p> <p>DISP:ANN:CLOC:SEC &lt;ON OFF&gt;</p> <p>SYST:BEEP:VOL &lt;num&gt;</p>

10 The **Service** menu is described in the *HP 8711 Service Manual* (HP part no. 08711-90003).

## Internal Measurement Arrays

The following paragraphs describe the sequence of math operations and the resulting data arrays as the measurement information flows from the raw data arrays to the display. This information explains the measurement arrays accessible via HP-IB.

Figure 9b-4 is a data processing flow diagram that represents the flow of numerical data. The data passes through several math operations, denoted in the figure by single-line boxes. Most of these operations can be selected and controlled with the front panel CONFIGURE block menus. The data is also stored in arrays along the way, denoted by double-line boxes. These arrays are places in the flow path where data is accessible via HP-IB. While only a single flow path is shown, two identical paths are available, corresponding to channel 1 and channel 2.



**Figure 9b-4. Numeric Data Flow Through the Network Analyzer**

## Raw Data Arrays

These arrays are linear measurements of the inputs used in the selected measurement. Note that these numbers are complex pairs. These arrays are directly accessible via HP-IB and referenced as CH<1|2>AFWD, CH<1|2>BFWD and CH<1|2>RFWD.

**Table 9b-1. Raw Data Arrays**

Selected Measurement	Raw Arrays
Transmission (B/R)	B = CH<1 2>BFWD, R = CH<1 2>RFWD
Reflection (A/R)	A = CH<1 2>AFWD, R = CH<1 2>RFWD
A	A = CH<1 2>AFWD
B	B = CH<1 2>BFWD
R	R = CH<1 2>RFWD
Power (B*)	B* = CH<1 2>BFWD
Conversion Loss (B*/R*)	B* = CH<1 2>BFWD, R* = CH<1 2>RFWD
R*	R* = CH<1 2>RFWD
X	X = CH<1 2>RFWD
Y	Y = CH<1 2>BFWD
Y/R*	Y = CH<1 2>BFWD, R* = CH<1 2>RFWD
Y/X, X/Y	Y = CH<1 2>BFWD, X = CH<1 2>RFWD

**Note**

Raw data for AUX INPUT is not available via HP-IB. Use the corrected data array to access AUX INPUT data.

**Ratio Calculations**

These are performed if the selected measurement is a ratio (e.g. A/R or B/R). This is simply a complex divide operation. If the selected measurement is absolute (e.g. A or B), no operation is performed.

**Error Correction**

Error correction is performed next if correction is turned on. Error correction removes repeatable systematic errors (stored in the error coefficient arrays) from the raw arrays. The operations performed depend on the selected measurement type.

**Error Coefficient Arrays**

The error coefficient arrays are either default values or are created during a measurement calibration. These are used whenever correction is on, contain complex number pairs, and are accessible via HP-IB and are referenced as CH<1|2>SCORR1, CH<1|2>SCORR2 and CH<1|2>SCORR3.

**Table 9b-2. Error Coefficient Arrays**

Selected Measurement	Error Coefficient Arrays
Transmission (B/R),A,B,R	CH<1 2>SCORR1 = Response
Reflection (A/R)	CH<1 2>SCORR1 = Directivity CH<1 2>SCORR2 = Source Match CH<1 2>SCORR3 = Tracking
Broadband Internal	CH<1 2>SCORR1 = R* Response

---

**Note**

These arrays do not apply to Broadband External measurements.



---

## Averaging

This is a noise reduction technique. This calculation involves taking the complex exponential average of several consecutive sweeps.

## Corrected Data Arrays

The combined results of the ratio, error correction and averaging operations are stored in the corrected data arrays as complex number pairs. These arrays are accessible via HP-IB and referenced as CH<1|2>SDATA.

## Corrected Memory Arrays

If the Data->Mem or Normalize operations are performed, the corrected data arrays are copied into the corrected memory arrays.

## Trace Math Operation

This selects either the corrected data array, or the corrected memory array, or both to continue flowing through the data processing path. In addition, the complex ratio of the two (Data/Memory) can also be selected. If memory is displayed, the data from the memory arrays goes through exactly the same data processing flow path as the data from the data arrays.

## Formatting

This converts the complex number pairs into a scalar representation for display, according to the selected format (e.g. Log Mag, SWR, etc). These formats are often easier to interpret than the complex number representation. Note that after formatting, it is impossible to recover the complex data.

## Formatted Arrays

The results so far are stored in the formatted data and formatted memory arrays. It is important to note that marker values and marker functions are all derived from the formatted arrays. Limit testing is also performed on the formatted arrays. These arrays are accessible to HP-IB and referenced as CH<1|2>FDATA and CH<1|2>FMEM.

## Offset and Scale. These operations prepare the formatted arrays for display.

This is where the reference position, reference value, and scale calculations are performed, as appropriate for the format.

## SCPI Command Summary

This section contains all of the HP-IB commands recognized by the HP 8711A and a brief description. <num>, <char>, <string> and <block> refer to the parameter type expected by the instrument as part of the command. All commands have both command and query forms unless specified as command only or query only. Unless otherwise specified, add a "?" to create a query from the command form. For example, the command to select the log magnitude format for the data displayed is CALCulate[1|2]:FORMAT ML0Garithmic. To query which format is active the corresponding command is CALCulate[1|2]:FORMAT?. The response to the query is the short form of the mnemonic for the active format, in this example MLOG.

The FORM column gives the parameter type returned by the instrument in response to a query. NR1, NR2 and NR3 refer to the different types of numeric data. CHAR (character data), STRING (string data) and BLOCK (block data) are also used to describe response types. These parameter types are described in the "Parameter Types" section.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
<b>ABORt</b>		
ABORt	command only	Abort and reset the sweep in progress.
<b>CALCulate</b>		
CALCulate[1 2]:DATA?	query only BLOCK or NR3 <sup>1</sup>	Query the formatted data trace — functionally equivalent to the command TRAC? CH<1 2>FDATA.
CALCulate[1 2]:FORMAT <char>	CHAR	Select the display format for the measurement data — choose from MLOGarithmic MLINear SWR DBV DBMV DBUV.
CALCulate[1 2]:LIMit:DISPlay <ON OFF> <sup>2</sup>	NR1	Turn on/off display of limit lines.
CALCulate[1 2]:LIMit:SEGment[1 2...12]:AMPLitude:STARt <num> <sup>3</sup>	NR3	Set the Begin Limit for the specified limit segment.
CALCulate[1 2]:LIMit:SEGment[1 2...12]:AMPLitude:STOP <num> <sup>3</sup>	NR3	Set the End Limit for the specified limit segment.

1 The parameter type of the data is determined by the format selected — FORMAT REAL uses BLOCK data, FORMAT ASCII uses NR3 data separated by commas.

2 Binary parameters accept the values of 1 (on) and 0 (off) in addition to ON and OFF.

3 Numeric parameters may include an appropriate terminator MHZ|HZ|EZ|DB|DBM|SEC|MS — if no terminator is included the default (HZ for frequency or SEC for time) is assumed.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
CALCulate[1 2]:LIMIT:SEGMENT:AOFF	command only	Turn off all limit segments for a given channel — deletes all segments in the channel's limit table.
CALCulate[1 2]:LIMIT:SEGMENT[1 2 ...12]:FREQUENCY:STARt <num> <sup>3</sup>	NR3	Set the Begin Frequency for the specified limit segment.
CALCulate[1 2]:LIMIT:SEGMENT[1 2 ...12]:FREQUENCY:STOP <num> <sup>3</sup>	NR3	Set the End Frequency for the specified limit segment.
CALCulate[1 2]:LIMIT:SEGMENT[1 2 ...12]:STATE <ON OFF> <sup>2</sup>	NR1	Turn on/off the specified limit segment — adds or deletes the segment.
CALCulate[1 2]:LIMIT:SEGMENT[1 2 ...12]:TYPE <char>	CHAR	Set the limit type for the specified segment, choose from LMAX LMIN PMAX PMIN (Max Line, Min Line, Max Point, Min Point) — sets all of the segment's limit parameters to their default values.
CALCulate[1 2]:LIMIT:STATE <ON OFF> <sup>2</sup>	NR1	Turn on/off the limit test.
CALCulate[1 2]:MARKer:AOFF	command only	Turn off all markers for a given channel — this has the effect of turning off marker functions and tracking as well.
CALCulate[1 2]:MARKer:BWIDth <num> <sup>3</sup>	NR3	Calculate the bandwidth of a bandpass filter — num is the target bandwidth (-3 for the 3 dB bandwidth).
CALCulate[1 2]:MARKer:FUNCTION :RESult?	query only NR3[,NR3, NR3, NR3]	Query the results of the active marker function — MAX and MIN return the amplitude; TARG returns the frequency; BWID returns bandwidth, center frequency, Q and loss.
CALCulate[1 2]:MARKer:FUNCTION [:SElect] <char>	CHAR	Select the active marker function — choose from OFF MAXimum MINimum TARGet BWIDth.
CALCulate[1 2]:MARKer:FUNCTION :TRACKing <ON OFF> <sup>2</sup>	NR1	Turn on/off marker function tracking.
CALCulate[1 2]:MARKer[1 2 ...8]:MAXimum	command only	Set the specified marker to the maximum value on the trace.
CALCulate[1 2]:MARKer[1 2 ...8]:MINimum	command only	Set the specified marker to the minimum value on the trace.
CALCulate[1 2]:MARKer:MODE <char>	CHAR	Turn on/off delta marker state — choose ABSolute or RELative.
CALCulate[1 2]:MARKer[1 2 ...8]:POINT	NR3	Set the specified marker point.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
CALCulate[1 2]:MARKer:REFerence:X?	query only NR3	Query the frequency of the reference marker.
CALCulate[1 2]:MARKer:REFerence:Y?	query only NR3	Query the amplitude of the reference marker.
CALCulate[1 2]:MARKer[1 2 ...8] [:STATe] <ON OFF>	NR1	Turn on/off the specified marker.
CALCulate[1 2]:MARKer[1 2 ...8] :TARGet <char>,<num> <sup>3</sup>	CHAR, NR3	Perform a marker search for a target value — <b>char</b> is the direction LEFT or RIGHT.
CALCulate[1 2]:MARKer[1 2 ...8]:X	NR3	Set the specified marker frequency.
CALCulate[1 2]:MARKer[1 2 ...8]:Y?	query only NR3	Query the specified marker amplitude.
CALCulate[1 2]:MATH[:EXPRession] <expr> <sup>4</sup>	EXPR <sup>4</sup>	Select a trace math expression — choose measurement trace from (IMPL) for “data only” or (IMPL/CH<1 2>SMEM) for “data / memory”.
<b>CALibration</b>		
CALibration:ZERO:AUTO <ON OFF ONCE> <sup>2</sup>	NR1	Turn on/off the broadband detector autozeroing function.
<b>CONFigure</b>		
CONFigure <string>	STRING	Configure the analyzer to measure a specific device type and parameter (the <b>BEGIN</b> function) — choose from one of the following strings: 'AMPLifier:TRANsmission', 'AMPLifier:REFlection', 'AMPLifier:POWER', 'FILTER:TRANsmission', 'FILTER:REFlection', 'BBAND:TRANsmission', 'BBAND:REFlection', 'MIXer:CLOSS', or 'MIXer:REFlection'.

<sup>4</sup> <expr> and EXPR represent expressions, a parameter type that consists of mathematical expressions that use character parameters and are enclosed in parentheses.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
<b>DIAGnostic</b>		
DIAGnostic:CCONstants:LOAD	command only	Load default factory calibration constants from floppy disk to memory.
DIAGnostic:CCONstants:STORE:DISK	command only	Store default factory calibration constants from memory to floppy disk.
DIAGnostic:CCONstants:STORE:EEPROM	command only	Store default factory calibration constants from memory to flash eeprom.
DIAGnostic:CPU:MEMORY[:WORD] <num1><num2>	NR1	Write a 16 bit num2 value to a 68000 address num1 — query returns an integer with the 16 bit value at num1.
DIAGnostic:DITHER <ON OFF> <sup>2</sup>	NR1	Turn on/off the dithering function.
DIAGnostic:SNUMber <string>?	query only STRING	Query the instrument's serial number.
DIAGnostic:SPUR:AVOID <ON OFF> <sup>2</sup>	NR1	Turn on/off the spur avoidance mode.
<b>DISPlay</b>		
DISPLAY:ANNotation:CLOCK:DATE:FORMAT <char>	CHAR	Select the Year/Month/Day ordering of the date in the clock display — choose from YMD MDY DMY.
DISPLAY:ANNotation:CLOCK:DATE:MODE <char>	CHAR	Select the format for the date in the clock display — choose NUMERIC or ALPHA.
DISPLAY:ANNotation:CLOCK:MODE <char>	CHAR	Select how the clock will appear in the measurement display title area — choose from LINE1 LINE2 OFF.
DISPLAY:ANNotation:CLOCK:SECONDS[:STATE] <ON OFF> <sup>2</sup>	NR1	Turn on/off display of seconds in the clock display.
DISPLAY:ANNotation:FREQUENCY[1 2]:MODE <char>	CHAR	Set the frequency annotation on the display — choose SSTOP (start/stop), CSPAN (center/span) or CW.
DISPLAY:ANNotation:FREQUENCY:RESOLUTION <char>	CHAR	Set the resolution of display frequency values — choose from MHZ KHZ HZ.
DISPLAY:ANNotation:MARKER[1 2]:STATE <ON OFF> <sup>2</sup>	NR1	Enable/disable the active marker annotation for channels 1 and 2.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
DISPlay:ANNotation:MESSAge:AOff	command only	Turns off any currently showing message window — includes message window, active entry and IBASIC window.
DISPlay:ANNotation:MESSAge:STATe <ON OFF> <sup>2</sup>	NR1	Enable/disable the message window — CAUTION: this suppresses display of all messages (even ERROR messages).
DISPlay:ANNotation:TITLe[1 2]:DATA <string>	STRING	Enter a string for the specified title line.
DISPlay:ANNotation:TITLe[:STATe] <ON OFF> <sup>2</sup>	NR1	Turn on/off display of the title and clock.
DISPlay:ANNotation:YAXis:MODE <char>	CHAR	Set mode for the Y-axis labels — choose RELATIVE or ABSolute
DISPlay:ANNotation:YAXis[:STATe] <ON OFF> <sup>2</sup>	NR1	Turn on/off Y-axis labels.
DISPlay:FORMAT <char>	CHAR	Select the format (full or split screen) for displaying trace data — choose SINGLE (overlay) or ULOWer (split).
DISPlay:MENU:KEY[1 2 ...7] <string>	STRING	Set up softkey labels for use in REMOTE operation (softkey 8 is always Return to LOCAL).
DISPlay:PROGraM[:MODE] <char>	CHAR	Select the portion of the analyzer's screen to be used as an HP Instrument BASIC display — choose from OFF FULL UPPer LOWer.
DISPlay:WINDOW[1 2 10]:GEOMetry :LLEFT?	query only NR1,NR1	Query the absolute pixel coordinates of the lower left corner of the selected display window.
DISPlay:WINDOW[1 2 10]:GEOMetry :SIZE?	query only NR1,NR1	Query the width and height (in pixels) of the selected display window.
DISPlay:WINDOW[1 2 10]:GEOMetry :URIGHT?	query only NR1,NR1	Query the absolute pixel coordinates of the upper right corner of the selected display window.
DISPlay:WINDOW:GRAPHics:BUFFer [:STATe] <ON OFF> <sup>2</sup>	NR1	Turn on/off buffering of user graphics commands.
DISPlay:WINDOW[1 2 10]:GRAPHics :CIRCLE <num>	command only	Draw a circle of the specified diameter centered at the current pen location — num is the diameter in pixels.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
DISPlay:WINDOW[1 2 10]:GRAPHics :CLEar	command only	Clear the user graphics and graphics buffer for the specified window.
DISPlay:WINDOW[1 2 10]:GRAPHics :COLor <num>	NR1	Set the color of the user graphics pen — choose from 0 for erase, 1 for bright, and 2 for dim.
DISPlay:WINDOW[1 2 10]:GRAPHics [:DRAW] <num1>,<num2>	command only	Draw a line from the current pen position to the specified new pen position — num1 and num2 are the new absolute X and Y coordinates in pixels.
DISPlay:WINDOW[1 2 10]:GRAPHics :LABel <string>	command only	Draw a label with the lower left corner at the current pen location.
DISPlay:WINDOW[1 2 10]:GRAPHics :LABel:FONT <char>	CHAR	Select the user graphics label font — choose from SMALL HSMall NORMAll HNORMal BOLD HBOLD SLANT HSLant.
DISPlay:WINDOW[1 2 10]:GRAPHics :MOVE <num1>,<num2>	NR1, NR1	Move the pen to the specified new pen position — num1 and num2 are the new absolute X and Y coordinates in pixels.
DISPlay:WINDOW[1 2 10]:GRAPHics :RECTangle <num1>,<num2>	command only	Draw a rectangle of the specified size with lower left corner at the current pen position — num1 and num2 are the width and height in pixels.
DISPlay:WINDOW[1 2 10]:GRAPHics :STATE?	query only NR1	Query whether a window is enabled for user graphics commands.
DISPlay:WINDOW[1 2]:TRACe: GRATICule:GRID[:STATE] <ON OFF> <sup>2</sup>	NR1	Turn on/off display graticule.
DISPlay:WINDOW[1 2]:TRACe[1 2] [:STATE] <ON OFF> <sup>2</sup>	NR1	Turn on/off the display of trace and memory data from the specified channel.
DISPlay:WINDOW[1 2]:TRACe:Y [:SCALE]:AUTO ONCE	command only	Scale the measurement data for a best fit display.
DISPlay:WINDOW[1 2]:TRACe:Y [:SCALE]:PDIVision <num> <sup>3</sup>	NR3	Specify the height (dB or units per division) of each vertical division of the specified channel.
DISPlay:WINDOW[1 2]:TRACe:Y [:SCALE]:RLEVel <num> <sup>3</sup>	NR3	Specify the value for the Y-axis reference position for the specified channel.
DISPlay:WINDOW[1 2]:TRACe:Y [:SCALE]:RPOSITION <num>	NR3	Specify the Y-axis reference position for the specified channel.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
<b>FORMAT</b>		
FORMAT:BORDer <char>	CHAR	Specify the byte order used for HPIB data transfer — choose NORMAL or SWAPPED (for PC-compatible systems).
FORMAT[:DATA] <char>[,<num>]	CHAR[,NR1]	Specify the data format for use during data transfer — choose from REAL,64 REAL,32 INTeger,16  ASCII.
<b>HCOPY</b>		
HCOPY:ABORT	command only	Aborts any hardcopy currently in progress.
HCOPY:DEVICE[1 2]:COLOR <ON OFF> <sup>2</sup>	NR1	Select monochrome OFF or color ON mode for hardcopy output — use device 1 for printers and 2 for plotters.
HCOPY:DEVICE:LANGUAGE <char>	CHAR	Select the language for hardcopy output — choose from PCL HPGL IBM.
HCOPY:DEVICE:MODE <char>	CHAR	Select the graph and/or table(s) to appear on a hardcopy plot — choose from GMARKer GRAPH ISETtings MARKer TABLE.
HCOPY:DEVICE:PORT <char>	CHAR	Select the communications port for hardcopy output — choose from CENTronics SERIAL GPIB MMEMORY.
HCOPY:DEVICE:RESolution <num>	NR1	Set the printer resolution in millimeters.
HCOPY[:IMMEDIATE]	command only	Initiates a hardcopy output (print or plot).
HCOPY:ITEM:ANNotation:STATE <ON OFF> <sup>2</sup>	NR1	Turns on/off channel and frequency annotation as part of hardcopy output.
HCOPY:ITEM[1 2]:FFEed:STATE <ON OFF> <sup>2</sup>	NR1	Turns on/off an automatic form feed at the completion of hardcopy output — use item 1 for printers and 2 for plotters.
HCOPY:ITEM:GRATICule:STATE <ON OFF> <sup>2</sup>	NR1	Turns on/off graticule as part of hardcopy output.
HCOPY:ITEM:MARKer:STATE <ON OFF> <sup>2</sup>	NR1	Turns on/off marker symbols as part of hardcopy output.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
HCOPy:ITEM:TITLe:STATE <ON OFF> <sup>2</sup>	NR1	Turns on/off title and clock lines as part of hardcopy output.
HCOPy:ITEM:TRACe:STATE <ON OFF> <sup>2</sup>	NR1	Turns on/off trace data as part of hardcopy output.
HCOPy:PAGE:MARGin:TOP <num>	NR2	Sets the top margin (for printer output) in millimeters.
HCOPy:PAGE:MARGin:LEFT <num>	NR2	Sets the left margin (for printer output) in millimeters.
HCOPy:PAGE:ORIentation <char>	CHAR	Sets printer output page orientation — choose PORTrait or LANDscape.
HCOPy:PAGE:WIDTh <num>	NR2	Sets the print width (for printer output) in millimeters.
<b>INITiate</b>		
INITiate[1 2]:CONTinuous <ON OFF> <sup>2</sup>	NR1	Set the trigger system to continuously sweep or to stop sweeping.
INITiate[1 2] [:IMMEDIATE]	command only	Initiate a new measurement sweep.
<b>MMEMemory</b>		
MMEMemory:CDIRectory <string>	STRING	Change the current directory on a DOS formatted disk — new directory must be on the same mass storage device.
MMEMemory:COPY <string1>,<string2> <sup>5</sup>	command only	Copy a file — <b>string1</b> is the source file, <b>string2</b> is the destination file.
MMEMemory:DELete <string> <sup>5</sup>	command only	Delete a file — <b>string</b> is the filename.
MMEMemory:INITialize [<string>[,<char>[,<num>]]]	command only	Format a disk — <b>string</b> is the mass storage device <b>MEM:</b> (internal memory), <b>INT:</b> (internal floppy disk drive) or <b>EXT:</b> . Choose the disk format <b>char</b> from DOS or <b>LIF</b> , and the interleave factor <b>num</b> .
MMEMemory:LOAD:STATE 1, <string> <sup>5</sup>	command only	Recall an instrument state from mass storage — <b>string</b> is the filename.
MMEMemory:MDIRectory <string>	command only	Make a new directory on a DOS formatted disk.

5 Filenames may include the mass storage device — **MEM:** (internal memory), **INT:** (internal floppy disk drive) or **EXT:**. Wildcards ? and \* may be used.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
MMEMemory:MSIS <string>	STRING	Select a mass storage device — choose <b>MEM:</b> (internal memory), <b>INT:</b> (internal floppy disk drive) or <b>EXT:</b> .
MMEMemory:MOVE <string1>,<string2> <sup>5</sup>	command only	Move or rename a file — <b>string1</b> is the source (or old) filename and <b>string2</b> is the destination (or new) filename.
MMEMemory:RDIRectory <string>	command only	Delete a directory from a DOS formatted disk.
MMEMemory:STORE:STATE 1, <string> <sup>5</sup>	command only	Save an instrument state to mass storage — <b>string</b> is the filename.
MMEMemory:STORE:STATE:ISTate <ON OFF> <sup>2</sup>	NR1	Turn on/off the instrument state — part of the definition of a saved file.
MMEMemory:STORE:STATE:CORRection <ON OFF> <sup>2</sup>	NR1	Turn on/off the calibration — part of the definition of a saved file.
MMEMemory:STORE:STATE:TRACe <ON OFF> <sup>2</sup>	NR1	Turn on/off the data trace — part of the definition of a saved file.
MMEMemory:STORE:TRACe <char>,<string> <sup>5</sup>	command only	Stores an ASCII list of trace and frequency values to a file — <b>char</b> is the formatted data trace CH<1 2>FDATA and <b>string</b> is the filename.

#### OUTPut

OUTPut[:STATE] <ON OFF> <sup>2</sup>	NR1	Turn on/off RF power from the source.
--------------------------------------	-----	---------------------------------------

#### PROGram<sup>6</sup>

PROGram <sup>6</sup> :CATalog?	query only STRING	List the names of the defined IBASIC programs — response is "PROG" (if a program is present) or the null string ("").
PROGram <sup>6</sup> [:SELected] <sup>7</sup> :DEFIne <block>	BLOCK	Download an IBASIC program from an external controller.
PROGram <sup>6</sup> [:SELected] <sup>7</sup> :DELetE:ALL	command only	Delete all IBASIC programs from the program buffer — equivalent to an HP BASIC SCRATCH A command.

<sup>6</sup> Commands in the PROGram subsystem are only available when the HP Instrument BASIC (IBASIC) option is installed (option 1C2). They allow you to generate and control IBASIC programs in the analyzer.

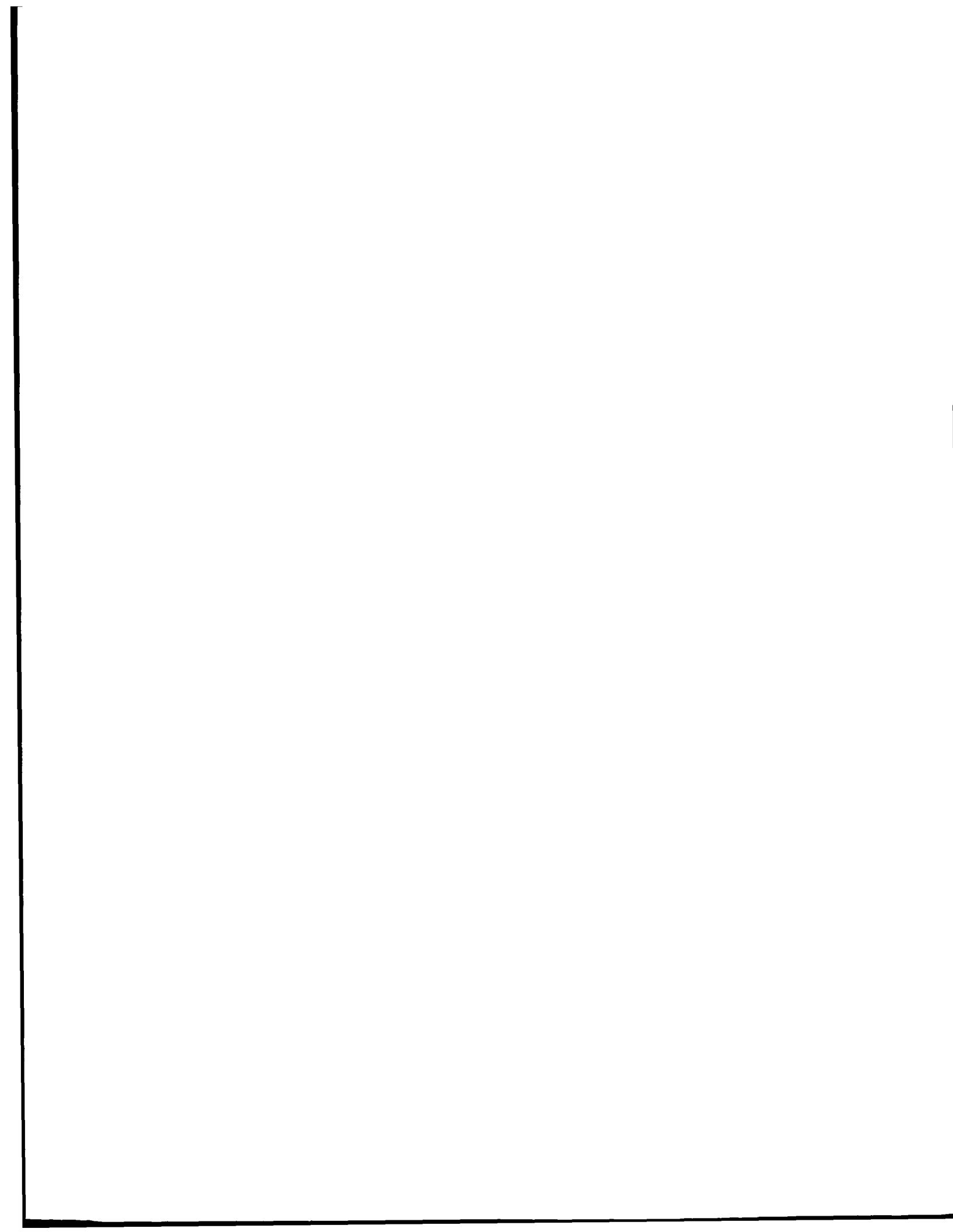
<sup>7</sup> Commands grouped under the SELected mnemonic in the PROGram subsystem operate on the active program buffer.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
PROGram <sup>6</sup> [:SElected] <sup>7</sup> :DELETE [:SElected]	command only	Delete the active IBASIC program — equivalent to an HP BASIC SCRATCH A command.
PROGram <sup>6</sup> [:SElected] <sup>7</sup> :EXECute <string>	command only	Execute an IBASIC command.
PROGram <sup>6</sup> [:SElected] <sup>7</sup> :MALLOCate <num>	NR1	Allocate memory space for IBASIC programs — choose from a real number between 2048 and 4000000 bytes.
PROGram <sup>6</sup> [:SElected] <sup>7</sup> :NAME 'PROG'	STRING	Select the IBASIC program in the program buffer to be active.
PROGram <sup>6</sup> [:SElected] <sup>7</sup> :NUMBER <string>, <data> <sup>1</sup>	BLOCK or NR3 <sup>1</sup>	Load a new value for a numeric variable string in the active IBASIC program — num is the new value.
PROGram <sup>6</sup> [:SElected] <sup>7</sup> :STATE <char>	CHAR	Select the state of the active IBASIC program — choose from STOP PAUSE RUN CONTinue.
PROGram <sup>6</sup> [:SElected] <sup>7</sup> :STRing <string1>, <string2>	STRING	Load a new value for a string variable string1 in the active IBASIC program — string2 is the new value.
PROGram <sup>6</sup> [:SElected] <sup>7</sup> :WAIT	NR1	Wait until the IBASIC program completes.
<b>SENSe[1 2]</b>		
SENSe[1 2]:AVERage:CLEAR	command only	Re-start the trace averaging function.
SENSe[1 2]:AVERage:COUNT <num>	NR1	Specify a count or weighting factor for the averaged measurement data.
SENSe[1 2]:AVERage[:STATe] <ON OFF> <sup>2</sup>	NR1	Turn on/off the trace averaging function.
SENSe[1 2]:BWIDth[:RESolution] <num> HZ	NR2	Specify the bandwidth of the IF receiver (narrow, medium or wide) to be used in the measurement — choose 250 (narrow) 750 (medium) or 6500 (wide).
SENSe[1 2]:CORRection:COLlect [:ACQuire] <char>	command only	Measure a calibration standard — select from STANDARD1 STANDARD2 STANDARD3.

<sup>7</sup> Commands grouped under the SESelected mnemonic in the PROGram subsystem operate on the active program buffer.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
SENSe[1 2]:CORRection:COLLect [:CKIT[:SElect]]	STRING	Select Cal Kit Choose from one of the following strings: 'COAX,7MM,TYPE-N,50,FEMALE' 'COAX,7MM,TYPE-N,50,MALE' 'COAX,3.5,APC-3.5,50,IMPLIED' 'USER,IMPLIED,IMPLIED,IMPLIED, IMPLIED'  'COAX,7MM,TYPE-N,75,FEMALE' 'COAX,7MM,TYPE-N,75,MALE'
SENSe[1 2]:CORRection:COLLect [:ISTate[:AUTO] <ON OFF> <sup>2</sup> ]	NR1	Select the instrument state for calibration — choose Full Band (ON) or User Defined (OFF).
SENSe[1 2]:CORRection:COLLect [:METHod <char>]	command only	Select the type of calibration — choose from TRAN1 or REFL3.
SENSe[1 2]:CORRection:COLLect:SAVE	command only	Complete and save current calibration.
SENSe[1 2]:CORRection:CSET [:SElect] DEFault	command only	Restore the “factory” default calibration for the current measurement and channel.
SENSe[1 2]:CORRection:CSET [:SElect]? <sup>3</sup>	query only CHAR	Query the current calibration type — returns DEF (factory default), FULL (full band) or USER (user defined).
SENSe[1 2]:CORRection[:STATE] <ON OFF> <sup>2</sup>	NR1	Turn on/off calibration.
SENSe[1 2]:COUPLE <char>	CHAR	Turn on/off the alternate sweep mode — choose ALL (coupled sweep) or NONE (alternate sweep).
SENSe[1 2]:DETector[:FUNCTION] <char>	CHAR	Specify which detection mode is used to make the measurement — choose BBAND (broadband) or NBAND (narrowband).
SENSe[1 2]:FREQuency:CENTER <num> <sup>3</sup>	NR3	Set the center frequency of the RF source.
SENSe[1 2]:FREQuency:SPAN <num> <sup>3</sup>	NR3	Set the frequency span of the RF source.
SENSe[1 2]:FREQuency:STARt <num> <sup>3</sup>	NR3	Set the start frequency of the RF source.
SENSe[1 2]:FREQuency:STOP <num> <sup>3</sup>	NR3	Set the stop frequency of the RF source.
SENSe[1 2]:FUNCTION?	query only STRING	Query the measurement function — returns one of the 'XFR:POW ...' or 'XFR:POW:RAT ...' strings described below.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
SENSe[1 2]:FUNCTION 'XFREQUENCY :POWER <num>'	command only	Specify that the receiver will measure the power into a single channel — choose from channels 0 (R), 1 (A), 2 (B), 11 (Ext X) or 12 (Ext Y).
SENSe[1 2]:FUNCTION 'XFREQUENCY :POWER:RATIO <num>,<num>'	command only	Specify that the receiver will measure a ratio of the power into the two selected channels — choose from ratios 1,0 (A/R), 2,0 (B/R), 12,0 (Ext Y/R), 11,12 (Ext X/Ext Y) or 12,11 (Ext Y/Ext X).
SENSe[1 2]:ROSCILLATOR:SOURce <char>	CHAR	Specify the source of the reference oscillator — select INTERNAL or EXTERNAL.
SENSe[1 2]:STATE <ON OFF> <sup>2</sup>	NR1	Turn on/off the specified channel.
SENSe[1 2]:SWEep:POINTS <num>	NR1	Set the number of data points for the measurement — choose from 51 101 201 401 801 1601.
SENSe[1 2]:SWEep:TIME <num> <sup>3</sup>	NR3	Set the sweep time.
SENSe[1 2]:SWEep:TIME:AUTO <ON OFF ONCE> <sup>2</sup>	NR1	Turn on/off the automatic sweep time function.
SENSe:SWEep:TRIGGER:SOURce <char>	CHAR	Set the trigger source for each point in a sweep — choose IMMEDIATE or EXTERNAL (used in conjunction with TRIGGER[:SEQUENCE]:SOURce).
<b>SOURce</b>		
SOURce[1 2]:POWER[:LEVEL] [:IMMEDIATE] [:AMPLITUDE] <num> <sup>3</sup>	NR3	Set the RF power output from the source.
<b>STATus</b>		
STATus:DEVICE:CONDITION?	query only NR1	Read and clear the Device Status condition register <sup>8</sup> .
STATus:DEVICE:ENABLE <num>	NR1	Set and query bits in the Device Status enable register <sup>9</sup> .
STATus:DEVICE[:EVENT]?	query only NR1	Read and clear the Device Status event register <sup>8</sup> .
STATus:DEVICE:NTRANSITION <num>	NR1	Set and query bits in the Device Status negative transition register <sup>9</sup> .
STATus:DEVICE:PTRANSITION <num>	NR1	Set and query bits in the Device Status positive transition register <sup>9</sup> .



SUBSYSTEM COMMANDS	FORM	DESCRIPTION
STATus:OPERation:AVERaging :CONDition?	query only NR1	Read the Averaging status condition register <sup>8</sup> .
STATus:OPERation:AVERaging:ENABLE <num>	NR1	Set and query bits in the Averaging status enable register <sup>9</sup> .
STATus:OPERation:AVERaging[:EVENT]?	query only NR1	Read and clear the Averaging status event register <sup>8</sup> .
STATus:OPERation:AVERaging :NTRansition <num>	NR1	Set and query bits in the Averaging status negative transition register <sup>9</sup> .
STATus:OPERation:AVERaging :PTRansition <num>	NR1	Set and query bits in the Averaging status positive transition register <sup>9</sup> .
STATus:OPERation:CONDition?	query only NR1	Read the Operational Status condition register <sup>8</sup> .
STATus:OPERation:ENABLE <num>	NR1	Set and query bits in the Operational Status enable register <sup>9</sup> .
STATus:OPERation[:EVENT]?	query only NR1	Read and clear the Operational Status event register <sup>8</sup> .
STATus:OPERation:MEASuring :CONDition?	query only NR1	Read the Measuring status condition register <sup>8</sup> .
STATus:OPERation:MEASuring:ENABLE <num>	NR1	Set and query bits in the Measuring status enable register <sup>9</sup> .
STATus:OPERation:MEASuring[:EVENT]?	query only NR1	Read and clear the Measuring status event register <sup>8</sup> .
STATus:OPERation:MEASuring :NTRansition <num>	NR1	Set and query bits in the Measuring status negative transition register <sup>9</sup> .
STATus:OPERation:MEASuring :PTRansition <num>	NR1	Set and query bits in the Measuring status positive transition register <sup>9</sup> .
STATus:OPERation:NTRansition <num>	NR1	Set and query bits in the Operational Status negative transition register <sup>9</sup> .
STATus:OPERation:PTRansition <num>	NR1	Set and query bits in the Operational Status positive transition register <sup>9</sup> .
STATus:PRESet	command only	Set bits in most enable and transition registers to their default state.
STATus:QUESTIONable:CONDition?	query only NR1	Read and clear the Questionable Status condition register <sup>8</sup> .

8 Returns the sum of the decimal weights ( $2^n$  where n is the bit number) of all bits currently set. For more information on using the status registers refer to the *HP-IB Programming Reference*.

9 num is the sum of the decimal weights of all bits to be set.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
STATus:QUEStionable:ENABLE <num>	NR1	Set and query bits in the Questionable Status enable register <sup>9</sup> .
STATus:QUEStionable[:EVENT]?	query only NR1	Read and clear the Questionable Status event register <sup>8</sup> .
STATus:QUEStionable:LIMit :CONDition?	query only NR1	Read and clear the Limit Fail condition register <sup>8</sup> .
STATus:QUEStionable:LIMit:ENABLE <num>	NR1	Set and query bits in the Limit Fail enable register <sup>9</sup> .
STATus:QUEStionable:LIMit[:EVENT]?	query only NR1	Read and clear the Limit Fail event register <sup>8</sup> .
STATus:QUEStionable:LIMit :NTRansition <num>	NR1	Set and query bits in the Limit Fail negative transition register <sup>9</sup> .
STATus:QUEStionable:LIMit :PTRansition <num>	NR1	Set and query bits in the Limit Fail positive transition register <sup>9</sup> .
STATus:QUEStionable:NTRansition <num>	NR1	Set and query bits in the Questionable Status negative transition register <sup>9</sup> .
STATus:QUEStionable:PTRansition <num>	NR1	Set and query bits in the Questionable Status positive transition register <sup>9</sup> .
<b>SYSTem</b>		
SYSTem:BEEPer:VOLUME <num>	NR2	Set the volume of the beeper — num is a number between 0 for 0% and 1 for 100%.
SYSTem:COMMUnicatE:GPIB:CONTroller [:STATE] <ON OFF> <sup>2,10</sup>	NR1	Makes the HP 8711 the system controller.
SYSTem:COMMUnicatE:GPIB:ECHO <ON OFF> <sup>2</sup>	NR1	Turn on/off HP-IB mnemonic echo.
SYSTem:COMMUnicatE:GPIB:HCOPy :ADDResS <num>	NR1	Set the address of an HP-IB printer or plotter for hardcopy output — num must be an integer between 0 and 30.
SYSTem:COMMUnicatE:GPIB:MMEMory :ADDResS <num>	NR1	Set the HP-IB address of an external disk drive — num must be an integer between 0 and 30.
SYSTem:COMMUnicatE:GPIB:MMEMory :UNIT <num>	NR1	Set the unit number of an external disk drive.
SYSTem:COMMUnicatE:GPIB:MMEMory :VOLume <num>	NR1	Set the volume number of an external disk drive.

10 For use with IBASIC — this command cannot be executed from an external controller.

11 A delay of 5 seconds is required before a command is sent to the new address.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
SYSTem:COMMUnicatE:GPIB[:SELF] :ADDReSS <num> <sup>11</sup>	NR1	Set the HP 8711's HP-IB address — num must be an integer between 0 and 30.
SYSTem:COMMUnicatE:SERial:TRANsmiT :BAUD <num>	NR1	Set the baud rate for hardcopy output to a device on the serial port — choose from 1200 2400 4800 9600 19200.
SYSTem:COMMUnicatE:SERial:TRANsmiT :HANDshake <char>	CHAR	Set the handshake for communication to a hardcopy device on the serial port — choose XON or DTR.
SYSTem:DATE <num1>,<num2>,<num3>	NR1, NR1, NR1	Set the year (num1), month (num2) and day (num3) of the real time clock.
SYSTem:ERRor?	query only NR1, STRING	Query the error queue — returns the error number and message.
SYSTem:KEY:MASK?	query only NR1	Query the mask (shift, ctrl, alt) associated with a keypress on an external keyboard.
SYSTem:KEY:QUEue:CLEar	command only	Clears the key queue.
SYSTem:KEY:QUEue:COUNT?	query only NR1	Query the number of key codes in the queue.
SYSTem:KEY:QUEue:MAXimum?	query only NR1	Query the size of the key queue (the maximum number of key codes it can hold).
SYSTem:KEY:QUEue[:STATe] <ON OFF> <sup>2</sup>	NR1	Turn on/off the key queue.
SYSTem:KEY:TYPE?	query only CHAR	Query the type of key that was pressed — returns NONE, RPG, KEY (front panel key) or ASC (external keyboard).
SYSTem:KEY[:VALue]?	query only NR1	Query the key code value for the last key pressed — RPG type returns the knob count, positive for clockwise rotation, KEY type returns the front panel keycode, <sup>12</sup> and ASC type returns the ASCII code number.
SYSTem:PRESet	command only	Perform a system preset — this is the same as the front panel [PRESET] key.
SYSTem:SET <block>	command only	Send a learn string (obtained using *LRN?) to the analyzer — this command is included in the learn string.

12 A list of the HP 8711's front panel keycodes is provided in the *HP-IB Programming Guide*.

SUBSYSTEM COMMANDS	FORM	DESCRIPTION
SYSTem:TIME <num1>,<num2>,<num3>	NR1,NR1, NR1	Set the hour ( <b>num1</b> ), minute ( <b>num2</b> ) and second ( <b>num3</b> ) of the real time clock.
SYSTem:VERSiOn?	query only NR2	Query the SCPI version of the analyzer.
<b>TEST</b>		
TEST:RESULT?	query only CHAR	Query the result of the selected adjustment or self-test — the response will be <b>NULL PASS FAIL</b> .
TEST:SElect <num>	NR1	Select the adjustment or self-test to execute.
TEST:STATE <char>	CHAR	Select the state of the active adjustment or self-test — choose from <b>RUN CONTinue STOP</b> for the command. Query returns <b>NULL RUn PAUs DONE</b> .
TEST:VALue <num>	NR1	Set or query a value for an adjustment or self-test.
<b>TRACe</b>		
TRACe[:DATA]? <char>	query only BLOCK or NR3 <sup>1</sup>	Query trace data — choose from <b>CH&lt;1 2&gt;FDATA</b> formatted data, <b>CH&lt;1 2&gt;FMEM</b> formatted memory, <b>CH&lt;1 2&gt;SDATA</b> unformatted data, <b>CH&lt;1 2&gt;SMEM</b> unformatted memory, <b>CH&lt;1 2&gt;&lt;A B R&gt;FWD</b> raw data, or <b>CH&lt;1 2&gt;SCORR&lt;1 2 3&gt;</b> correction data. Note: See "Internal Measurement Arrays" next, for data array details.
TRACe[:DATA] <char>,<data>	command only	Input trace data — choose from the above list of arrays. The data can be either BLOCK or NR3 type. <sup>1</sup>
TRACe[:DATA] <char1>,<char2>	command only	Move data from one internal array to another — <b>char1</b> is the target array ( <b>CH&lt;1 2&gt;SMEM</b> ) while <b>char2</b> is the source array ( <b>CH&lt;1 2&gt;SDATA</b> ). Note that the source and target arrays must be from the same measurement channel.
<b>TRIGger</b>		
TRIGger[:SEQUence]:SOURCE <char>	CHAR	Set the source for the sweep trigger signal — choose <b>IMMEDIATE</b> or <b>EXTernal</b> (used in conjunction with <b>SENSe:SWEep:TRIGger:SOURce</b> ).

---

## SCPI Conformance Information

The HP 8711 RF Network Analyzer conforms to the 1992.0 version of SCPI.

### SCPI Standard Commands

The following are the SCPI 1992.0 standard commands implemented by the HP 8711 RF Network Analyzer.

```
*CLS
*ESE
*ESE?
*ESR?
*IDN?
*LRN?
*OPC
*OPC?
*OPT?
*PCB
*PSC
*RST
*SRE
*SRE?
*STB?
*TST?
*WAI

ABORt

CALCulate[1|2]:DATA?
CALCulate[1|2]:FORMAT
CALCulate[1|2]:FORMAT?
CALCulate[1|2]:LIMit:STATE
CALCulate[1|2]:LIMit:STATE?
CALCulate[1|2]:MATH[:EXPRESSION]
CALCulate[1|2]:MATH[:EXPRESSION]?

CALibration:ZERO:AUTO
CALibration:ZERO:AUTO?

DISPlay:MENU:KEY[1|2|... 7]
DISPlay:MENU:KEY[1|2|... 7]?
DISPlay:WINDOW[1|2|10]:GEOMETRY:LLEFT?
DISPlay:WINDOW[1|2|10]:GEOMETRY:SIZE?
DISPlay:WINDOW[1|2|10]:GEOMETRY:URIGHT?
DISPlay:WINDOW[1|2|10]:GRAPHics:CLEAR
DISPlay:WINDOW[1|2|10]:GRAPHics:COLOR
DISPlay:WINDOW[1|2|10]:GRAPHics:COLOR?
DISPlay:WINDOW[1|2|10]:GRAPHics[:DRAW]
DISPlay:WINDOW[1|2|10]:GRAPHics:LABEL
DISPlay:WINDOW[1|2|10]:GRAPHics:MOVE
DISPlay:WINDOW[1|2|10]:GRAPHics:MOVE?
```

DISPlay:WINDOW[1|2|10]:GRAPhics:STATE?  
DISPlay:WINDOW[1|2]:TRACe:GRATicule:GRID[:STATE]  
DISPlay:WINDOW[1|2]:TRACe:GRATicule:GRID[:STATE]?  
DISPlay:WINDOW[1|2]:TRACe[1|2][:STATE]  
DISPlay:WINDOW[1|2]:TRACe[1|2][:STATE]?  
DISPlay:WINDOW[1|2]:TRACe:Y[:SCALE]:AUTO  
DISPlay:WINDOW[1|2]:TRACe:Y[:SCALE]:PDIVision  
DISPlay:WINDOW[1|2]:TRACe:Y[:SCALE]:PDIVision?  
DISPlay:WINDOW[1|2]:TRACe:Y[:SCALE]:RLEVel  
DISPlay:WINDOW[1|2]:TRACe:Y[:SCALE]:RLEVel?  
DISPlay:WINDOW[1|2]:TRACe:Y[:SCALE]:RPOSITION  
DISPlay:WINDOW[1|2]:TRACe:Y[:SCALE]:RPOSITION?  
  
FORMAT:BORDer  
FORMAT:BORDer?  
FORMAT[:DATA]  
FORMAT[:DATA]?  
  
INITiate[1|2]:CONTinuous  
INITiate[1|2]:CONTinuous?  
INITiate[1|2][:IMMEDIATE]  
  
MMEMory:CDIRectory  
MMEMory:CDIRectory?  
MMEMory:COPY  
MMEMory:DELetE  
MMEMory:INITialize  
MMEMory:LOAD:STATE  
MMEMory:MOVE  
MMEMory:MSIS  
MMEMory:MSIS?  
MMEMory:STORe:STATE  
MMEMory:STORe:TRACe  
  
OUTPut[:STATE]  
OUTPut[:STATE]?  
  
PROGram:CATalog?  
PROGram[:SElected]:DEFine  
PROGram[:SElected]:DEFine?  
PROGram[:SElected]:DELetE:ALL  
PROGram[:SElected]:DELetE[:SElected]  
PROGram[:SElected]:EXECute  
PROGram[:SElected]:MALLocate  
PROGram[:SElected]:MALLocate?  
PROGram[:SElected]:NAME  
PROGram[:SElected]:NAME?  
PROGram[:SElected]:NUMBer  
PROGram[:SElected]:NUMBer?  
PROGram[:SElected]:STATE  
PROGram[:SElected]:STATE?  
PROGram[:SElected]:STRing  
PROGram[:SElected]:STRing?

```

PROGram[:SELected]:WAIT
PROGram[:SELected]:WAIT?

SENSe[1|2]:AVERage:COUNt
SENSe[1|2]:AVERage:COUNt?
SENSe[1|2]:AVERage[:STATe]
SENSe[1|2]:AVERage[:STATe]?
SENSe[1|2]:BWIDth[:RESolution]
SENSe[1|2]:BWIDth[:RESolution]?
SENSe[1|2]:CORRection:COLlect[:ACQuire]
SENSe[1|2]:CORRection:COLlect:METHod
SENSe[1|2]:CORRection:COLlect:SAVE
SENSe[1|2]:CORRection:CSET[:SELect]
SENSe[1|2]:CORRection:CSET[:SELect]?
SENSe[1|2]:CORRection[:STATe]
SENSe[1|2]:CORRection[:STATe]?
SENSe[1|2]:FREQuency:CENTer
SENSe[1|2]:FREQuency:CENTer?
SENSe[1|2]:FREQuency:SPAN
SENSe[1|2]:FREQuency:SPAN?
SENSe[1|2]:FREQuency:STARt
SENSe[1|2]:FREQuency:STARt?
SENSe[1|2]:FREQuency:STOP
SENSe[1|2]:FREQuency:STOP?
SENSe[1|2]:FUNCtion
SENSe[1|2]:FUNCtion?
SENSe:ROSCillator:SOURce
SENSe:ROSCillator:SOURce?
SENSe[1|2]:SWEep:POINTs
SENSe[1|2]:SWEep:POINTs?
SENSe[1|2]:SWEep:TIME
SENSe[1|2]:SWEep:TIME?
SENSe[1|2]:SWEep:TIME:AUTO
SENSe[1|2]:SWEep:TIME:AUTO?

SOURce[1|2]:POWer[:LEVel][:IMMEDIATE][:AMPLitude]
SOURce[1|2]:POWer[:LEVel][:IMMEDIATE][:AMPLitude]?

STATus:OPERation:CONDITION?
STATus:OPERation:ENABLE
STATus:OPERation:ENABLE?
STATus:OPERation[:EVENT]?
STATus:OPERation:NTRansition
STATus:OPERation:NTRansition?
STATus:OPERation:PTRansition
STATus:OPERation:PTRansition?
STATus:PRESet
STATus:QUESTIONable:CONDITION?
STATus:QUESTIONable:ENABLE
STATus:QUESTIONable:ENABLE?
STATus:QUESTIONable[:EVENT]?
STATus:QUESTIONable:NTRansition

```

```

STATus:QUEStionable:NTRansition?
STATus:QUEStionable:PTRansition
STATus:QUEStionable:PTRansition?

SYSTem:BEEPer:VOLume
SYSTem:BEEPer:VOLume?
SYSTem:COMMUnicatE:GPIB[:SELF]:ADDReSS
SYSTem:COMMUnicatE:GPIB[:SELF]:ADDReSS?
SYSTem:COMMUnicatE:SERial:TRANsmit:BAUD
SYSTem:COMMUnicatE:SERial:TRANsmit:BAUD?
SYSTem:DATE
SYSTem:DATE?
SYSTem:ERRor?
SYSTem:KEY[:VALue]?
SYSTem:PRESet
SYSTem:SET
SYSTem:TIME
SYSTem:TIME?
SYSTem:VERSion?

TRACe[:DATA]
TRACe[:DATA]?

TRIGger[:SEQUence]:SOURce
TRIGger[:SEQUence]:SOURce?

```

## Instrument Specific Commands

The following are instrument specific commands implemented by the HP 8711 RF Network Analyzer which are not part of the present SCPI 1992.0 definition.

```

CALCulate[1|2]:LIMit:DISPlay
CALCulate[1|2]:LIMit:DISPlay?
CALCulate[1|2]:LIMit:SEGMeNT[1|2|...12]:AMPLitude:STARt
CALCulate[1|2]:LIMit:SEGMeNT[1|2|...12]:AMPLitude:STARt?
CALCulate[1|2]:LIMit:SEGMeNT[1|2|...12]:AMPLitude:STOP
CALCulate[1|2]:LIMit:SEGMeNT[1|2|...12]:AMPLitude:STOP?
CALCulate[1|2]:LIMit:SEGMeNT:AOff
CALCulate[1|2]:LIMit:SEGMeNT[1|2|...12]:FREQuency:STARt
CALCulate[1|2]:LIMit:SEGMeNT[1|2|...12]:FREQuency:STARt?
CALCulate[1|2]:LIMit:SEGMeNT[1|2|...12]:FREQuency:STOP
CALCulate[1|2]:LIMit:SEGMeNT[1|2|...12]:FREQuency:STOP?
CALCulate[1|2]:LIMit:SEGMeNT[1|2|...12]:STATE
CALCulate[1|2]:LIMit:SEGMeNT[1|2|...12]:STATE?
CALCulate[1|2]:LIMit:SEGMeNT[1|2|...12]:TYPE
CALCulate[1|2]:LIMit:SEGMeNT[1|2|...12]:TYPE?
CALCulate[1|2]:MARKer:AOff
CALCulate[1|2]:MARKer:BWIDth
CALCulate[1|2]:MARKer:BWIDth?
CALCulate[1|2]:MARKer:FUNCTION:RESult?

```

```
CALCulate[1|2]:MARKer:FUNCTION[:SElect]
CALCulate[1|2]:MARKer:FUNCTION[:SElect]?
CALCulate[1|2]:MARKer:FUNCTION:TRACking
CALCulate[1|2]:MARKer:FUNCTION:TRACking?
CALCulate[1|2]:MARKer[1|2|...8]:MAXimum
CALCulate[1|2]:MARKer[1|2|...8]:MINimum
CALCulate[1|2]:MARKer:MODE
CALCulate[1|2]:MARKer:MODE?
CALCulate[1|2]:MARKer[1|2|...8]:POINT
CALCulate[1|2]:MARKer[1|2|...8]:POINT?
CALCulate[1|2]:MARKer:REFerence:X?
CALCulate[1|2]:MARKer:REFerence:Y?
CALCulate[1|2]:MARKer[1|2|...8][:STATe]
CALCulate[1|2]:MARKer[1|2|...8][:STATe]?
CALCulate[1|2]:MARKer[1|2|...8]:TARGet
CALCulate[1|2]:MARKer[1|2|...8]:TARGet?
CALCulate[1|2]:MARKer[1|2|...8]:X
CALCulate[1|2]:MARKer[1|2|...8]:X?
CALCulate[1|2]:MARKer[1|2|...8]:Y?
CALCulate[1|2]:MARKer[1|2|...8]:Y?

CONFigure
CONFigure?

DIAGnostic:CCONstants:LOAD
DIAGnostic:CCONstants:STORe:DISK
DIAGnostic:CCONstants:STORe:EEPROM
DIAGnostic:CPU:MEMORY[:WORD]
DIAGnostic:CPU:MEMORY[:WORD]?
DIAGnostic:DITHer
DIAGnostic:DITHer?
DIAGnostic:SNUMber
DIAGnostic:SNUMber?
DIAGnostic:SPUR:AVOid
DIAGnostic:SPUR:AVOid?

DISPlay:ANNotation:CLOCk:DATE:FORMAT
DISPlay:ANNotation:CLOCk:DATE:FORMAT?
DISPlay:ANNotation:CLOCk:DATE:MODE
DISPlay:ANNotation:CLOCk:DATE:MODE?
DISPlay:ANNotation:CLOCk:MODE
DISPlay:ANNotation:CLOCk:MODE?
DISPlay:ANNotation:CLOCk:SEConds[:STATe]
DISPlay:ANNotation:CLOCk:SEConds[:STATe]?
DISPlay:ANNotation:FREQuency[1|2]:MODE
DISPlay:ANNotation:FREQuency[1|2]:MODE?
DISPlay:ANNotation:FREQuency:RESolution
DISPlay:ANNotation:FREQuency:RESolution?
DISPlay:ANNotation:MARKer[1|2][:STATe]
DISPlay:ANNotation:MARKer[1|2][:STATe]?
DISPlay:ANNotation:MESSge:AOFF
DISPlay:ANNotation:MESSge:STATE
DISPlay:ANNotation:MESSge:STATE?
```

```
DISPlay:ANNotation:TITLe[1|2]:DATA
DISPlay:ANNotation:TITLe[1|2]:DATA?
DISPlay:ANNotation:TITLe[:STATe]
DISPlay:ANNotation:TITLe[:STATe]?
DISPlay:ANNotation:YAXis:MODE
DISPlay:ANNotation:YAXis:MODE?
DISPlay:ANNotation:YAXis[:STATe]
DISPlay:ANNotation:YAXis[:STATe]?
DISPlay:FORMAT
DISPlay:FORMAT?
DISPlay:PROGraM[:MODE]
DISPlay:PROGraM[:MODE]?
DISPlay:WINDOW:GRAphics:BUFFER[:STATe]
DISPlay:WINDOW:GRAphics:BUFFER[:STATe]?
DISPlay:WINDOW[1|2|10]:GRAphics:CIRCLE
DISPlay:WINDOW[1|2|10]:GRAphics:LABel:FONT
DISPlay:WINDOW[1|2|10]:GRAphics:LABel:FONT?
DISPlay:WINDOW[1|2|10]:GRAphics:RECTangle

HCOPy:ABORT
HCOPy:DEViCe[1|2]:COLor
HCOPy:DEViCe[1|2]:COLor?
HCOPy:DEViCe:LANGuage
HCOPy:DEViCe:LANGuage?
HCOPy:DEViCe:MODE
HCOPy:DEViCe:MODE?
HCOPy:DEViCe:PORT
HCOPy:DEViCe:PORT?
HCOPy:DEViCe:RESolution
HCOPy:DEViCe:RESolution?
HCOPy[:IMMediate]
HCOPy:ITEM:ANNotation:STATE
HCOPy:ITEM:ANNotation:STATE?
HCOPy:ITEM[1|2]:FFEd:STATE
HCOPy:ITEM[1|2]:FFEd:STATE?
HCOPy:ITEM:GRATicule:STATE
HCOPy:ITEM:GRATicule:STATE?
HCOPy:ITEM:MARKer:STATE
HCOPy:ITEM:MARKer:STATE?
HCOPy:ITEM:TITLe:STATE
HCOPy:ITEM:TITLe:STATE?
HCOPy:ITEM:TRACe:STATE
HCOPy:ITEM:TRACe:STATE?
HCOPy:PAGE:MARGIN:LEFT
HCOPy:PAGE:MARGIN:LEFT?
HCOPy:PAGE:MARGIN:TOP
HCOPy:PAGE:MARGIN:TOP?
HCOPy:PAGE:ORIentation
HCOPy:PAGE:ORIentation?
HCOPy:PAGE:WIDTH
HCOPy:PAGE:WIDTH?
```

MMEMemory:MDIRectomy  
MMEMemory:RDIRectomy  
MMEMemory:STORE:STATE:CORRection  
MMEMemory:STORE:STATE:CORRection?  
MMEMemory:STORE:STATE:ISTate  
MMEMemory:STORE:STATE:ISTate?  
MMEMemory:STORE:STATE:TRACe  
MMEMemory:STORE:STATE:TRACe?  
  
SENSe[1|2]:AVERage:CLEar  
SENSe[1|2]:CORRection:COLLect:CKIT[:SElect]  
SENSe[1|2]:CORRection:COLLect:CKIT[:SElect]?  
SENSe[1|2]:CORRection:COLLect:ISTate[:AUTO]  
SENSe[1|2]:CORRection:COLLect:ISTate[:AUTO]?  
SENSe:COUPLE  
SENSe:COUPLE?  
SENSe[1|2]:DETector[:FUNCTION]  
SENSe[1|2]:DETector[:FUNCTION]?  
SENSe[1|2]:STATE  
SENSe[1|2]:STATE?  
SENSe:SWEep:TRIGger:SOURce  
SENSe:SWEep:TRIGger:SOURce?  
  
STATus:DEVice:CONDition?  
STATus:DEVice:ENABLE  
STATus:DEVice:ENABLE?  
STATus:DEVice[:EVENT]?  
STATus:DEVice:NTRansition  
STATus:DEVice:NTRansition?  
STATus:DEVice:PTRansition  
STATus:DEVice:PTRansition?  
STATus:OPERation:AVERaging:CONDition?  
STATus:OPERation:AVERaging:ENABLE  
STATus:OPERation:AVERaging:ENABLE?  
STATus:OPERation:AVERaging[:EVENT]?  
STATus:OPERation:AVERaging:NTRansition  
STATus:OPERation:AVERaging:NTRansition?  
STATus:OPERation:AVERaging:PTRansition  
STATus:OPERation:AVERaging:PTRansition?  
STATus:OPERation:MEASuring:CONDition?  
STATus:OPERation:MEASuring:ENABLE  
STATus:OPERation:MEASuring:ENABLE?  
STATus:OPERation:MEASuring[:EVENT]?  
STATus:OPERation:MEASuring:NTRansition  
STATus:OPERation:MEASuring:NTRansition?  
STATus:OPERation:MEASuring:PTRansition  
STATus:OPERation:MEASuring:PTRansition?  
STATus:QUESTIONable:LIMit:CONDition?  
STATus:QUESTIONable:LIMit:ENABLE  
STATus:QUESTIONable:LIMit:ENABLE?  
STATus:QUESTIONable:LIMit[:EVENT]?  
STATus:QUESTIONable:LIMit:NTRansition

STATus:QUESTIONable:LIMit:NTRansition?  
STATus:QUESTIONable:LIMit:PTRansition  
STATus:QUESTIONable:LIMit:PTRansition?  
  
SYSTem:COMMUnicatE:GPIB:CONTroller[:STATe]  
SYSTem:COMMUnicatE:GPIB:CONTroller[:STATe]?  
SYSTem:COMMUnicatE:GPIB:ECHO  
SYSTem:COMMUnicatE:GPIB:ECHO?  
SYSTem:COMMUnicatE:GPIB:HCOPy:ADDResS  
SYSTem:COMMUnicatE:GPIB:HCOPy:ADDResS?  
SYSTem:COMMUnicatE:GPIB:MMEMory:ADDResS  
SYSTem:COMMUnicatE:GPIB:MMEMory:ADDResS?  
SYSTem:COMMUnicatE:GPIB:MMEMory:UNIT  
SYSTem:COMMUnicatE:GPIB:MMEMory:UNIT?  
SYSTem:COMMUnicatE:GPIB:MMEMory:VOLume  
SYSTem:COMMUnicatE:GPIB:MMEMory:VOLume?  
SYSTem:COMMUnicatE:SERial:TRANsmiT:HANDshake  
SYSTem:COMMUnicatE:SERial:TRANsmiT:HANDshake?  
SYSTem:KEY:MASK?  
SYSTem:KEY:QUEue:CLEar  
SYSTem:KEY:QUEue:COUNT?  
SYSTem:KEY:QUEue:MAXimum?  
SYSTem:KEY:QUEue[:STATe]  
SYSTem:KEY:QUEue[:STATe]?  
SYSTem:KEY:TYPE?  
  
TEST:RESult?  
TEST:SELect  
TEST:SELect?  
TEST:STATE  
TEST:STATE?  
TEST:VALue  
TEST:VALue?

111-90110



H - P - and

58

filename 08711-90110  
Prognum

Model 8711

46

# **HP-IB Programming Guide**

**For the HP 8711 RF Network Analyzer**



**HP Part No. 08711-90109  
Printed in USA July 1993**



## HP-IB Programming

---

This document is an introduction to programming the HP 8711 RF network analyzer over the Hewlett-Packard Interface Bus (HP-IB). Its purpose is to provide concise information about the operation of the instrument under HP-IB control. It provides some background information on the HP-IB and a tutorial introduction using programming examples to demonstrate the remote operation of the HP 8711. The examples are provided on two disks that are included with the *HP 8711 Operating Manual*. Both disks contain the same examples, only the disk format is different.

- *Example Programs Disk — DOS Format* has examples written in HP Instrument BASIC (IBASIC). These programs can be run on the instrument's internal controller (option 1C2) or on a PC-compatible running Instrument BASIC for Windows (HP model no. E2200A).
- *Example Programs Disk — LIF Format* also has examples written in IBASIC. These programs can be run on the instrument's internal controller or on an HP series 300 workstation running HP BASIC (version 3.0 or higher).

The reader should become familiar with the operation of the HP 8711 network analyzer before controlling it over HP-IB. This document is not intended to teach programming or to discuss HP-IB theory except at an introductory level. Related information can be found in the following references:

- Information on the HP 8711's HP-IB command mnemonics is available in the *HP-IB Command Reference*.
- Information on making measurements with the HP 8711 is available in the *HP 8711 Operating Manual* and the *HP 8711 User's Guide*.
- Information on HP Instrument BASIC is available in the *HP Instrument BASIC User's Handbook* and *Using HP Instrument BASIC with the HP 8711*.
- Information concerning HP Instrument BASIC for Windows is available in *Installing and Using HP Instrument BASIC for Windows*.
- Information on HP BASIC programming is available in the manual set for the BASIC revision being used. For example: *BASIC 5.0 Programming Techniques* and *BASIC 5.0 Language Reference*.
- Information on using the HP-IB is available in the *Tutorial Description of the Hewlett-Packard Interface Bus* (HP literature no. 5952-0156).

## Table of Contents

HP-IB Overview	9a-3
Bus Structure	9a-3
Sending Commands	9a-5
HP-IB Requirements	9a-5
Interface Capabilities	9a-6
Programming Fundamentals	9a-7
Controller Capabilities	9a-7
Response to Bus Management Commands	9a-7
Message Exchange	9a-10
Synchronization	9a-12
Overlapped Commands	9a-12
Passing Control	9a-15
Transferring Data	9a-16
Data Types	9a-16
Data Encoding for Large Data Transfers	9a-18
Using the Status Registers	9a-19
General Status Register Model	9a-19
How to Use Registers	9a-21
The Service Request Process	9a-22
The HP 8711's Status Register Sets	9a-24
HP 8711 Register Set Summary	9a-30
User Graphics	9a-31
Window Geometry	9a-32
The Graphics Buffer	9a-32
Front Panel Keycodes	9a-33
Example Programs	9a-34
SETUP — Setting up a basic measurement	9a-37
MARKERS — Transferring data using the markers	9a-38
LIMITEST — Performing automatic PASS/FAIL testing with limit lines	9a-40
ASCDATA — Transferring data using the ASCII format	9a-43
REALDATA — Transferring data using the IEEE 64-bit floating point REAL format	9a-45
INTDATA — Transferring data using the 16-bit INTEGER format	9a-47
TRANCAL — Performing a transmission calibration	9a-49
REFLCAL — Performing a reflection calibration	9a-50
LOADCALS — Uploading and downloading correction arrays	9a-53
LEARNSTR — Using the learn string to upload and download instrument states	9a-57
SAVERCL — Saving and recalling instrument states, calibrations and data	9a-58
PRINTPLT — Using the serial and parallel ports for hardcopy output	9a-60
PASSCTRL — Using pass control and the HP-IB for hardcopy output	9a-62
AVERAGE — Generating a service request interrupt	9a-64
GRAPHICS — Using graphics and softkeys to create a customized procedure	9a-66
CALKIT — Instrument state file for downloading user-defined cal kit definitions	9a-71

---

## HP-IB Overview

HP-IB — the Hewlett-Packard Interface Bus — is a high-performance bus that allows individual instruments and computers to be combined into integrated test systems. The bus and its associated interface operations are defined by the IEEE 488.1 standard. The IEEE 488.2 standard defines the interface capabilities of instruments and controllers in a measurement system, including some frequently used commands.

HP-IB cables provide the physical link between devices on the bus. There are eight data lines on each cable that are used to send data from one device to another. Devices that send data over these lines are called **Talkers**. **Listeners** are devices that receive data over the same lines. There are also five control lines on each cable that are used to manage traffic on the data lines and to control other interface operations. **Controllers** are devices that use these control lines to specify the talker and listener in a data exchange.

When an HP-IB system contains more than one device with controller capabilities, only one of the devices is allowed to control data exchanges at any given time. The device currently controlling data exchanges is called the **Active Controller**. Also, only one of the controller-capable devices can be designated as the **System Controller**, the one device that can take control of the bus even if it is not the active controller. The HP 8711 can act as a talker, listener, active controller or system controller at different times.

HP-IB addresses provide a way to identify devices on the bus. The active controller uses HP-IB addresses to specify which device talks and which device listens during a data exchange. This means that each device's address must be unique. A device's address is set on the device itself, using either a front-panel key sequence or a rear-panel switch.

To set the HP-IB address on the HP 8711 network analyzer use the **HP8711 Address** softkey located in the **SYSTEM OPTIONS** / **HP-IB** menu. The factory default address for the HP 8711 is 16.

## Bus Structure

### Data Bus

The data bus consists of eight lines that are used to transfer data from one device to another. Programming commands and data sent on these lines is typically encoded in the ASCII format, although binary encoding is often used to speed up the transfer of large arrays. Both ASCII and binary data formats are available to the HP 8711. In addition, every byte transferred over HP-IB undergoes a handshake to ensure valid data.

## **Handshake Lines**

A three-line handshake scheme coordinates the transfer of data between talkers and listeners. This technique forces data transfers to occur at the speed of the slowest device, and ensures data integrity in multiple listener transfers. With most computing controllers and instruments, the handshake is performed automatically, which makes it transparent to the programmer.

## **Control Lines**

The data bus also has five control lines that the controller uses both to send bus commands and to address devices.

<b>IFC</b>	Interface Clear. Only the system controller uses this line. When this line is true (low) all devices (addressed or not) unaddress and go to an idle state.
<b>ATN</b>	Attention. The active controller uses this line to define whether the information on the data bus is a <b>command</b> or is <b>data</b> . When this line is true (low) the bus is in the command mode and the data lines carry bus commands. When this line is false (high) the bus is in the data mode and the data lines carry device-dependent instructions or data.
<b>SRQ</b>	Service Request. This line is set true (low) when a device requests service: the active controller services the requesting device. The analyzer can be enabled to pull the SRQ line for a variety of reasons.
<b>REN</b>	Remote Enable. Only the system controller uses this line. When this line is set true (low) the bus is in the remote mode and devices are addressed either to listen or talk. When the bus is in remote and a device is addressed, it receives instructions from HP-IB rather than from its front panel (pressing the <b>Return to Local</b> softkey returns the device to front panel operation). When this line is set false (high) the bus and all devices return to local operation.
<b>EOI</b>	End or Identify. This line is used by a talker to indicate the last data byte in a multiple byte transmission, or by an active controller to initiate a parallel poll sequence. The analyzer recognizes the EOI line as a terminator and it pulls the EOI line with the last byte of a message output (data, markers, plots, prints, error messages). The analyzer does not respond to parallel poll.

## Sending Commands

Commands are sent over the HP-IB via a controller's language system, such as IBASIC, QuickBASIC or C. The keywords used by a controller to send HP-IB commands vary among systems. When determining the correct keywords to use, keep in mind that there are two different kinds of HP-IB commands:

- Bus management commands, which control the HP-IB interface.
- Device commands, which control analyzer functions.

Language systems usually deal differently with these two kinds of HP-IB commands. For example, HP BASIC uses a unique keyword to send each bus management command, but always uses the keyword OUTPUT to send device commands.

The following example shows how to send a typical device command:

```
OUTPUT 716;"CALCULATE:MARKER:MAXIMUM"
```

This sends the command within the quotes (CALCULATE:MARKER:MAXIMUM) to the HP-IB device at address 716. If the device is an HP 8711, the command instructs the analyzer to set a marker to the maximum point on the data trace.

## HP-IB Requirements

**Number of Interconnected Devices:** 15 maximum

**Interconnection Path/ Maximum Cable Length:** 20 meters maximum or 2 meters per device, whichever is less.

**Message Transfer Scheme:** Byte serial/ bit parallel asynchronous data transfer using a 3-line handshake system.

**Data Rate:** Maximum of 1 megabyte per second over limited distances with tri-state drivers. Actual data rate depends on the transfer rate of the slowest device involved.

**Address Capability:** Primary addresses: 31 talk, 31 listen. A maximum of 1 talker and 14 listeners at one time.

**Multiple Controller Capability:** In systems with more than one controller (like the analyzer system), only one can be active at a time. The active controller can pass control to another controller, but only the system controller can assume unconditional control. Only one system controller is allowed. The system controller is hard-wired to assume bus control after a power failure.

## Interface Capabilities

The HP 8711 has the following interface capabilities, as defined by the IEEE 488.1 standard:

SH1	full Source handshake capability
AH1	full Acceptor handshake capability
T6	basic Talker, Serial Poll, no Talk Only, unaddress if MLA
TE0	no Extended Talker capability
L4	basic Listener, no Listen Only, unaddress if MTA
LE0	no Extended Listener capability
SR1	full Service Request capability
RL1	full Remote/Local capability
DC1	full Device Clear capability
C1	System Controller capability
C2	send IFC and take charge Controller capability
C3	send REN Controller capability
C4*	respond to SRQ
C6*	send IFC, receive control, pass control, parallel poll, pass control to self
C10*	send IFC, receive control, pass control, parallel poll
C12**	send IF messages, receive control, pass control
E2	tri-state drivers

\* only when an HP Instrument BASIC program is running

\*\* only when an HP Instrument BASIC program is not running

---

## Programming Fundamentals

This section includes specific information that is needed to program an HP 8711 network analyzer. It includes how the analyzer interacts with a controller, how data is transferred between the analyzer and a controller, and how to use the analyzer's status register structure to generate service requests.

### Controller Capabilities

The HP 8711 can be configured as an HP-IB system controller or as a talker/listener on the bus. To configure the analyzer, select either the **System Controller** or the **Talker/Listener** softkey in the **SYSTEM OPTIONS** **HP-IB** menu.

The HP 8711 is not usually configured as the system controller unless it is the only controller on the bus. This setup would be used if the analyzer only needed to control printers or plotters. It would also be used if HP Instrument BASIC was being used to control other test equipment.

When the analyzer is used with another controller on the bus, it is usually configured as a talker/listener. In this configuration, when the analyzer is passed control it can function as the active controller.

### Response to Bus Management Commands

The HP-IB contains an attention (ATN) line that determines whether the interface is in command mode or data mode. When the interface is in command mode (ATN TRUE) a controller can send bus management commands over the bus. Bus management commands specify which devices on the interface can talk (send data) and which can listen (receive data). They also instruct devices on the bus, either individually or collectively, to perform a particular interface operation.

This section describes how the HP 8711 responds to the HP-IB bus management commands. The commands themselves are defined by the IEEE 488.1 standard. Refer to the documentation for your controller's language system to determine how to send these commands.

#### Device Clear (DCL)

When the analyzer receives this command, it:

- Clears its input and output queues.
- Resets its command parser (so it is ready to receive a new program message).
- Cancels any pending \*OPC command or query.

The command does not affect:

- Front panel operation.
- Any analyzer operations in progress (other than those already mentioned).
- Any instrument settings or registers (although clearing the output queue may indirectly affect the Status Byte's Message Available (MAV) bit).

### **Go To Local (GTL)**

This command returns the analyzer to local (front-panel) control. All keys on the analyzer's front-panel are enabled.

### **Interface Clear (IFC)**

This command causes the analyzer to halt all bus activity. It discontinues any input or output, although the input and output queues are not cleared. If the analyzer is designated as the active controller when this command is received, it relinquishes control of the bus to the system controller. If the analyzer is enabled to respond to a Serial Poll it becomes Serial Poll disabled.

### **Local Lockout (LLO)**

This command causes the analyzer to enter the local lockout mode, regardless of whether it is in the local or remote mode. The analyzer only leaves the local lockout mode when the HP-IB's Remote Enable (REN) line is set FALSE.

Local Lockout ensures that the analyzer's remote softkey menu (including the ~~Return to LOCAL~~ softkey) is disabled when the analyzer is in the remote mode. When the key is enabled, it allows a front-panel operator to return the analyzer to local mode, enabling all other front-panel keys. When the key is disabled, it does not allow the front-panel operator to return the analyzer to local mode.

### **Parallel Poll**

The HP 8711 ignores all of the following parallel poll commands:

- Parallel Poll Configure (PPC).
- Parallel Poll Unconfigure (PPU).
- Parallel Poll Enable (PPE).
- Parallel Poll Disable (PPD).

### **Remote Enable (REN)**

REN is a single line on the HP-IB. When it is set TRUE, the analyzer will enter the remote mode when addressed to listen. It will remain in remote mode until it receives the Go to Local (GTL) command or until the REN line is set FALSE.

When the analyzer is in remote mode and local lockout mode, all front-panel keys are disabled. When the analyzer is in remote mode but not in local lockout mode, all front-panel keys are disabled except for the softkeys. The remote softkey menu includes seven keys that are available for use by a program. The eighth softkey is the **Return to LOCAL** key which allows a front-panel operator to return the analyzer to local mode, enabling all other front-panel keys.

### **Selected Device Clear (SDC)**

The analyzer responds to this command in the same way that it responds to the Device Clear (DCL) command.

When the analyzer receives this command it:

- Clears its input and output queues.
- Resets its command parser (so it is ready to receive a new program message).
- Cancels any pending \*OPC command or query.

The command does not affect:

- Front-panel operation.
- Any analyzer operations in progress (other than those already mentioned).
- Any analyzer settings or registers (although clearing the output queue may indirectly affect the Status Byte's MAV bit).

### **Serial Poll**

The analyzer responds to both of the serial poll commands. The Serial Poll Enable (SPE) command causes the analyzer to enter the serial poll mode. While the analyzer is in this mode, it sends the contents of its Status Byte register to the controller when addressed to talk.

When the Status Byte is returned in response to a serial poll, bit 6 acts as the Request Service (RQS) bit. If the bit is set, it will be cleared after the Status Byte is returned.

The Serial Poll Disable (SPD) command causes the analyzer to leave the serial poll mode.

### **Take Control Talker (TCT)**

If the analyzer is addressed to talk, this command causes it to take control of the HP-IB. It becomes the active controller on the bus. The analyzer automatically passes control back when it completes the operation that required it to take control. Control is passed back to the address specified by the \*PCB command (which should be sent prior to passing control).

If the analyzer does not require control when this command is received, it immediately passes control back.

## Message Exchange

The HP 8711 communicates with the controller and other devices on the HP-IB using program messages and response messages. Program messages are used to send commands, queries, and data to the analyzer.

Response messages are used to return data from the analyzer. The syntax for both kinds of messages is discussed in the *HP-IB Command Reference*.

There are two important things to remember about the message exchanges between the analyzer and other devices on the bus:

- The analyzer only talks after it receives a terminated query (see “Query Response Generation” later in this section).
- Once it receives a terminated query, the analyzer expects to talk before it is told to do something else.

## HP-IB Queues

Queues enhance the exchange of messages between the HP 8711 and other devices on the bus. The analyzer contains:

- An input queue.
- An error queue.
- An output queue.

**Input Queue.** The input queue temporarily stores the following until they are read by the analyzer’s command parser:

- Device commands and queries.
- The HP-IB END message (EOI asserted while the last data byte is on the bus).

The input queue also makes it possible for a controller to send multiple program messages to the analyzer without regard to the amount of time required to parse and execute those messages. The queue holds up to 128 bytes. It is cleared when:

- The analyzer is turned on.
- The Device Clear (DCL) or Selected Device Clear (SDC) command is received.

**Error Queue.** The error queue temporarily stores up to 20 error messages. Each time the analyzer detects an error, it places a message in the queue. When you send the SYST:ERR? query, one message is moved from the error queue to the output queue so it can be read by the controller. Error messages are delivered to the output queue in the order they were received.

The error queue is cleared when:

- All the error messages are read using the SYST:ERR? query.
- The analyzer is turned on.
- The \*CLS command is received.

**Output Queue.** The output queue temporarily stores a single response message until it is read by a controller. It is cleared when:

- The message is read by a controller.
- The analyzer is turned on.
- The Device Clear (DCL) or Selected Device Clear (SDC) command is received.

### Command Parser

The command parser reads program messages from the input queue in the order they were received from the bus. It analyzes the messages to determine what actions the analyzer should take.

One of the parser's most important functions is to determine the position of a program message in the analyzer's command tree (described in the *HP-IB Command Reference*). When the command parser is reset, the next command it receives is expected to arise from the base of the analyzer's command tree.

The parser is reset when:

- The analyzer is turned on.
- The Device Clear (DCL) or Selected Device Clear (SDC) command is received.
- A colon immediately follows a semicolon in a program message. (For more information see "Sending Multiple Commands" in the *HP-IB Command Reference*.)
- A program message terminator is received. A program message terminator can be an ASCII carriage return ( $C_R$ ) or newline character or the HP-IB END message (EOI set true).

### Query Response Generation

When the HP 8711 parses a query, the response to that query is placed in the analyzer's output queue. The response should be read immediately after the query is sent. This ensures that the response is not cleared before it is read. The response is cleared when one of the following message exchange conditions occurs:

- Unterminated condition — the query is not properly terminated with an ASCII carriage return character or the HP-IB END message (EOI set true) before the response is read.
- Interrupted condition — a second program message is sent before the response to the first is read.
- Buffer deadlock — a program message is sent that exceeds the length of the input queue or that generates more response data than fits in the output queue.

---

## Synchronization

The IEEE 488.2 standard provides tools that can be used to synchronize the analyzer and a controller. Proper use of these tools ensures that the analyzer is in a known state when you send a particular command or query.

Device commands can be divided into two broad classes:

- Sequential commands.
- Overlapped commands.

Most of the HP 8711's commands are processed sequentially. A sequential command holds off the processing of subsequent commands until it has been completely processed.

Some commands do not hold off the processing of subsequent commands; they are called overlapped commands.

### Overlapped commands

Typically, overlapped commands take longer to process than sequential commands. For example, the :INITIATE:IMMEDIATE command restarts a measurement. The command is not considered to have been completely processed until the measurement is complete. This can take a long time with a narrow system bandwidth or when averaging is enabled.

The HP 8711 has the following overlapped commands:

```
:ABORt
:CALibration:ZERO:AUTO
:CONFigure[1|2]
:DIAGnostic:CCONstants:LOAD
:DIAGnostic:CCONstants:STORe:DISK
:DIAGnostic:CCONstants:STORe:EEPROM
:DIAGnostic:DITHer
:DIAGnostic:SPUR:AVOID
:HCOPy[:IMMediate]
:INITiate[1|2]:CONTinuous
:INITiate[1|2][:IMMediate]
:MMEMory:LOAD:STATE
:OUTPut[:STATE]
:PROGram[:SElected]:EXECute
:SENSe[1|2]:AVERage:CLEAR
:SENSe[1|2]:AVERage:COUNT
:SENSe[1|2]:AVERage[:STATE]
:SENSe[1|2]:BWIDth[:RESolution]
:SENSe[1|2]:CORRection:COLlect[:ACQuire]
:SENSe[1|2]:CORRection:COLlect:ISTate[:AUTO]
:SENSe[1|2]:CORRection:COLlect:METHod
:SENSe[1|2]:CORRection:COLlect:SAVE
:SENSe[1|2]:CORRection:CSET[:SElect]
```

```

:SENSe[1|2]:CORRection[:STATe]
:SENSe:COUPLE
:SENSe[1|2]:DETEctor[:FUNCTION]
:SENSe[1|2]:FREQuency:CENTER
:SENSe[1|2]:FREQuency:SPAN
:SENSe[1|2]:FREQuency:STARt
:SENSe[1|2]:FREQuency:STOP
:SENSe[1|2]:FUNCTION
:SENSe:ROSCillator:SOURce
:SENSe[1|2]:STATE
:SENSe[1|2]:SWEep:POINTs
:SENSe[1|2]:SWEep:TIME
:SENSe[1|2]:SWEep:TIME:AUTO
:SENSe:SWEep:TRIGger:SOURce
:SOURCE[1|2]:POWER[:LEVel][:IMMEDIATE][:AMPLitude]
:SYSTEM:PRESet
:TRACE[:DATA]
:TRIGger[:SEQUence]:SOURce

```

The analyzer uses a No Pending Operation (NPO) flag to keep track of overlapped commands. The NPO flag is reset to 0 when an overlapped command has not completed (still pending). It is set to 1 when no overlapped commands are pending. The NPO flag cannot be read directly but all of the following common commands take some action based on the setting of the flag.

- **\*WAI** — Holds off the processing of subsequent commands until the NPO flag is set to 1. This ensures that commands in the analyzer's input queue are processed in the order received.

The program continues to run and additional commands are received and parsed by the analyzer (but not executed) while waiting for the NPO flag to be set. Use of the \*WAI command is demonstrated in the SETUP example program.

- **\*OPC?** — Places a 1 in the analyzer's output queue when the NPO flag is set to 1. If the program is designed to read the output queue before it continues, this effectively pauses the controller until all pending overlapped commands are completed. Use of the \*OPC? command is demonstrated in the TRANCAL and REFLCAL example programs.
- **\*OPC** — Sets bit 0 of the Standard Event Status event register to 1 when the NPO flag is set to 1. The analyzer's status registers can then be used to generate a service request when all pending overlapped commands are completed. This synchronizes the controller to the completion of an overlapped command, but also leaves the controller free to perform other tasks while the command is executing.

---

**Note**



\*OPC only informs you when the NPO flag is set to 1. It does not hold off the processing of subsequent commands. No commands should be sent to the analyzer between sending the \*OPC command and receiving the service request. Any command sent will be executed and may affect how the instrument responds to the previously sent \*OPC.

---

The \*CLS and \*RST commands cancel any preceding \*OPC command or query. Pending overlapped commands are still completed, but their completion is not reported in either the status register or the output queue. Two HP-IB bus management commands — Device Clear (DCL) and Selected Device Clear (SDC) — also cancel any preceding \*OPC command or query.

---

**Note**  Use \*WAI, \*OPC? or \*OPC whenever overlapped commands are used. A recommended technique is to send \*WAI at the end of each group of commands.

---

**Caution**  **ALWAYS** trigger an individual sweep (using \*OPC? and waiting for the reply) before reading data over the bus or executing a marker function. The analyzer has the ability to process the commands it receives faster than it can make a measurement. If the measurement is not complete when the data is read or a marker search function is executed the results are invalid.

The command to use (in an IBASIC OUTPUT statement) is:

```
OUTPUT @Hp8711;"ABOR;:INIT:CONT OFF;:INIT;*OPC?  
ENTER @Hp8711;Opc_done
```

or another form of the :INITiate[1|2] [:IMMEDIATE] command combined with the \*OPC? query.

---

---

## Passing Control

The HP 8711 requires temporary control of the HP-IB to complete some commands (such as print/plot or external disk operations). After sending such a command, the active controller must pass control to the analyzer. When the analyzer completes the command, it automatically passes control of the bus back to the controller.

For smooth passing of control, take steps that ensure the following conditions are met:

- The analyzer must know the controller's address so it can pass control back.
- The controller must be informed when the analyzer passes control back.

The following is a procedure for passing control:

1. Send the controller's HP-IB address to the analyzer with the \*PCB command.
2. Clear the analyzer's status registers with the \*CLS command.
3. Enable the analyzer's status registers to generate a service request when the Operation Complete bit is set. (Send \*ESE with a value of 1 and \*SRE with a value of 32.)
4. Enable the controller to respond to the service request.
5. Send the command that requires control of the bus followed by the \*OPC command.
6. Pass control to the analyzer and wait for the service request. The service request indicates that the command has been completed and control has been passed back to the controller.

---

**Note**

For this procedure to work properly, only the command that requires control of the bus should be pending. Other overlapped commands should not. For more information on overlapped commands, see "Synchronization" in this chapter.

---

An example program, PASSCTRL, demonstrates passing control to the analyzer. In this example program control is passed so the analyzer can control a printer for hardcopy output.

---

## Transferring Data

Data is transferred between the HP 8711 and a controller via the HP-IB data lines, DIO1 through DIO8. Such transfers occur in a byte-serial (one byte at a time), bit-parallel (8 bits at a time) manner. This section discusses the following aspects of data transfer:

- The different data types used during data transfers.
- Data encoding used during transfers of numeric block data.

### Data Types

The HP 8711 uses a number of different data types during data transfers. Data transfer occurs in response to a query. The data type used is determined by the parameter being queried. The different parameter types are described in the "Parameter Types" section of the *HP-IB Command Reference*. Data types described in this section are:

- Numeric Data.
- Character Data
- String Data
- Expression Data
- Block Data

### Numeric Data

The analyzer returns three types of numeric data in response to queries:

- NR1 data — Integers (such as +1, 0, -1, 123, -12345). This is the response type for boolean parameters as well as some numeric parameters.
- NR2 data — Floating point numbers with an explicit decimal point (such as 12.3, +1.234, -0.12345).
- NR3 data — Floating point numbers in scientific notation (such as +1.23E+5, +123.4E-3, -456.789E+6).

### Character Data

Character data consists of ASCII characters grouped together in mnemonics that represent specific instrument settings (such as MAXimum, MINimum or MLOGarithmic). The analyzer always returns the short form of the mnemonic in upper-case alpha characters.

### String Data

String data consists of ASCII characters. The string must be enclosed by a delimiter, either single quotes ('This is string data.') or double quotes ("This is also string data."). To include the delimiter as a character in the string it must be typed twice without any characters in between. The analyzer always uses double quotes when it returns string data.

## **Expression Data**

Expression data consists of mathematical expressions that use character parameters. When expression data is sent to the analyzer it is always enclosed in parentheses (such as (IMPL/CH1\$MEM) or (IMPL)). The analyzer returns expression data enclosed in double quotes.

## **Block Data**

Block data are typically used to transfer large quantities of related data (like a data trace). Blocks can be sent as definite length blocks or indefinite length blocks — the instrument will accept either form. The analyzer always returns definite length block data in response to queries.

**Definite Block Length.** The general form for a definite block length transfer is:

```
#<num_digits><num_bytes><data_bytes>
```

In the definite length block, two numbers must be specified. The single decimal digit <num\_digits> specifies how many digits are contained in <num\_bytes>. The decimal number <num\_bytes> specifies how many data bytes will follow in <data\_bytes>. An example IBASIC (or HP BASIC) statement to send ABC+XYZ as a definite block length parameter is shown, note that the data block contains seven bytes (7) and only one digit is needed to describe the block length 1.

```
OUTPUT 716;"#17ABC+XYZ"
```

**Indefinite Block Length.** The general form for an indefinite block length transfer is:

```
#0<data_bytes><C_R><EOI>
```

Note that a mandatory  $C_R$  (carriage return or newline) EOI sequence immediately follows the last byte of block data in an indefinite length block. This forces the termination of the program message. An example IBASIC (or HP BASIC) statement to send ABC+XYZ as an indefinite block length parameter is shown, note that ,END is used to properly terminate the message.

```
OUTPUT 716;"#0ABC+XYZ",END
```

## **Data Encoding for Large Data Transfers**

The **FORMAT:DATA** command selects the type of data and the type of data encoding that is used to transfer large blocks of numeric data between the analyzer and a controller. There are two specifiers:

- **REAL** specifies the block data type. Either the definite or indefinite length syntax can be used. The block is transferred as a series of binary-encoded floating-point numbers. Data transfers of the **REAL,64** data type are demonstrated in the **REALDATA** example program.
- **INTEGER** specifies the block data type. Either the definite or indefinite length syntax can be used. The block is transferred as an array of binary-encoded data with each point represented by a set of three 16-bit integers. This is the instrument's internal format — it

should only be used for data that will be returned to the instrument for later use. Data transfers of the INTeger,16 data type are demonstrated in the INTDATA and LOADCALS example programs.

- ASCII specifies the numeric data type (NR1, NR2 or NR3 syntax). The data is transferred as a series of ASCII-encoded numbers separated by commas. ASCII formatted data transfers are demonstrated in the ASCDATA example program.

Blocks that contain mixed data — both numbers and ASCII characters — ignore the setting of FORMat:DATA. These blocks always transfer as either definite length or indefinite length block data. The following commands transfer blocks of mixed data:

```
PROGram[:SELected]:DEFine  
SYSTem:SET
```

### ASCII Encoding

The ANSI X3.4-1977 standard defines the ASCII 7-bit code. When an ASCII-encoded byte is sent over the HP-IB, bits 0 through 6 of the byte (bit 0 being the least significant bit) correspond to the HP-IB data lines DIO1 through DIO7. DIO8 is ignored.

When ASCII encoding is used for large blocks of data, the number of significant digits to be returned for each number in the block can be specified. For example, the following command returns all numbers as NR3 data with 7 significant digits.

```
FORMat:DATA ASCII,7
```

### Binary Encoding

When binary encoding is used for large blocks of data, all numbers in the block are transferred as 32-bit or 64-bit binary floating point numbers or as an array of 16-bit integers. The binary floating-point formats are defined in the IEEE 754-1985 standard.

FORMat:DATA REAL,32 selects the IEEE 32-bit format (not supported by IBASIC or HP BASIC).

FORMat:DATA REAL,64 selects the IEEE 64-bit format.

FORMat:DATA INTeger,16 selects the 16-bit integer format.

### Byte Swapping

PC compatibles frequently use a modification of the IEEE floating point formats with the byte order reversed. To reverse the byte order for data transfer into a PC, the FORMat:BORDer command should be used.

FORMat:BORDer SWAPPed selects the byte-swapped format.

FORMat:BORDer:NORMAl selects the standard format.

## Using the Status Registers

The HP 8711's status registers contain information about the condition of the network analyzer and its measurements. This section describes the registers and their use in HP-IB programming.

Example programs using the status registers are included in the "Example Programs" section later in this chapter. These programs include AVERAGE and GRAPHICS which use of service request interrupt routines, PASSCTRL which uses the status byte to request control of the HP-IB and LIMITEST which uses the Limit Fail condition register.

### General Status Register Model

The HP 8711's status system is based on the general status register model shown in Figure 9a-1. Most of the analyzer's register sets include all of the registers shown in the model, although commands are not always available for reading or writing a particular register. The information flow within a register set starts at the condition register and ends at the register summary bit (see Figure 9a-2). This flow is controlled by setting bits in the transition and enable registers.

Two register sets — the Status Byte and the Standard Event Status Register — are 8-bits wide. All others are 16-bits wide, but the most significant bit (bit 15) in the larger registers is always set to 0.

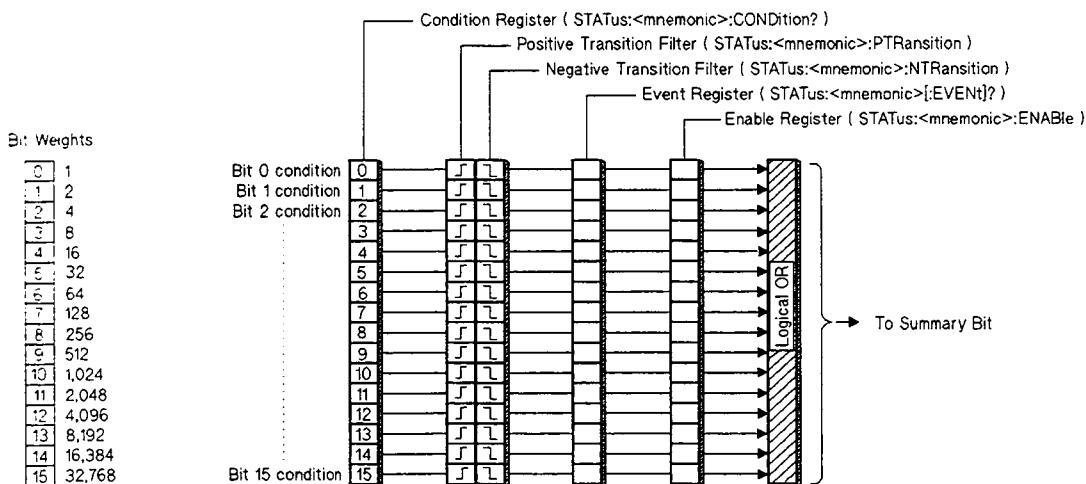


Figure 9a-1. General Status Register Model

### **Condition Register**

Condition registers continuously monitor the instrument's hardware and firmware status. Bits in a condition register are not latched or buffered, they are updated in real time. When the condition monitored by a specific bit becomes true, the bit is set to 1. When the condition becomes false the bit is reset to 0. Condition registers are read-only.

### **Transition Registers**

Transition registers control what type of change in a condition register will set the corresponding bit in the event register. Positive state transitions (0 to 1) are only reported to the event register if the corresponding positive transition bit is set to 1. Negative state transitions (1 to 0) are only reported if the corresponding negative transition bit is set to 1. Setting both transition bits to 1 causes both positive and negative changes to be reported. Transition registers are read-write, and are unaffected by \*CLS (clear status) or queries. They are reset to instrument default conditions at power up and after \*RST and SYSTem:PRESet commands.

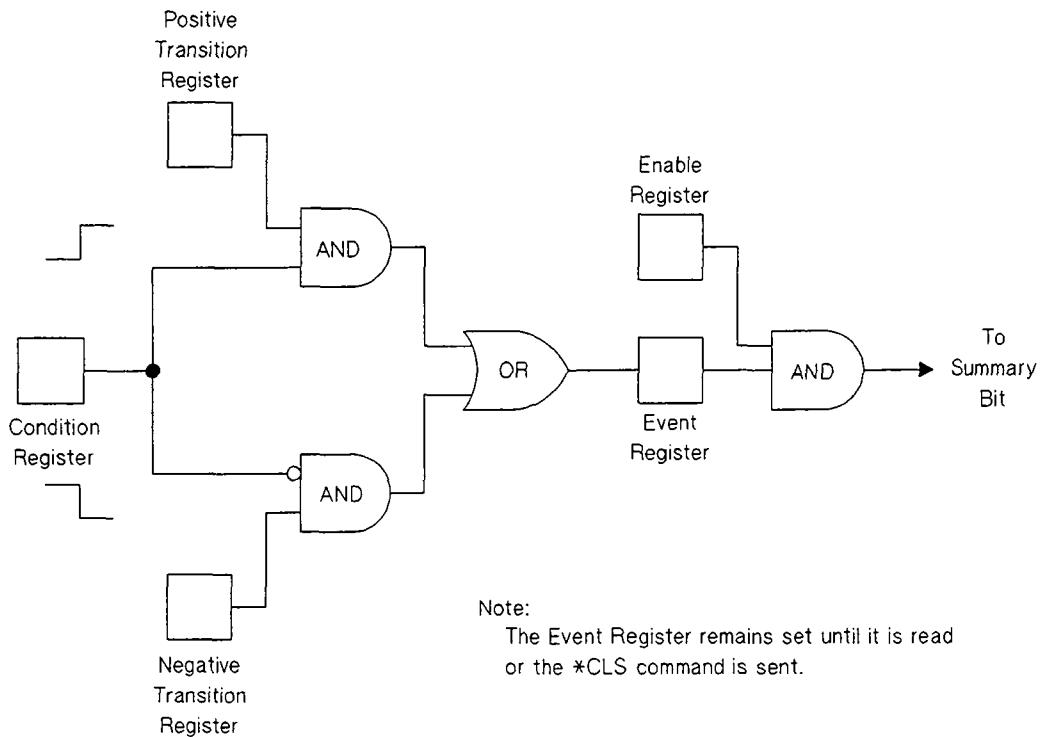
### **Event Register**

Event registers latch any reported condition changes. When a transition bit allows a condition change to be reported, the corresponding event bit is set to 1. Once set, an event bit is no longer affected by condition changes. It remains set until the event register is cleared. Event registers are read-only.

An event register is cleared when you read it. All event registers are cleared when you send the \*CLS (clear status) command.

### **Enable Register**

Enable registers control the reporting of events (latched conditions) to the register summary bit. If an enable bit is set to 1 the corresponding event is included in the logical ORing process that determines the state of the summary bit. (The summary bit is only set to 1 if one or more enabled event bits are set to 1.) Summary bits are recorded in the instrument's status byte. Enable registers are read-write and are cleared by \*CLS (clear status).



**Figure 9a-2. Flow of information within a register set**

## How to Use Registers

There are two methods of accessing the information in status registers:

- The direct-read method.
- The service request (SRQ) method.

In the direct-read method the analyzer is passive. It only tells the controller that conditions have changed when the controller asks the right question. In the SRQ method, the analyzer is more active. It tells the controller when there has been a condition change without the controller asking. Either method allows you to monitor one or more conditions.

The following steps are used to monitor a condition with the direct read method:

1. Determine which register contains the bit that monitors the condition.
2. Send the unique HP-IB query that reads that register.
3. Examine the bit to see if the condition has changed.

The direct-read method works well when it is not necessary to know about changes the moment they occur. It does not work well if immediate knowledge of the condition change is needed. A program that used this method to detect a change in a condition would need to continuously read the registers at very short intervals. The SRQ method is better suited for that type of need.

## The Service Request Process

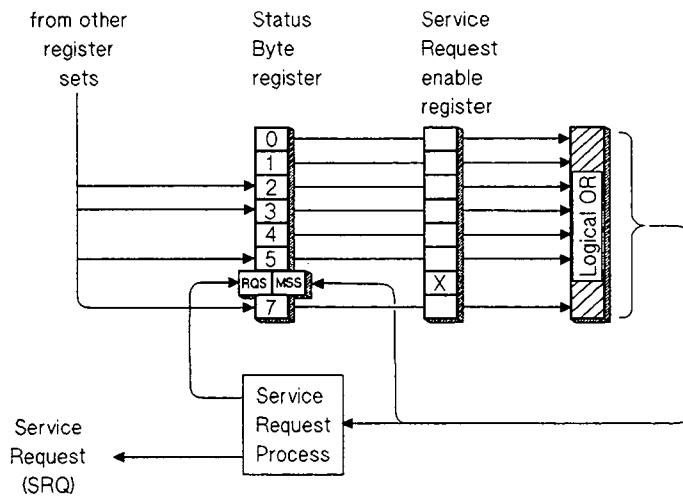
The following steps are used to monitor a condition with the SRQ method:

1. Determine which bit monitors the condition.
2. Determine how that bit reports to the request service (RQS) bit of the Status Byte.
3. Send HP-IB commands to enable the bit that monitors the condition and to enable the summary bits that report the condition to the RQS bit.
4. Enable the controller to respond to service requests.

When the condition changes, the analyzer sets its RQS bit and the HP-IB's SRQ line. The controller is informed of the change as soon as it occurs. The time the controller would otherwise have used to monitor the condition can now be used to perform other tasks. The controller's response to the SRQ is determined by the program being run.

## Generating a Service Request

A service request is generated using the Status Byte. As shown in Figure 9a-3, the HP 8711's other register sets report to the Status Byte. Some of them report directly while others report indirectly through other register sets.



**Figure 9a-3. Generating a Service Request**

The process of preparing the analyzer to generate a service request, and the handling of that interrupt when it is received by a program, are demonstrated in the AVERAGE example program.

When a register set causes its summary bit in the Status Byte to change from 0 to 1, the analyzer can initiate the service request (SRQ) process. If both the following conditions are true the process is initiated:

- The corresponding bit of the Service Request enable register is also set to 1.
- The analyzer does not have a service request pending. (A service request is considered to be pending between the time the analyzer's SRQ process is initiated and the time the controller reads the Status Byte register with a serial poll).

The SRQ process sets the HP-IB's SRQ line true and sets the Status Byte's request service (RQS) bit to 1. Both actions are necessary to inform the controller that the HP 8711 requires service. Setting the SRQ line informs the controller that some device on the bus requires service. Setting the RQS bit allows the controller to determine that the HP 8711 was the device that initiated the request.

When a program enables a controller to detect and respond to service requests, it should instruct the controller to perform a serial poll when the HP-IB's SRQ line is set true. Each device on the bus returns the contents of its Status Byte register in response to this poll. The device whose RQS bit is set to 1 is the device that requested service.

---

**Note** When the analyzer's Status Byte is read with a serial poll, the RQS bit is reset to 0. Other bits in the register are not affected.



As implied in Figure 9a-3, bit 6 of the Status Byte register serves two functions; the request service function (RQS) and the master summary status function (MSS). Two different methods for reading the register allow you to access the two functions. Reading the register with a serial poll allows you to access the bit's RQS function. Reading the register with \*STB allows you to access the bit's MSS function.

## The HP 8711's Status Register Sets

The HP 8711 uses eight register sets to keep track of instrument status:

- |                         |                            |
|-------------------------|----------------------------|
| • Status Byte           | *STB? and *SRE             |
| • Device Status         | STATus:DEvice              |
| • Limit Fail            | STATus:QUESTIONable:LIMit  |
| • Questionable Status   | STATus:QUESTIONable        |
| • Standard Event Status | *ESR? and *ESE             |
| • Measuring Status      | STATus:OPERation:MEASuring |
| • Averaging Status      | STATus:OPERation:AVERaging |
| • Operational Status    | STATus:OPERation           |

Their reporting structure is summarized in Figure 9a-4. They are described in greater detail in the following section.

**Note** Register bits not explicitly presented in the following sections are not used by the HP 8711. A query to one of these bits returns a value of 0.

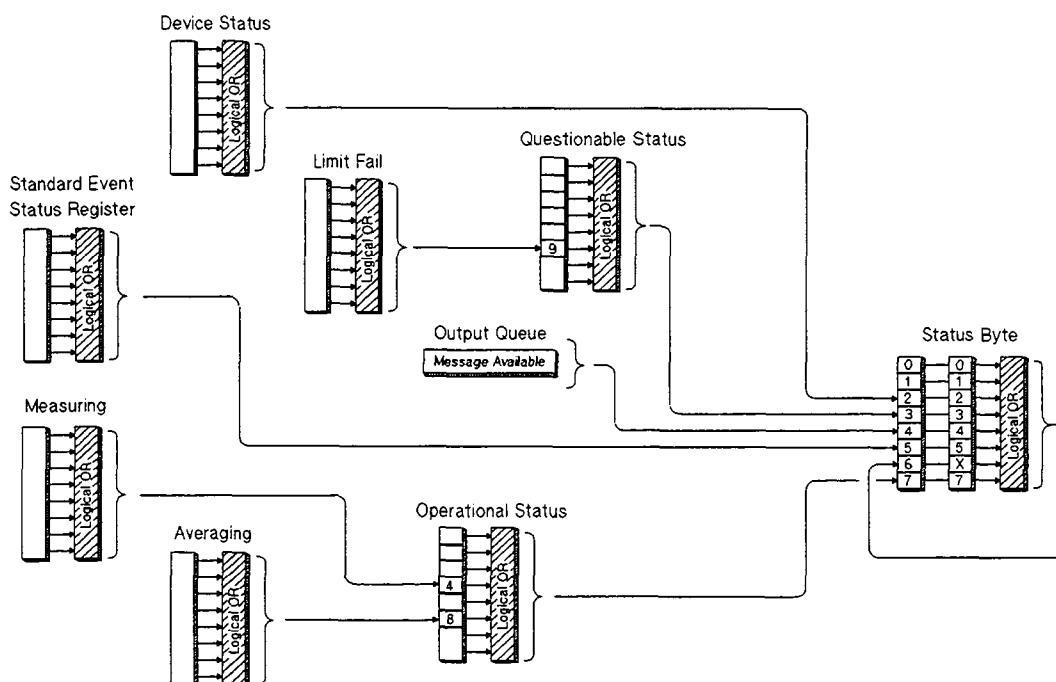
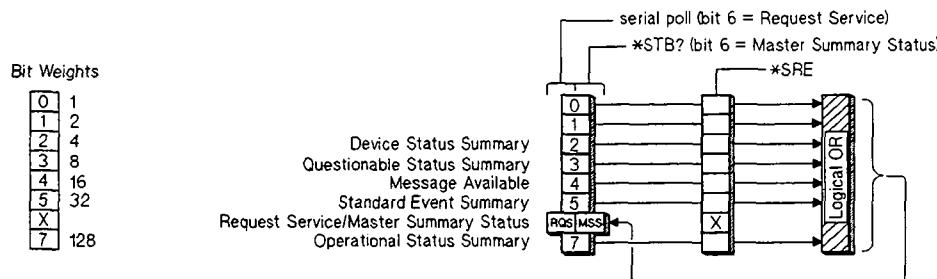


Figure 9a-4. HP 8711 Register Sets

## Status Byte

The Status Byte register set summarizes the states of the other register sets and monitors the analyzer's output queue. It is also responsible for generating service requests (see "Generating Service Requests" earlier in this chapter). See Figure 9a-5



**Figure 9a-5. The Status Byte Register Set**

The Status Byte register set does not conform to the general status register model described at the beginning of this chapter. It contains only two registers: the Status Byte register and the Service Request enable register. The Status Byte register behaves like a condition register for all bits except bit 6. The Service Request enable register behaves like a standard enable register except that bit 6 is always set to 0.

Bits in the Status Byte register are set to 1 under the following conditions:

- **Device Status Summary** (bit 2) is set to 1 when one or more enabled bits in the Device Status event register are set to 1.
- **Questionable Status Summary** (bit 3) is set to 1 when one or more enabled bits in the Questionable Status event register are set to 1.
- **Message Available** (bit 4) is set to 1 when the output queue contains a response message.
- **Standard Event Status Summary** (bit 5) is set to 1 when one or more enabled bits in the Standard Event Status event register are set to 1.
- **Master Summary Status** (bit 6, when read by \*STB) is set to 1 when one or more enabled bits in the Status Byte register are set to 1.
- **Request Service** (bit 6, when read by serial poll) is set to 1 by the service request process (see "Generating a Service Request" earlier in this chapter).
- **Operational Status Summary** (bit 7) is set to 1 when one or more enabled bits in the Operational Status event register are set to 1.

The commands used to read and write the Status Byte registers are listed below:

- **SPOLL** — an IBASIC (or HP BASIC) command used in the service request process to determine which device on the bus is requesting service.
- **\*STB?** reads the value of the instrument's status byte. This is a non-destructive read, the Status Byte is cleared by the **\*CLS** command.
- **\*SRE <num>** sets bits in the Service Request Enable register. The current setting of the Service Request Enable register is stored in non-volatile memory.
- **\*SRE?** reads the current state of the Service Request Enable register.

### **Device Status Register Set**

The Device Status register set monitors the state of device-specific parameters.

Bits in the Device Status condition register are set to 1 under the following conditions:

- **Key Pressed** (bit 0) is set to 1 when one of the HP 8711's front panel keys has been pressed.

### **Limit Fail Register Set**

The Limit Fail register set monitors limit test results for both measurement channels.

Bits in the Limit Fail condition register are set to 1 under the following conditions:

- **Channel 1 Limit Failed** (bit 0) is set to 1 when limit testing is enabled and any point on channel 1 fails the limit test.
- **Channel 2 Limit Failed** (bit 1) is set to 1 when limit testing is enabled and any point on channel 2 fails the limit test.

### **Questionable Status Register Set**

The Questionable Status register set monitors conditions that affect the quality of measurement data.

Bits in the Questionable Status condition register are set to 1 under the following conditions:

- **Limit Fail** (bit 9) is set to 1 when one or more enabled bits in the Limit Fail event register are set to 1.
- **Data Questionable** (bit 10) is set to 1 when a change in the analyzer's configuration requires that new measurement data be taken.

## Standard Event Status Register Set

The Standard Event Status register set monitors HP-IB errors and synchronization conditions. See Figure 9a-6

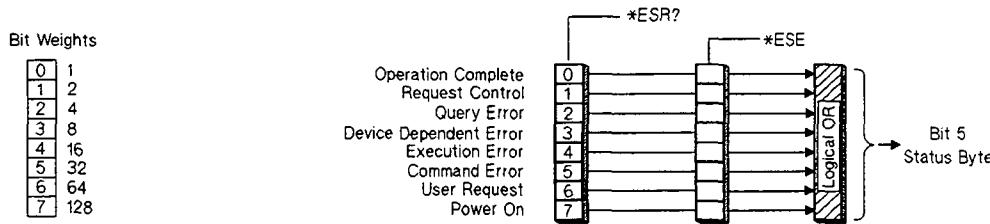


Figure 9a-6. The Standard Event Status Register Set

The Standard Event Status register set does not conform to the general status register model described at the beginning of this section. It contains only two registers: the Standard Event Status event register and the Standard Event Status enable register. The Standard Event Status event register is similar to other event registers, but behaves like a register set that has a positive transition register with all bits set to 1. The Standard Event Status enable register is the same as other enable registers.

- **Operation Complete** (bit 0) is set to one when the following two events occur (in the order listed):
  - The \*OPC command is sent to the analyzer.
  - The analyzer completes all pending overlapped commands.
- **Request Control** (bit 1) is set to 1 when both of the following conditions are true:
  - The analyzer is configured as a talker/listener for HP-IB operation.
  - The analyzer is instructed to do something (such as plotting or printing) that requires it to take control of the bus.
- **Query Error** (bit 2) is set when the command parser detects a query error.
- **Device Dependent Error** (bit 3) is set to 1 when the command parser detects a device-dependent error.
- **Execution Error** (bit 4) is set to 1 when the command parser detects an execution error.
- **Command Error** (bit 5) is set to 1 when the command parser detects a command error.
- **Power On** (bit 7) is set to 1 when you turn on the analyzer.

The commands used to read and write the Standard Event Status registers are listed below:

- \*ESR? reads the value of the standard event status register.
- \*ESE <num> sets bits in the standard event status enable register. The current setting of the standard event status enable register is stored in non-volatile memory.
- \*ESE? reads the current state of the standard event status enable register.

### **Measuring Status Register Set**

The Measuring Status register set monitors conditions in the analyzer's measurement process.

Bits in the Measuring Status condition register are set to 1 under the following conditions:

- **Channel 1 Measuring** (bit 0) is set to 1 while the analyzer is collecting measurement data on channel 1.
- **Channel 2 Measuring** (bit 1) is set to 1 while the analyzer is collecting measurement data on channel 2.

### **Averaging Status Register Set**

The Averaging Status register set monitors conditions in the analyzer's measurement process when the trace averaging function is in use.

Bits in the Averaging Status condition register are set to 1 under the following conditions:

- **Channel 1 Averaging** (bit 0) is set to 1 while the analyzer is sweeping on channel 1 and the number of sweeps completed (since "average restart") is less than the averaging factor.
- **Channel 2 Averaging** (bit 1) is set to 1 while the analyzer is sweeping on channel 2 and the number of sweeps completed (since "average restart") is less than the averaging factor.

### **Operational Status Register Set**

The Operation Status register set monitors conditions in the analyzer's measurement process, disk operations, and printing/plotting operations. It also monitors the state of the current HP Instrument BASIC program.

Bits in the Operational Status condition register are set to 1 under the following conditions:

- **Calibrating** (bit 0) is set to 1 while the instrument is zeroing the broadband diode detectors.
- **Settling** (bit 1) is set to 1 while the measurement hardware is settling.
- **Measuring** (bit 4) is set to 1 when one or more enabled bits in the Measuring Status event register are set to 1.
- **Correcting** (bit 7) is set to 1 while the analyzer is performing a calibration function.

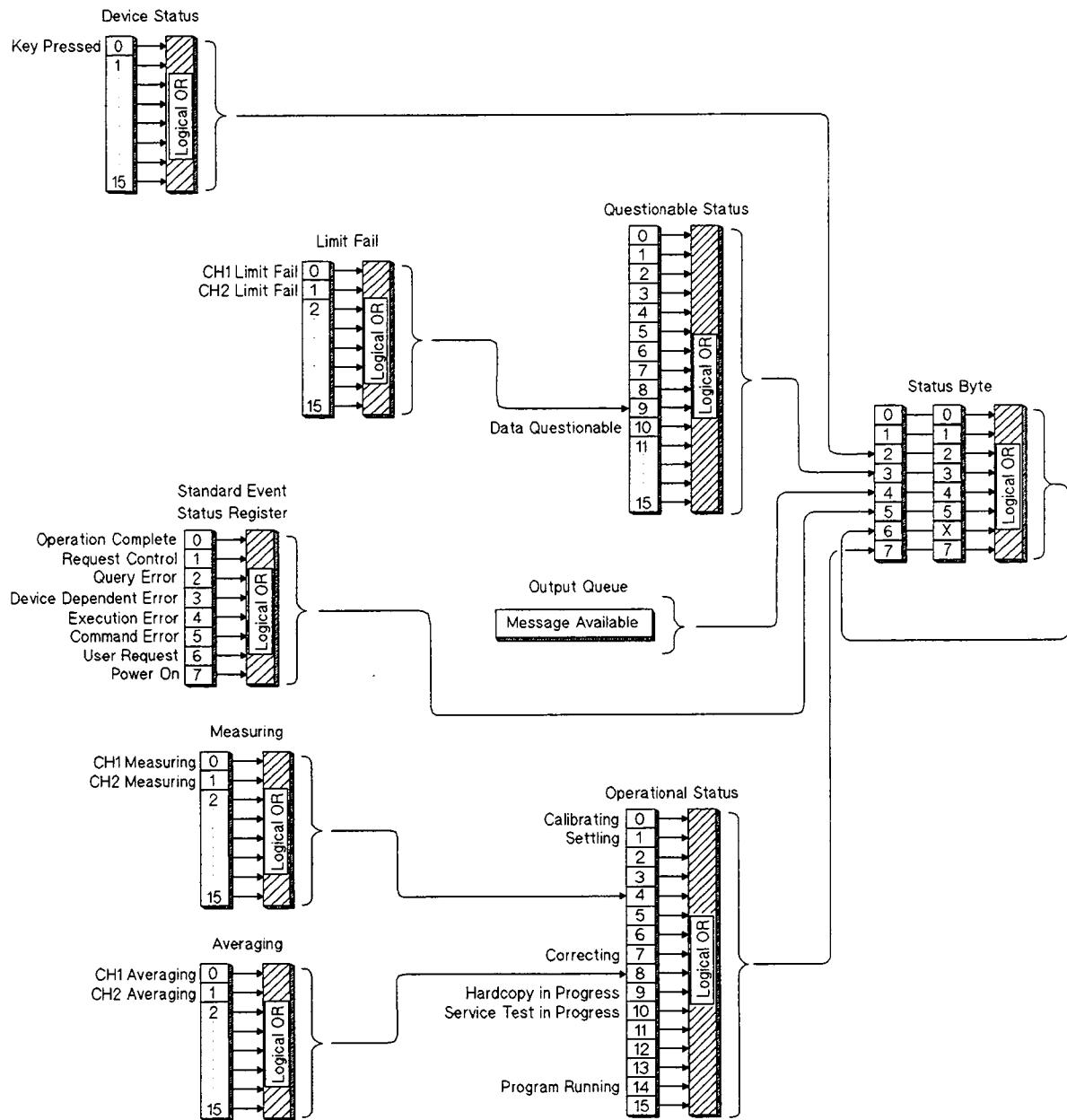
- **Averaging** (bit 8) is set to 1 when one or more enabled bits in the Averaging Status event register are set to 1.
- **Hardcopy Running** (bit 9) is set to 1 while the analyzer is performing a hardcopy (print or plot) function.
- **Test Running** (bit 10) is set to 1 when one of the analyzer's internal service tests is being run.
- **Program Running** (bit 14) is set to 1 while an HP Instrument BASIC program is running on the analyzer's internal controller.

### **STATus:PRESet Settings**

Executing the **STATUS:PRESet** command changes the settings in the enable (**ENAB**), positive transition (**PTR**) and negative transition (**NTR**) registers. The table below shows the settings after the command is executed.

Register Set	ENABLE	PTRansition	NTRansition
STATus:DEVice	all 0s	all 1s	all 0s
STATus:QUESTIONable:LIMit	all 1s	all 1s	all 0s
STATus:QUESTIONable	all 0s	all 1s	all 0s
STATus:OPERation:MEASuring	all 1s	all 0s	all 1s
STATus:OPERation:AVERaging	all 1s	all 0s	all 1s
STATus:OPERation	all 0s	all 1s	all 0s

## HP 8711 Register Set Summary



---

## User Graphics

The HP 8711 has a set of user graphics commands that can be used to create graphics and messages on the display. The GRAPHICS example program uses some of these commands to draw a simple setup diagram. These commands, listed below, are of the form:

DISPLAY:WINDOW[1|2|10]:GRAPHICS:<mnemonic>.

The number specified in the WINDOW part of the command selects where the graphics are to be written.

- WINDOW1 draws the graphics to the channel 1 measurement screen. (This is the default if no window is specified in the mnemonic.)
- WINDOW2 draws the graphics to the channel 2 measurement screen.
- WINDOW10 draws the graphics to an IBASIC display partition. (This window is only available on instruments with IBASIC — option 1C2.)

---

### Note



When graphics commands are used to write directly to a measurement screen they write to the static graphics plane (the same plane where the graticule is drawn). There is no sweep-to-sweep speed penalty once the graphics have been drawn.

---

Unless otherwise specified, the graphics commands listed below start at the current pen location. All sizes are dimensioned in pixels.

```
DISPLAY:WINDOW[1|2|10]:GRAPHICS:CIRCLE <diameter>
DISPLAY:WINDOW[1|2|10]:GRAPHICS:CLEar
DISPLAY:WINDOW[1|2|10]:GRAPHICS:COLor <pen>
    color choices are: 0 for erase, 1 for bright, 2 for dim
DISPLAY:WINDOW[1|2|10]:GRAPHICS[:DRAW] <new_x>,<new_y>
DISPLAY:WINDOW[1|2|10]:GRAPHICS:LABEL <string>
DISPLAY:WINDOW[1|2|10]:GRAPHICS:LABEL:FONT <font>
    font choices are: SMALL, HSMall, NORMal, HNORMal, BOLD, HBOLD, SLANT, HSLant
    (H as the first letter of the font name indicates highlighted text - inverse video).
DISPLAY:WINDOW[1|2|10]:GRAPHICS:MOVE <new_x>,<new_y>
DISPLAY:WINDOW[1|2|10]:GRAPHICS:RECTangle <width>,<height>
DISPLAY:WINDOW[1|2|10]:GRAPHICS:STATE?
```

## Window Geometry

Even though there are only three graphics windows, these windows can have different sizes and locations.

The size and location of the graphics window are determined by the display configuration currently in use — split screen measurements, full screen measurements, and full or partial IBASIC display partitions will affect the dimensions of the graphics window in use.

The sizes of the different graphics windows are listed below.

- Channel 1 or channel 2 full screen measurement: width=801 pixels, height=321 pixels.
- Channel 1 or channel 2 split screen measurement: width=801 pixels, height=161 pixels.
- IBASIC full screen display: width=861 pixels, height=352 pixels.
- IBASIC upper display: width=861 pixels, height=160 pixels.
- IBASIC lower display: width=861 pixels, height=158 pixels.

There is a set of queries that can be used to determine the size and location of the display window in use.

These queries, listed below, return the width and height of the window or the absolute location of its lower left or upper right corners. All the coordinates and sizes are dimensioned in pixels.

```
DISPlay:WINDOW[1|2|10]:GEOmetry:LLEFT?  
DISPlay:WINDOW[1|2|10]:GEOmetry:SIZE?  
DISPlay:WINDOW[1|2|10]:GEOmetry:URIGHT?
```

---

### Note



The origin of EVERY graphics window is its lower left corner. The locations returned in response to the LLEFT and URIGht are relative to the ABSOLUTE origin of the entire display, NOT to the graphics window.

---

## The Graphics Buffer

The analyzer has a graphics buffer that is used to refresh the graphics display if needed. When the buffer is full, additional graphics can still be drawn — BUT they will not be refreshed. The graphics buffer can be turned on and off using the following command (which is used in the GRAPHICS example program).

```
DISPlay:WINDOW:GRAphics:BUFFER[:STATE] <ON|OFF>
```

The graphics buffer will hold up to:

500 lines  
40 circles  
40 rectangles  
50 strings (60 characters long)

## Front Panel Keycodes

Each key on the front panel of the HP 8711 produces a “key code” when it is pressed. It is possible to trap key presses using the Device Status register. Demonstrations of this capability are included in the REFLCAL and GRAPHICS example programs. Information about the key press can be obtained using the following commands.

```
SYSTem:KEY:QUEue:CLEAR
SYSTem:KEY:COUNT?
SYSTem:KEY:QUEue:MAXimum?
SYSTem:KEY:QUEue[:STATe]
SYSTem:KEY:QUEue[:STATe]?
```

SYSTem:KEY:TYPE?

returns: NONE, RPG (knob), KEY (front panel), ASC (external keyboard)

SYSTem:KEY[:VALue]?

returns: knob count (RPG), key code (KEY), ASCII code (ASC)

Key Group	Key Label	Key Code	Key Group	Key Label	Key Code
Softkeys	Softkey 1 (top key) Softkey 2 Softkey 3 Softkey 4 Softkey 5 Softkey 6 Softkey 7 Softkey 8 (bottom key)	0 1 2 3 4 5 6 7	Numeric Keys	— / ← (minus/backspace) ↑ (step up) ↓ (step down)	22 23 24
			Feature Keys	BEGIN CHAN 1 CHAN 2 POWER MENU FREQ SWEEP CAL DISPLAY SCALE AVG FORMAT MARKER SAVE/RECALL SYSTEM/OPTIONS HARD/COPY PRESET	40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
Numeric Keys	0 (zero) 1 (one) 2 (two) 3 (three) 4 (four) 5 (five) 6 (six) 7 (seven) 8 (eight) 9 (nine) ENTER . (decimal)	10 11 12 13 14 15 16 17 18 19 20 21			

---

## Example Programs

The example programs listed in this manual are all written in IBASIC (HP Instrument BASIC). An optional internal controller can be purchased with an HP 8711 RF Network Analyzer (option 1C2). This controller runs IBASIC directly on the analyzer. It controls the analyzer over an internal interface bus that operates the same way as the external HP-IB interface. For more information about IBASIC refer to the *HP Instrument BASIC User's Handbook* or *Using HP Instrument BASIC with the HP 8711*.

IBASIC is a programming language that developed from HP BASIC. Because of this relationship, programs written for IBASIC can be run on external controllers that run HP BASIC. In addition to running on external workstations, the same programs can be run on PC-compatibles using IBASIC for Windows.

The example programs are provided on two disks that are included with the *HP 8711 Operating Manual*. Both disks contain the same examples written in IBASIC, only the disk format is different. Because the HP 8711's internal floppy disk drive is designed to be both DOS and LIF compatible, both disks can be used to supply programs for the analyzer's internal IBASIC controller.

- *Example Programs Disk -DOS Format* has programs that can be run on the instrument's internal controller or on a PC-compatible running IBASIC for Windows.
- *Example Programs Disk - LIF Format* has programs can be run on the instrument's internal controller or on an HP series 300 workstation running HP BASIC (version 3.0 or higher).

Following is a list of all the example programs included on both disks.

**SETUP — Setting up a basic measurement.** The example also demonstrates the use of the \*WAI command.

**MARKERS — Transferring data using markers.** The example also demonstrates the use of the query form of command mnemonics.

**LIMITEST — Performing automatic PASS/FAIL testing with limit lines.** The example also demonstrates some methods of combining mnemonics for more efficient programming.

**ASCDATA — Transferring data using the ASCII format.** The section also lists the different data arrays (and their mnemonics) that can be accessed.

**REALDATA — Transferring data using the IEEE 64-bit floating point REAL format.** The example also demonstrates block data transfers of both indefinite length and definite length syntax. Also demonstrated is access to the swapped-byte data format designed for PCs.

**INTDATA — Transferring data using the 16-bit INTEGER format.**

**TRANCAL — Performing a transmission calibration.** The calibration is User Defined (performed over the instruments current source settings). This example also demonstrates the use of the \*OPC? command.

**REFLCAL — Performing a reflection calibration.** The calibration is Full Band (performed over the instrument's preset source settings). This example also demonstrates the detection of front panel key presses, the use of softkeys, and the use of the \*OPC? command.

**LOADCALS — Uploading and downloading correction arrays.** The data transfer is performed in the 16-bit integer format. The arrays must be dimensioned properly for both the number of data points and the format of the data being transferred.

**LEARNSTR — Using the learn string to upload and download instrument states.**

**SAVERCL — Saving and recalling instrument states, calibrations and data.** The example also demonstrates saving data in an ASCII file that includes both magnitude and frequency information.

**PRINTPLT — Using the serial and parallel ports for hardcopy output.** The example also demonstrates plotting test results to an HPGL file.

**PASSCTRL — Using pass control and the HP-IB for hardcopy output.** The example uses an HP-IB printer.

**AVERAGE — Generating a service request interrupt.** The example uses the status reporting structure to generate an interrupt as soon as averaging is complete.

**GRAPHICS — Using graphics and softkeys to create customized procedures.** The example demonstrates the use of some of the user graphics commands including the one to erase a previously drawn line. It also demonstrates use of the softkeys and detecting a front panel keypress with the service request interrupt process.

**CALKIT — Instrument state file for downloading User Defined cal kit definitions.** This example is NOT a program. It is an instrument state file example. This type of file enables the user to calibrate the analyzer for use with connector types that are not in the firmware. To see an example of using this feature, refer to "To Calibrate with Other Cal Kits" in chapter 6 of the *Operating Manual*.

Because the examples are designed to run in different environments, the setup at the beginning of each program must determine the operating environment and properly set the analyzer's HP-IB address. In these examples, the internal IBASIC controller uses the address 800 when communicating with the analyzer (the internal HP-IB is at select code 8). The default address of 716 is used when the programs are being run on an external controller.

A version of the following lines is included in all of the example programs. The use of the `Isc` variable (for addressing commands to the bus itself) and the `Internal` (internal-controller) flag vary due to differences in the programs needs.

10 <b>IF</b> SYSTEM\$("SYSTEM ID")="HP 8711" <b>THEN</b>	Identify the operating system.
20       ASSIGN @Hp8711 TO 800	If internal, set address to 800.
30       Isc=8	Set interface select code to 8 (internal).
35       Internal=1	Set internal-control flag to 1.
40 <b>ELSE</b>	
50       ASSIGN @Hp8711 TO 716	If external, set address to 716.
60       Isc=7	Set interface select code to 7 (external).
65       Internal=0	Set internal-control flag to 0.
70       ABORT 7	Abort all bus transactions and give active control of the bus to the computer.
80       CLEAR 716	Send a selected device clear (SDC) to the analyzer – this clears all HP-IB errors, resets the HP-IB interface and clears syntax errors. (It does not affect the status reporting system.)
90 <b>END IF</b>	

## SETUP — Setting up a basic measurement

This program demonstrates how to set up the analyzer to make a basic measurement. The \*WAI command is used extensively throughout this program. This has the effect of making sure that the commands are executed in the order they are received. More information about making measurements with the HP 8711 is available in the *HP 8711 Operating Manual*.

Lines 10-70 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

```
10  IF SYSTEM$("SYSTEM ID")="HP 8711" THEN
20      ASSIGN @Hp8711 TO 800
30  ELSE
40      ASSIGN @Hp8711 TO 716
50      ABORT 7
60      CLEAR 716
70  END IF
80  OUTPUT @Hp8711;"SYST:PRES;*WAI"
90  OUTPUT @Hp8711;"CONF 'FILT:TRAN';*WAI"

100 OUTPUT @Hp8711;"ABOR;:INIT:CONT OFF;*WAI"
110 OUTPUT @Hp8711;"SENS2:STAT ON; *WAI"
120 OUTPUT @Hp8711;"SENS2:FUNC 'XFR:POW:RAT 1,0';
DET NBAN;*OPC?"
125 ENTER @Hp8711;Opc

130 INPUT "Enter Start Frequency (MHz):",Start_f
140 INPUT "Enter Stop Frequency (MHz):",Stop_f
150 OUTPUT @Hp8711;"SENS2:FREQ:STAR";Start_f;
"MHZ;STOP";Stop_f;"MHZ;*WAI"
160 OUTPUT @Hp8711;"INIT;*OPC?"
170 ENTER @Hp8711;Opc
180 OUTPUT @Hp8711;"DISP:WIND1:TRAC:Y:PDIV 10 dB;
RLEV 0 dB;RPOS 8"
190 OUTPUT @Hp8711;"DISP:WIND2:TRAC:Y:PDIV 5 dB;
RLEV 0 dB;RPOS 8"
200 DISP "Start =";Start_f;"MHz      Stop =";
Stop_f;"MHz"
210 END
```

Preset the instrument.

Configure the analyzer to measure transmission of a filter on channel 1 – this is the command for the **BEGIN** **Filter** **Transmission** key sequence.

Put the instrument in trigger hold mode.

Turn on channel 2.

Configure channel 2 to measure reflection – this is the command for the **CHAN 2** **Reflection** key sequence.

Wait for the previous commands to complete execution.

Input a start frequency.

Input a stop frequency.

Set the start and stop frequencies of the analyzer to the values entered.

Trigger a single sweep.

Wait for the sweep to complete.

Set up the scale and reference parameters for channel 1.

Set up the scale and reference parameters for channel 2.

Display the current start and stop frequencies.

## **MARKERS — Transferring data using the markers**

This program demonstrates how to transfer measurement data by using the markers. Before any data is read over the HP-IB a controlled sweep should be taken. The analyzer has the ability to process and execute commands very quickly when they are received over the HP-IB. This speed can lead to commands (such as marker searches) being executed before any data has been taken. To ensure that the sweep has completed and the data is present before it is read, the command for a single sweep is used before data is requested. Note that \*WAI is sent with that command. More information about making measurements with the HP 8711 is available in the *HP 8711 Operating Manual*.

Lines 10-70 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

```
10  IF SYSTEM$("SYSTEM ID")="HP 8711" THEN
20    ASSIGN @Hp8711 TO 800
30  ELSE
40    ASSIGN @Hp8711 TO 716
50    ABORT 7
60    CLEAR 716
70  END IF
80  OUTPUT @Hp8711;"SENS1:STAT ON;FREQ:
STAR 10 MHZ;STOP 400 MHZ;*WAI"

90  OUTPUT @Hp8711;"SENS1:FUNC 'XFR:POW:RAT 2,0';
DET NBAN;*WAI"
100 OUTPUT @Hp8711;"ABOR;:INIT:CONT OFF;
:INIT;*WAI"
110 OUTPUT @Hp8711;"CALC1:MARK1 ON"
120 OUTPUT @Hp8711;"CALC1:MARK1:X 175 MHZ"
130 OUTPUT @Hp8711;"CALC1:MARK1:Y?"
140 ENTER @Hp8711;Data1
150 DISP "Marker 1 (175 MHz) = ";Data1
160 WAIT 5
170 OUTPUT @Hp8711;"CALC1:MARK2 ON;MARK2:MAX"
```

Turn on channel 1 and set up start and stop frequencies for the example – these frequencies were chosen for the demonstration filter that is shipped with the analyzer.

Configure a transmission measurement on channel 1 using the narrowband detection mode.

Take a single controlled sweep and have the analyzer wait until it has completed before executing the next command.

Turn on the first marker.

Set marker 1 to a frequency of 175 MHz.

Query the amplitude of the signal at 175 MHz.

Read the data – the data is in the NR3 format.

Turn on the second marker and use a marker search function to find the maximum point on the data trace.

```

180  OUTPUT @Hp8711;"CALC1:MARK2:X?;Y?"
Query the frequency and amplitude of
the maximum point – note that the two
queries can be combined into one
command.

190  ENTER @Hp8711;Freq_2,Data_2
Read the data.

200  DISP "Max = ";Data_2;"dB at";Freq_2/1.E+6;
"MHz"
Display the results of the marker search.

210  OUTPUT @Hp8711;"INIT:CONT ON;*WAI"
Put the analyzer into its continuously
sweeping mode – this mode works very
well for tuning applications.

220  OUTPUT @Hp8711;"CALC1:MARK2:FUNC:TRAC ON"
Turn on the marker search tracking
function. This function causes the
marker 2 to track the maximum value
each time the analyzer takes a sweep.

230  WAIT 5
Turn off marker 2.

240  OUTPUT @Hp8711;"CALC1:MARK2 OFF"
Take a single controlled sweep.

250  OUTPUT @Hp8711;"ABOR;:INIT:CONT OFF;
:INIT;*WAI"
Perform a search for the -3 dB
bandwidth of the filter. This function
uses several markers to find four key
values.

260  OUTPUT @Hp8711;"CALC1:MARK:BWID -3;FUNC:RES??
Read the four values – the bandwidth,
center frequency, Q and the insertion
loss.

270  ENTER @Hp8711;Bwid,Center_f,Q,Loss
Display the bandwidth search values on
the screen.

280  DISP "BW: ";Bwid,Center_f,Q,Loss
Turn off all the markers.

290  OUTPUT @Hp8711;"CALC1:MARK:AOFF"
300  END

```

## LIMITEST — Performing automatic PASS/FAIL testing with limit lines

This program demonstrates how to set up and use limit lines over the HP-IB. The example device used in this program is the demonstration filter that is shipped with the analyzer. The program sets up the basic measurement, downloads the limit lines and uses the status registers to determine if the device passes its specifications. For more information about limit lines, refer to the *HP 8711 Operating Manual*. For information about using the status registers, refer to the previous section “Using the Status Registers.”

This example also demonstrates how multiple command mnemonics can be combined together. The easiest commands to combine are ones that are closely related on the command tree (such as the start and stop frequency of a limit segment). For more information of command mnemonics, refer to the *HP-IB Command Reference*.

Lines 20-80 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

10	DIM Title\$[30]	
20	IF SYSTEM\$("SYSTEM ID")="HP 8711" THEN	
30	ASSIGN @Hp8711 TO 800	
40	ELSE	
50	ASSIGN @Hp8711 TO 716	
60	ABORT 7	
70	CLEAR 716	
80	END IF	
90	OUTPUT @Hp8711;"SYST:PRES;*WAI"	Perform a system preset – this clears the limit table.
100	OUTPUT @Hp8711;"SENS1:FREQ:STAR 10 MHZ; STOP 400 MHZ;*WAI"	Set up the source frequencies for the measurement.
110	OUTPUT @Hp8711;"SENS1:FUNC 'XFR:POW:RAT 2,0'; DET NBAN;*WAI"	Set up the receiver for the measurement parameters (Transmission in this case).
120	OUTPUT @Hp8711;"DISP:WIND1:TRAC:Y:PDIV 10 DB; RLEV 0 DB;RPOS 9"	Configure the display so measurement results are easy to see.
130	OUTPUT @Hp8711;"DISP:ANN:YAX OFF"	Reduce the distractions on the display by getting rid of notation that won't be used in this example.
140	OUTPUT @Hp8711;"DISP:WIND1:TRAC:GRAT: GRID OFF"	Erase the graticule grid for the same reason.
150	OUTPUT @Hp8711;"CALC1:LIM:SEGM1:TYPE LMAX; STAT ON"	Create and turn on the first segment for the new limit lines – this one is a maximum limit.
160	OUTPUT @Hp8711;"CALC1:LIM:SEGM1:AMPL: STAR -70;STOP -70"	Set the amplitude limits for the first limit segment.
170	OUTPUT @Hp8711;"CALC1:LIM:SEGM1:FREQ: STAR 10 MHZ;STOP 75 MHZ"	Set the frequency range of the first limit segment.

```

180  OUTPUT @Hp8711;"CALC1:LIM:SEGM2:TYPE LMAX;
      STAT ON"
190  OUTPUT @Hp8711;"CALC1:LIM:SEGM2:AMPL:STAR 0;
      STOP 0"
200  OUTPUT @Hp8711;"CALC1:LIM:SEGM2:FREQ:
      STAR 145 MHZ;STOP 200 MHZ"
210  OUTPUT @Hp8711;"CALC1:LIM:SEGM3:TYPE LMIN;
      STAT ON"
220  OUTPUT @Hp8711;"CALC1:LIM:SEGM3:AMPL:STAR -6;
      STOP -6"
230  OUTPUT @Hp8711;"CALC1:LIM:SEGM3:FREQ:
      STAR 150 MHZ;STOP 195 MHZ"
240  OUTPUT @Hp8711;"CALC1:LIM:SEGM4:TYPE LMAX;
      STAT ON"
250  OUTPUT @Hp8711;"CALC1:LIM:SEGM4:AMPL:
      STAR -60;STOP -60"
260  OUTPUT @Hp8711;"CALC1:LIM:SEGM4:FREQ:
      STAR 290 MHZ;STOP 400 MHZ"
270  OUTPUT @Hp8711;"*OPC?"

280  ENTER @Hp8711;Opc
290  OUTPUT @Hp8711;"CALC1:LIM:DISP ON"
300  OUTPUT @Hp8711;"CALC1:LIM:STAT ON"

310  OUTPUT @Hp8711;"ABOR,:INIT1:CONT OFF;
      :INIT1;*WAI"
320  OUTPUT @Hp8711;"STAT:QUES:LIM:COND?"
330  ENTER @Hp8711;Fail_flag
340  IF BIT(Fail_flag,0)=1 THEN

350  Title$="Limit Test FAIL - Tune device"

360  OUTPUT @Hp8711;"DISP:ANN:TITL1:DATA ''
      &Title$'';STAT ON"
370  OUTPUT @Hp8711;"INIT1:CONT ON;*WAI"
380  LOOP
390      OUTPUT @Hp8711;"STAT:QUES:LIM:COND?"
400      ENTER @Hp8711;Fail_flag
410      EXIT IF BIT(Fail_flag,0)=0

```

Create and turn on a second maximum limit segment.

Set the amplitude limits for segment 2.

Set the frequency range for segment 2.

Create and turn on a third limit segment – this one is a minimum limit.

Set the amplitude limits for segment 3.

Set the frequency range for segment 3.

Create and turn on a fourth limit segment – another maximum segment.

Set the amplitude limits for segment 4.

Set the frequency range for segment 4.

Send an operation complete query to ensure that all overlapped commands have been executed.

Wait for the reply.

Turn on display of the limit lines.

Turn on the pass/fail testing –watch the analyzer's display for the pass/fail indicator.

Take a controlled sweep to ensure that there is real data present for the limit test.

Query the limit fail condition register to see if there is a failure.

Read the register's contents.

Bit 0 is the test result for channel 1 while bit 1 is the result for channel 2 testing.

In case of a failure give additional direction to the operator using the title strings.

Turn on the title string.

Turn on continuous sweep mode for tuning.

Loop while the tuning is taking place.

Monitor the status of the limit fail condition register.

Check the limit fail bit. Exit if the device has been tuned to pass the test.

```
420    END LOOP
430    END IF
440    OUTPUT @Hp8711;"DISP:ANN:TITL1 OFF;
:INIT1:CONT ON;*WAI"
450    END
```

Turn off the prompt to the operator  
and return the analyzer to the  
continuously sweeping mode.

## **ASCDATA — Transferring data using the ASCII format**

This program demonstrates how to read data arrays from the analyzer and write them back again. The ASCII data format is being used with a resolution of 5 digits. More information about data transfer is available in the “Transferring Data” section earlier in this chapter.

In addition to the channel 1 formatted data array used in this example, there are a number of arrays that can be accessed inside the instrument. These arrays and their corresponding mnemonics are listed below.

<b>CH1FDATA</b>	Formatted data – channel 1	<b>CH2AFWD</b>	Raw uncorrected measurement data from the A (reflection) channel – channel 2
<b>CH2FDATA</b>	Formatted data – channel 2	<b>CH2BFWD</b>	Raw uncorrected measurement data from the B (transmission) channel – channel 2
<b>CH1SDATA</b>	Error corrected but unformatted data – channel 1	<b>CH2RFWD</b>	Raw uncorrected measurement data from the R (reference) channel – channel 2
<b>CH2SDATA</b>	Error corrected but unformatted data – channel 2	<b>CH1SCORR1</b>	First correction array – channel 1
<b>CH1SMEM</b>	Trace memory, error corrected but unformatted – channel 1	<b>CH1SCORR2</b>	Second correction array – channel 1
<b>CH2SMEM</b>	Trace memory, error corrected but unformatted – channel 1	<b>CH1SCORR3</b>	Third correction array – channel 1
<b>CH1AFWD</b>	Raw uncorrected measurement data from the A (reflection) channel – channel 1	<b>CH2SCORR1</b>	First correction array – channel 2
<b>CH1BFWD</b>	Raw uncorrected measurement data from the B (transmission) channel – channel 1	<b>CH2SCORR2</b>	Second correction array – channel 2
<b>CH1RFWD</b>	Raw uncorrected measurement data from the R (reference) channel – channel 1	<b>CH2SCORR3</b>	Third correction array – channel 2

Lines 20-80 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

```
10  REAL Data1(1:51)
20  IF SYSTEM$("SYSTEM ID")="HP 8711" THEN
30    ASSIGN @Hp8711 TO 800
40  ELSE
50    ASSIGN @Hp8711 TO 716
60    ABORT 7
70    CLEAR 716
80  END IF
```

```

90  OUTPUT @Hp8711;"SENS1:SWE:POIN 51;*WAI"
Set the analyzer to measure 51 data points.

100 OUTPUT @Hp8711;"ABOR;:INIT1:CONT OFF;
:INIT1;*WAI"
Take a single sweep, leaving the analyzer in trigger hold mode.

110 OUTPUT @Hp8711;"FORM:DATA ASC,5;:TRAC?
CH1FDATA"
Set up the ASCII data format with 5 significant digits and request the channel 1 formatted data array from the instrument.

120 ENTER @Hp8711;Data1(*)
Enter the data array.

130 DISP Data1(1),Data1(2),Data1(3),".... "
Display the first three numbers in the data array.

140 WAIT 10
Use this time to visually compare the numbers to the visible data trace.

150 DISP "Disconnect the test device --
Press Continue"
Wait for the operator to disconnect the test device and continue the program.

160 PAUSE
Clear the display line.

165 DISP
Take a sweep so the display shows new data.

170 OUTPUT @Hp8711;":INIT1;*WAI"
Prepare the analyzer to receive the data, suppress the "end" character by using the semicolon.

180 WAIT 5
Send the data array one point at a time, using the semicolon to suppress the "end" character.

190 OUTPUT @Hp8711;"TRAC CH1FDATA";
;

200 FOR I=1 TO 51
Send the data array one point at a time, using the semicolon to suppress the "end" character.

210   OUTPUT @Hp8711;", ";Data1(I);
;

220 NEXT I
Send an "end" character to terminate the input.

230 OUTPUT @Hp8711;"""
;

240 DISP "DATA TRANSFER COMPLETE!"
;

250 END

```

## REALDATA — Transferring data using the IEEE 64-bit floating point REAL format

This program demonstrates how to read data arrays from the analyzer and write them back again. The REAL,64 data format is being used. Note that the analyzer outputs the data using the definite length block syntax. This example uses the indefinite length block syntax when data is being written back to the analyzer. More information about data transfer is available in the “Transferring Data” section earlier in this chapter. All of the arrays listed in the ASCDATA example section can also be accessed using this data format.

Lines 30-70 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

Lines 120-140 check the operating system to determine if it is IBASIC for Windows (a PC based system). When running on a PC based system, swapped byte order must be used for binary data transfers.

```
10  DIM A$[10],Data1(1:51)
20  INTEGER Digits,Bytes
30  IF SYSTEM$("SYSTEM ID")="HP 8711" THEN
40    ASSIGN @Hp8711 TO 800
50  ELSE
60    ASSIGN @Hp8711 TO 716
70    ABORT 7
80    CLEAR 716
90  END IF
100 OUTPUT @Hp8711;"SENS1:SWE:POIN 51;*WAI"
110 OUTPUT @Hp8711;"ABOR;:INIT1:CONT OFF;
:INIT1;*WAI"
120 OUTPUT @Hp8711;"FORM:DATA REAL,64;BORD NORM"
130 IF SYSTEM$("SYSTEM ID")="IBASIC/WINDOWS" THEN
140   OUTPUT @Hp8711;"FORM:BORD SWAP"
150 END IF
160 OUTPUT @Hp8711;"TRAC? CH1FDATA"
170 ASSIGN @Hp8711;FORMAT ON
180 ENTER @Hp8711 USING "%,A,D";A$,Digits
190 ENTER @Hp8711 USING "%,"&VAL$(Digits)&"D";
Bytes
```

Set up the analyzer to measure 51 data points.

Take a single sweep, leaving the analyzer in trigger hold mode.

Set up normal byte order and the REAL,64 data format.

If operating system is a PC:

Set up byte swapping.

Request the channel 1 formatted data array from the instrument.

Turn on ASCII formatting on the I/O path – it is needed for reading the header information.

Enter the data header. A\$ is the # sign indicating a block data transfer, Digits is the number of digits in the number of bytes.

Enter the number of bytes in the data array – note the use of Digits in the command.

```
200 ASSIGN @Hp8711;FORMAT OFF
          Turn off ASCII formatting on the I/O
          path – it is not needed for transferring
          binary formatted data.

210 ENTER @Hp8711;Data1(*)
          Enter the data array.

220 ASSIGN @Hp8711;FORMAT ON
          Turn on ASCII formatting.

230 ENTER @Hp8711;A$
          Enter the “end of data” character.

240 DISP Data1(1),Data1(2),".... "
          Display the first two numbers in the
          data array.

250 WAIT 10
          Use this time to visually compare the
          numbers to the visible data trace.

260 DISP "Disconnect the test device --
          Press Continue"
          Wait for the operator to disconnect the
          test device and continue the program.

270 PAUSE
          Clear the display line.

275 DISP
          Take a sweep so the display shows new
          data.

280 OUTPUT @Hp8711;":INIT1;*WAI"
          Send the header for an indefinite block
          length data transfer.

290 WAIT 5
          Turn off ASCII formatting.

300 OUTPUT @Hp8711;"TRAC CH1FDATA, #0";
          Send the data back to the analyzer.

310 ASSIGN @Hp8711;FORMAT OFF
          Turn on ASCII formatting.

320 OUTPUT @Hp8711;Data1(*),END
          Send the data back to the analyzer.

330 ASSIGN @Hp8711;FORMAT ON
          Turn on ASCII formatting.

340 DISP "DATA TRANSFER COMPLETE!"
          Turn on ASCII formatting.

350 END
```

## INTDATA — Transferring data using the 16-bit INTEGER format

This program demonstrates how to read data arrays from the analyzer and write them back again. The INTeger,16 data format is being used. This data format is the instrument's internal format. It should only be used to read data that will later be returned to the instrument.

The data array dimensioned in line 20 is different from the arrays in either REAL,64 or ASCII examples. This is because each data point is represented by a set of three 16-bit integers. Another difference in using this data format is that all arrays cannot be accessed with it. The formatted data arrays CH1FDATA and CH2FDATA cannot be read using the INTEGER format.

Note that the analyzer outputs the data using the definite length block syntax. This example uses the indefinite length block syntax when data is being written back to the analyzer. More information about data transfer is available in the "Transferring Data" section earlier in this chapter.

Lines 30-70 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

Lines 120-140 check the operating system to determine if it IBASIC for Windows (a PC based system). When running on a PC based system, swapped byte order must be used for binary data transfers.

```
10  DIM A$[10]
20  INTEGER Digits,Bytes,Data1(1:51,1:3)
30  IF SYSTEM$("SYSTEM ID")="HP 8711" THEN
40    ASSIGN @Hp8711 TO 800
50  ELSE
60    ASSIGN @Hp8711 TO 716
70    ABORT 7
80    CLEAR 716
90  END IF
100 OUTPUT @Hp8711;"SENS1:SWE:POIN 51;*WAI"      Set up the analyzer to measure 51 data points.
110 OUTPUT @Hp8711;"ABOR;:INIT1:CONT OFF;
:INIT1;*WAI"                                     Take a single sweep, leaving the analyzer in trigger hold mode.
120 OUTPUT @Hp8711;"FORM:DATA INT,16;BORD NORM"   Set up normal byte order and the INTeger,16 data format.
130 IF SYSTEM$("SYSTEM ID")="IBASIC/WINDOWS" THEN
140   OUTPUT @Hp8711;"FORM:BORD SWAP"             If operating system is a PC:
150 END IF                                         Set up byte swapping.
160 OUTPUT @Hp8711;"TRAC? CH1SDATA"              Request the channel 1 unformatted data array from the instrument.
170 ASSIGN @Hp8711;FORMAT ON                      Turn on ASCII formatting on the I/O path – it is needed for reading the header information.
```

<pre> 180 ENTER @Hp8711 USING "%,A,D";A\$,Digits  190 ENTER @Hp8711 USING "%,"&amp;VAL\$(Digits)&amp;"D";       Bytes  200 ASSIGN @Hp8711;FORMAT OFF  210 ENTER @Hp8711;Data1(*) 220 ASSIGN @Hp8711;FORMAT ON 230 ENTER @Hp8711;A\$ 240 DISP Data1(1,1),Data1(1,2),Data1(1,3),"...."  250 WAIT 10  260 DISP "Disconnect the test device --       Press Continue" 270 PAUSE  275 DISP 280 OUTPUT @Hp8711;":INIT1,*WAI"  290 WAIT 5 300 OUTPUT @Hp8711;"TRAC CH1SDATA, #0";  310 ASSIGN @Hp8711;FORMAT OFF 320 OUTPUT @Hp8711;Data1(*),END 330 ASSIGN @Hp8711;FORMAT ON 340 DISP "DATA TRANSFER COMPLETE!" 350 END </pre>	<p>Enter the data header. A\$ is the # sign indicating a block data transfer, Digits is the number of digits in the number of bytes.</p> <p>Enter the number of bytes in the data array – note the use of Digits in the command.</p> <p>Turn off ASCII formatting on the I/O path – it is not needed for transferring binary formatted data.</p> <p>Enter the data array.</p> <p>Turn on ASCII formatting.</p> <p>Enter the “end of data” character.</p> <p>Display the first three numbers in the data array – there is no visible similarity between these numbers and the data displayed on the analyzer.</p> <p>Use this time to visually compare the numbers to the visible data trace.</p> <p>Wait for the operator to disconnect the test device and continue the program.</p> <p>Clear the display line.</p> <p>Take a sweep so the display shows new data.</p> <p>Send the header for an indefinite block length data transfer.</p> <p>Turn off ASCII formatting.</p> <p>Send the data back to the analyzer.</p> <p>Turn on ASCII formatting.</p>
---	--

## TRANCAL — Performing a transmission calibration

This program demonstrates a transmission calibration performed over user-defined source settings (frequency range, power and number of points). The operation complete query is used at each step in the process to make sure the steps are taken in the correct order. More information on calibration is available in the *HP 8711 Operating Manual*.

Lines 10-70 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

10	IF SYSTEM\$("SYSTEM ID")="HP 8711" THEN	
20	ASSIGN @Hp8711 TO 800	
30	ELSE	
40	ASSIGN @Hp8711 TO 716	
50	ABORT 7	
60	CLEAR 716	
70	END IF	
80	OUTPUT @Hp8711;"SENS1:FUNC 'XFR:POW: RAT 2,0'; DET NBAN;*WAI"	Configure the analyzer to measure transmission on channel 1.
90	OUTPUT @Hp8711;"SENS:CORR:COLL:CKIT 'COAX,7MM, TYPE-N,50,FEMALE'"	Select a calibration kit type.
100	OUTPUT @Hp8711;"SENS1:CORR:COLL:IST OFF; METH TRAN1"	Select a transmission calibration for the current source settings – the IST OFF part of the mnemonic indicates that the current source settings will be used.
110	DISP "Connect THRU - Press Continue"	Prompt the operator to make a through connection.
120	PAUSE	Pause until the thru connection is made and continue is pressed.
130	DISP "Measuring THRU"	
140	OUTPUT @Hp8711;"SENS1:CORR:COLL STAN1;*OPC?"	Measure the thru.
150	ENTER @Hp8711;0pc	Wait until the measurement is complete.
160	DISP "Calculating Error Coefficients"	
170	OUTPUT @Hp8711;"SENS1:CORR:COLL:SAVE;*OPC?"	Calculate the error coefficients and save the calibration to internal memory – this is not the same as saving the calibration using the <b>SAVE RECALL</b> key.
180	ENTER @Hp8711;0pc	Wait for the calculations to complete and the calibration to be saved.
190	DISP "User Defined TRANSMISSION CAL COMPLETED!"	
200	END	

## REFLCAL — Performing a reflection calibration

This program demonstrates a reflection calibration performed over the preset source settings (frequency range, power and number of points). The operation complete query is used at each step in the process to make sure the steps are taken in the correct order. More information on calibration is available in the *HP 8711 Operating Manual*.

Lines 20-100 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

10	DIM Msg\$[50]	
20	IF SYSTEM\$("SYSTEM ID")="HP 8711" THEN	
30	ASSIGN @Hp8711 TO 800	
40	Internal=1	
50	ELSE	
60	ASSIGN @Hp8711 TO 716	
70	Internal=0	
80	ABORT 7	
90	CLEAR 716	
100	END IF	
110	OUTPUT @Hp8711;"SENS1:FUNC 'XFR:POW:RAT 1,0'; DET NBAN;*WAI"	Configure the analyzer to measure reflection on channel 1.
120	OUTPUT @Hp8711;"SENS:CORR:COLL:CKIT 'COAX,7MM, TYPE-N,50,FEMALE'"	Select a calibration kit type.
130	OUTPUT @Hp8711;"SENS1:CORR:COLL:IST ON; METH REFL3"	Select a reflection calibration for the preset source settings – the IST ON part of the mnemonic indicates that the preset source configuration will be used.
140	Msg\$="Connect OPEN"	
150	GOSUB Get_continue	Prompt the operator to connect an open and press a key.
160	DISP "Measuring OPEN"	
170	OUTPUT @Hp8711;"SENS1:CORR:COLL STAN1;*OPC?"	Measure the open.
180	ENTER @Hp8711;0pc	Wait until the measurement is complete.
190	Msg\$="Connect SHORT"	
200	GOSUB Get_continue	Prompt the operator to connect an short and press a key.
210	DISP "Measuring SHORT"	
220	OUTPUT @Hp8711;"SENS1:CORR:COLL STAN2;*OPC?"	Measure the short.
230	ENTER @Hp8711;0pc	Wait until the measurement is complete.

```

240  Msg$="Connect LOAD"
250  GOSUB Get_continue

260  DISP "Measuring LOAD"
270  OUTPUT @Hp8711;"SENS1:CORR:COLL STAN3;*OPC?"
280  ENTER @Hp8711;Opc

290  DISP "Calculating Error Coefficients"
300  OUTPUT @Hp8711;"SENS1:CORR:COLL:SAVE;*OPC?"

310  ENTER @Hp8711;Opc

320  DISP "Full Band REFLECTION CAL COMPLETED!"
330  STOP

340 Get_continue: !
350  IF Internal=1 THEN
360    DISP Msg$&" - Press Continue"

370    ON KEY 2 LABEL "Continue" RECOVER Go_on
380    LOOP

390    END LOOP
400  ELSE
410    OUTPUT @Hp8711;"SYST:KEY:QUE:CLE"

420    DISP Msg$&" - Press BEGIN to continue"
430    OUTPUT @Hp8711;"SYST:KEY:QUE ON"
440    LOOP

450    OUTPUT @Hp8711;"STAT:DEV:COND?"
460    ENTER @Hp8711;Dev_cond
470    IF BIT(Dev_cond,0)=1 THEN

480      OUTPUT @Hp8711;"SYST:KEY?"

490      ENTER @Hp8711;Key_code
500      END IF
510      EXIT IF Key_code=40

```

*Prompt the operator to connect a load and press a key.*

*Measure the load.  
Wait until the measurement is complete.*

*Calculate the error coefficients and save the calibration – this is not the same as saving the calibration using the **SAVE RECALL** key.  
Wait for the calculations to complete and the calibration to be saved.*

*If internal control:  
Use the display line for the prompt.*

*Use the softkeys for the response.  
Loop while waiting for softkey 2 to be pressed.*

*If external control:  
Clear the key queue so previous key presses will not interfere.*

*Use the **BEGIN** key for the response.*

*Turn on the key queue to trap all key presses.  
Loop while waiting for a key to be pressed.*

*Read the device status condition register.*

*Check the bit that indicates a key press.*

*Read the code for the key that was pressed.*

*Stop looping if the **BEGIN** key was pressed.*

```
520    END LOOP
530    Key_code=0
600    GOSUB 1000
610    Key_code=0
620    GOSUB 1000
630    GOSUB 1000
640    GOSUB 1000
650    GOSUB 1000
660    GOSUB 1000
670    GOSUB 1000
680    GOSUB 1000
690    GOSUB 1000
700    GOSUB 1000
710    GOSUB 1000
720    GOSUB 1000
730    GOSUB 1000
740    GOSUB 1000
750    GOSUB 1000
760    GOSUB 1000
770    GOSUB 1000
780    GOSUB 1000
790    GOSUB 1000
800    GOSUB 1000
810    GOSUB 1000
820    GOSUB 1000
830    GOSUB 1000
840    GOSUB 1000
850    GOSUB 1000
860    GOSUB 1000
870    GOSUB 1000
880    GOSUB 1000
890    GOSUB 1000
900    GOSUB 1000
910    GOSUB 1000
920    GOSUB 1000
930    GOSUB 1000
940    GOSUB 1000
950    GOSUB 1000
960    GOSUB 1000
970    GOSUB 1000
980    GOSUB 1000
990    GOSUB 1000
1000   END
```

*Clear the key\_code to start fresh the next time the routine is called.*

*Turn off the softkeys on the instrument and the computer. Return to the main body of the program.*

## LOADCALS — Uploading and downloading correction arrays

This program demonstrates how to read the correction arrays from the analyzer and write them back again. The INTEGER,16 data format is being used because the data does not need to be interpreted, only stored and retrieved. More information about calibration is available in the *HP 8711 Operating Manual*.

The size of the arrays into which the data is read is critical. If they are not dimensioned correctly the program will not work. Most correction arrays, including the factory default (DEF) and the full band (FULL, preset source settings) arrays have 801 points. For user defined calibrations (USER) the number of points must be determined. If the number of points is other than 801 line 30 will need to be changed to allocate arrays for the correct number of points. The number of points can be found by reading the correction array's header and determining the size as shown in the example below.

Lines 40-110 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

```
10  DIM Func$[20],A$[10]
20  INTEGER Swap,Arrays,Digits,Bytes,Points
30  INTEGER Corr1(1:801,1:3),Corr2(1:801,1:3),
    Corr3(1:801,1:3)
40  IF SYSTEM$("SYSTEM ID")="HP 8711" THEN
50      ASSIGN @Hp8711 TO 800
60  ELSE
70      ASSIGN @Hp8711 TO 716
80      IF SYSTEM$("SYSTEM ID")="IBASIC/WINDOWS"
        THEN Swap=1
90      ABORT 7
100     CLEAR 716
110 END IF
120 OUTPUT @Hp8711;"SENS1:FUNC?"
130 ENTER @Hp8711;Func$
140 SELECT Func$
150 CASE """XFR:POW:RAT 2, 0"""
160     Arrays=1
170 CASE """XFR:POW:RAT 1, 0"""
180     Arrays=3
```

If operating system is a PC set a flag for byte swapping.

Query the measurement parameter.  
Read the analyzer's response.

This is the transmission case – a ratio of the powers being measured by detector 2 (B) and detector 0 (R).  
The transmission calibration has only one correction array.

This is the reflection case – a ratio of the powers being measured by detector 1 (A) and detector 0 (R).  
The reflection calibration has three correction arrays.

```

190 END SELECT
200 OUTPUT @Hp8711;"FORM:DATA INT,16"

210 OUTPUT @Hp8711;"FORM:BORD NORM"
220 IF Swap=1 THEN OUTPUT @Hp8711;
"FORM:BORD SWAP"
230 OUTPUT @Hp8711;"TRAC? CH1SCORR1"

240 ASSIGN @Hp8711;FORMAT ON

250 ENTER @Hp8711 USING "%,A,D";A$,Digits

260 ENTER @Hp8711 USING "%,"&VAL$(Digits)&"D";
Bytes
270 Points=Bytes/6

280 IF Points<>801 THEN

290   DISP "Arrays are not dimensioned for this
calibration"

300 STOP
310 END IF
320 DISP "Uploading calibration arrays ... "
330 ASSIGN @Hp8711;FORMAT OFF

340 ENTER @Hp8711;Corr1(*)
350 ASSIGN @Hp8711;FORMAT ON
360 ENTER @Hp8711;A$
370 IF Arrays=3 THEN

380   OUTPUT @Hp8711;"TRAC? CH1SCORR2"

390   Read_array(@Hp8711,Corr2(*))
400   OUTPUT @Hp8711;"TRAC? CH1SCORR3"
410   Read_array(@Hp8711,Corr3(*))

420 END IF
430 DISP "Calibration arrays have been uploaded."
440 WAIT 5
450 DISP "Downloading calibration arrays ... "
460 OUTPUT @Hp8711;"SENS1:CORR:STAT OFF"

```

Select up the 16-bit integer binary data format.

Select the normal byte order.

If operating on a PC based system then change to the swapped byte order.

Request the first correction array from the analyzer.

Turn on ASCII formatting on the I/O path to read the file header.

Read the file header, including the number of digits needed to read the size of the file.

Read the size of the file.

Determine the number of points in the array from the size of the file.

This example was only designed for 801 points.

If another number of points is needed the arrays in line 30 above need to be dimensioned appropriately.

Turn off ASCII formatting on the I/O path.

Read the first correction array.

Turn on ASCII formatting.

Read the “end of file” character.

If this is a reflection calibration there are two more arrays.

Request the second correction array from the analyzer.

Read the second correction array.

Request the third correction array.

Read the third correction array.

Turn off correction before writing a calibration back into the analyzer.

```

470 OUTPUT @Hp8711;"SENS1:SWE:POIN";Points
Set the number of points for the
correction arrays – even if the
calibration was originally the default
from the factory or a full band
calibration, it is considered to be a
user-defined calibration when it is
downloaded.

480 OUTPUT @Hp8711;"TRAC CH1SCORR1, #0";
Prepare the analyzer to receive the first
correction array in the indefinite block
length format.

490 ASSIGN @Hp8711;FORMAT OFF
Turn off ASCII formatting.

500 OUTPUT @Hp8711;Corr1(*),END
Send the first correction array to the
analyzer. The array is followed
immediately by an “end of transfer”
signal.

510 ASSIGN @Hp8711;FORMAT ON
Turn on ASCII formatting.

520 IF Arrays=3 THEN
If this is a reflection calibration there
are two more arrays.

530 OUTPUT @Hp8711;"TRAC CH1SCORR2, ";
Prepare the analyzer to receive the
second array.

540 Write_array(@Hp8711,Corr2(*))
Send the second array.

550 OUTPUT @Hp8711;"TRAC CH1SCORR3, ";
Prepare the analyzer to receive the
third array.

560 Write_array(@Hp8711,Corr3(*))
Send the third array.

570 END IF
Turn on the calibration.

580 OUTPUT @Hp8711;"SENS1:CORR:STAT ON;*WAI"
DISP "Calibration arrays have been
downloaded."
600 END
610 SUB Read_array(@Hp8711,INTEGER Array(*))
This subprogram contains the standard
form used for transferring <block> data
in the binary formats. It assumes that
the command requesting the data has
already been sent.

620 DIM A$[10]
Dimension a string character to receive
the header and end of file symbols.

630 INTEGER Digits,Bytes
Declare integer variables for the number
of digits and the number of bytes.

640 ASSIGN @Hp8711;FORMAT ON
Turn on ASCII formatting to read the
header (some of the header is
non-numeric).

650 ENTER @Hp8711 USING "%,A,D";A$,Digits
Enter the header character and the
number of digits needed to read the size
of the file.

660 ENTER @Hp8711 USING "%,&VAL$(Digits)&"D";
Bytes
Enter the number of bytes in the file
(not including the header).

```

670	ASSIGN @Hp8711;FORMAT OFF	Turn off ASCII formatting for the transfer of binary data.
680	ENTER @Hp8711;Array(*)	Enter the data array.
690	ASSIGN @Hp8711;FORMAT ON	Turn on ASCII formatting to read the non-numeric character(s) at the end of the file.
700	ENTER @Hp8711;A\$	Enter the end of file characters.
710	SUBEND	
720	SUB Write_array(@Hp8711,INTEGER Array(*))	This subprogram contains the standard form used for writing <block> data to the analyzer in the binary formats. It assumes that the command preparing the analyzer for the data has already been sent and the I/O path has been left with ASCII formatting on.
730	OUTPUT @Hp8711;"#0";	Send the header, #0 signifies a block data transfer of undefined length.
740	ASSIGN @Hp8711;FORMAT OFF	Turn off ASCII formatting to write the binary formatted data.
750	OUTPUT @Hp8711;Array(*),END	Write the binary formatted data array. Follow it immediately with a BASIC END statement that signifies the end of the data transfer.
760	ASSIGN @Hp8711;FORMAT ON	Turn on ASCII formatting.
770	SUBEND	

## **LEARNSTR — Using the learn string to upload and download instrument states**

This program demonstrates how to upload and download instrument states using the learn string. The learn string is a fast and easy way to read an instrument state. It is read out using the \*LRN? query (an IEEE 488.2 common commands). To restore the learn string simply output the string to the analyzer.

The learn string contains a mnemonic at the beginning that tells the analyzer to restore the instrument state.

The learn string is transferred as a block. The header is ASCII formatted and the data is in the instrument's internal binary format. The number of bytes in the block of data is determined by the instrument state (no more than 20000 bytes).

```
"SYST:SET #<digits><bytes><learn string data>"
```

Lines 20-80 are explained in the introduction to the example programs section. They determine which system controller is being used and prepare the instrument for remote operation.

```
10  DIM Learnstr$[20000]
20  IF SYSTEM$("SYSTEM ID")="HP 8711" THEN
30    ASSIGN @Hp8711 TO 800
40  ELSE
50    ASSIGN @Hp8711 TO 716
60    ABORT 7
70    CLEAR 716
80  END IF
90  OUTPUT @Hp8711; "*LRN?"
100 ENTER @Hp8711 USING "-K";Learnstr$
```

Request the learnstring.

Read the learn string from the analyzer. The USING "-K" format allows the data being transmitted to include characters (such as the line feed character) that would otherwise terminate the input.

```
110 DISP "Learn string has been read"
```

Preset the analyzer.

```
120 WAIT 5
130 OUTPUT @Hp8711; "SYST:PRES;*OPC?"
```

Wait for the preset operation to complete.

```
140 ENTER @Hp8711;Opc
```

```
150 DISP "Instrument has been PRESET"
```

Output the learnstring to the analyzer. The mnemonic is included in the string so no command is necessary.

```
160 WAIT 5
170 OUTPUT @Hp8711;Learnstr$
```

```
180 DISP "Instrument state has been restored"
```

```
190 END
```

## SAVERCL — Saving and recalling instrument states, calibrations and data

This program demonstrates how to save instrument states, calibrations and data to a mass storage device. The device used in this example is the HP 8711's internal floppy disk drive. The only change needed to use this program with the internal non-volatile memory is to change the mass storage unit specifier. The three choices are the internal floppy disk drive ('INT:'), the internal non-volatile memory ('MEM:') and an external HP-IB floppy disk drive ('EXT:'). To perform a save/recall to an external disk drive requires passing control of the HP-IB from the controller to the analyzer. For more information on passing control of the bus refer to the previous "Passing Control" section and the PASSCTRL example program.

Lines 10-70 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

Lines 80-130 are an example of saving an instrument state and calibration on the internal floppy disk drive.

Lines 190-200 are an example of recalling that instrument state and calibration.

Lines 210-230 are an example of saving a data trace (magnitude and frequency values) to an ASCII formatted file on the internal floppy disk drive. This file cannot be recalled into the instrument. It can, however, be imported directly into spreadsheets and word processors.

10    IF SYSTEM\$("SYSTEM ID")="HP 8711" THEN	
20       ASSIGN @Hp8711 TO 800	
30    ELSE	
40       ASSIGN @Hp8711 TO 716	Select the internal floppy disk drive as the mass storage device.
50       ABORT 7	
60       CLEAR 716	
70    END IF	
80    OUTPUT @Hp8711;"MMEM:MSIS 'INT:'"	Turn on the instrument state as part of the definition of "Save/Recall".
90    OUTPUT @Hp8711;"MMEM:STOR:STAT:IST ON"	
100    OUTPUT @Hp8711;"MMEM:STOR:STAT:CORR ON"	Turn on the calibration so it will be saved with the instrument state.
110    OUTPUT @Hp8711;"MMEM:STOR:STAT:TRAC OFF"	
120    OUTPUT @Hp8711;"MMEM:STOR:STAT 1,'FILTER'; *OPC?"	Turn off the trace data.
130    ENTER @Hp8711;Opc	Save the current instrument state (STAT 1) into a file named 'FILTER'. Use *OPC? to make sure the operation is completed before any other operation begins.
140    DISP "Instrument state and calibration have been saved"	Wait for the reply.

```
150 OUTPUT @Hp8711;"SYST:PRES;*OPC?"  
  
160 ENTER @Hp8711;Opc  
170 DISP "Instrument has been PRESET"  
180 WAIT 5  
190 OUTPUT @Hp8711;"MMEM:LOAD:STAT 1,  
'INT:FILTER';*OPC?"  
  
200 ENTER @Hp8711;Opc  
210 OUTPUT @Hp8711;"ABOR;:INIT:CONT OFF;  
:INIT;*WAI"  
220 DISP "Instrument state and calibration  
have been recalled"  
230 OUTPUT @Hp8711;"MMEM:STOR:TRAC CH1FDATA,  
'INT:DATA0001'"  
  
240 DISP "Data has been saved (ASCII format)"  
250 END
```

Preset the instrument so the change in instrument state is easy to see.

Wait for the preset to complete.

Load the file 'FILTER' off the internal floppy disk drive. This becomes the new instrument state. Use the \*OPC? query to hold off further commands until the analyzer is reconfigured.

Wait for the reply.

Take a single sweep to make sure there is valid measurement data present.

Save that measurement data into an ASCII file (named DATA0001) on the internal drive.

## PRINTPLT — Using the serial and parallel ports for hardcopy output

This program demonstrates how to send a hardcopy to a printer on the serial interface. This is done by selecting the appropriate device, setting up the baud rate and hardware handshaking, and sending the command to print or plot. The \*OPC? query is used in this example to indicate when the printout is complete. Another method of obtaining the same results is to monitor the Hardcopy in Progress bit (bit 9 in the Operational Status Register). More information on printing or plotting is available in the *HP 8711 Operating Manual*.

Lines 10-70 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

Lines 80-150 demonstrate sending a hardcopy output to a printer connected to the serial port. The same program could be used to send hardcopy output to a device on the parallel port. The only changes would be deleting lines 100-110 and changing line 90 to read HCOP:DEV:PORT PAR.

Lines 160-260 demonstrate how to create an HPGL file (plotter language) and send it to the disk in the internal floppy disk drive. HPGL files are supported by many applications including the leading word processors and desktop publishing products.

10        IF SYSTEM\$("SYSTEM ID")="HP 8711" THEN	
20        ASSIGN @Hp8711 TO 800	
30        ELSE	
40        ASSIGN @Hp8711 TO 716	Select the output language (PCL – printer control language).
50        ABORT 7	Select the baud rate.
60        CLEAR 716	Select the handshaking protocol.
70        END IF	Select the type of output – listing of values table.
80        OUTPUT @Hp8711;"HCOP:DEV:LANG PCL;PORT SER"	Send the command to start a hardcopy to the selected device. Use the *OPC? query to make sure the hardcopy is complete before continuing.
	Wait for a reply.
90        OUTPUT @Hp8711;"SYST:COMM:SER:TRAN:BAUD 19200"	Select the HPGL output language (for graphics) and the output device (mass memory).
100      OUTPUT @Hp8711;"SYST:COMM:SER:TRAN:HAND XON"	Include the data trace in the plot (default).
110      OUTPUT @Hp8711;"HCOP:DEV:MODE TABL"	
120      OUTPUT @Hp8711;"HCOP;*OPC?"	
130      ENTER @Hp8711;Opc	
140      DISP "Hardcopy to serial printer - COMPLETE!"	
150      OUTPUT @Hp8711;"HCOP:DEV:LANG HPGL;PORT MMEM"	
160      OUTPUT @Hp8711;"HCOP:ITEM:TRAC:STAT ON"	

170	OUTPUT @Hp8711;"HCOP:ITEM:GRAT:STAT OFF"	<i>Do not show the graticule in the plot (default is on).</i>
180	OUTPUT @Hp8711;"HCOP:ITEM:ANN:STAT ON"	<i>Include the frequency and measurement annotation in the plot (default).</i>
190	OUTPUT @Hp8711;"HCOP:ITEM:MARK:STAT ON"	<i>Include the marker symbols in the plot (default).</i>
200	OUTPUT @Hp8711;"HCOP:ITEM:TITL:STAT ON"	<i>Include the title and/or time/date stamp in the plot (default).</i>
210	OUTPUT @Hp8711;"HCOP:DEV:MODE GMAR"	<i>Define the plot to be both the graph and a marker table if markers are in use.</i>
220	OUTPUT @Hp8711;"HCOP;*OPC?"	<i>Begin the plot to a file, use *OPC?.</i>
230	ENTER @Hp8711;0pc	
240	DISP "Plot to floppy disk - COMPLETE!"	<i>Wait for the reply.</i>
250	END	

## PASSCTRL — Using pass control and the HP-IB for hardcopy output

This program demonstrates how to send a hardcopy to an HP-IB printer. This is done by passing active control of the bus to the analyzer so it can control the printer. More information about passing control to the analyzer is available in the “Passing Control” section earlier in this chapter.

Lines 10-90 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

```
10  IF SYSTEM$("SYSTEM ID")="HP 8711" THEN          Select the output port (HP-IB) and
20    ASSIGN @Hp8711 TO 800                          language (PCL – printer control
30    Internal=1                                     language).
40  ELSE
50    ASSIGN @Hp8711 TO 716
60    Internal=0
70    ABORT 7
80    CLEAR 716
90  END IF
100 OUTPUT @Hp8711;"HCOP:DEV:LANG PCL;PORT GPIB"      Set the HP-IB address of the output
110 OUTPUT @Hp8711;"SYST:COMM:GPIB:HCOP:ADDR 1"        device.
120 OUTPUT @Hp8711;"HCOP:DEV:MODE GRAP"                Select the type of output – graph only.
130 IF Internal=1 THEN
140   OUTPUT @Hp8711;"SYST:COMM:GPIB:CONT ON"          If the internal controller is being used:
150 END IF
160 OUTPUT @Hp8711;"*CLS"                            make the analyzer the system controller.
170 OUTPUT @Hp8711;"*ESE 2"
180 OUTPUT @Hp8711;"*SRE 0"
190 OUTPUT @Hp8711;"HCOP"
200 LOOP
210   Stat=SPOLL(@Hp8711)
220 EXIT IF BIT(Stat,5)=1
230 END LOOP

```

Clear all the event registers.  
Enable the Request Control bit in the event status register.  
Clear the Service Request enable register – SRQ is not being used.  
Start the hardcopy output.  
Read the status byte using a serial poll.  
Exit when the analyzer requests control of the bus by setting bit 5 of the status byte.

```
240 PASS CONTROL @Hp8711          Pass control of the bus to the analyzer.  
250 DISP "Hardcopy in Progress"  
260 IF Internal=1 THEN            If using the internal controller:  
270   OUTPUT @Hp8711;"*OPC?"  
280   ENTER @Hp8711;0pc           Send the operation complete query.  
290 ELSE  
300   LOOP  
310     STATUS 7,6;Hplib          Wait for the reply when the hardcopy is  
320     EXIT IF BIT(Hplib,6)=1      complete.  
330   END LOOP  
340 END IF  
350 DISP "HARDCOPY COMPLETE!"  
360 END
```

## AVERAGE -- Generating a service request interrupt

This program demonstrates generating a service request interrupt. The SRQ is used to indicate when averaging is complete. More information on service requests and the status registers is available in the “Using the Status Registers” section earlier in this chapter.

In this program, the STATUS:PRESet executed in line 130 has the effect of setting all bits in the averaging status transition registers (positive transitions to 0, negative transitions to 1). It also sets up the operational status transition registers (positive transitions to 1, negative transitions to 0). These are the states needed to generate an interrupt when averaging is complete.

Lines 10–90 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

10	IF SYSTEM\$("SYSTEM ID")="HP 8711" THEN	
20	ASSIGN @Hp8711 TO 800	
30	Isc=8	
40	ELSE	
50	ASSIGN @Hp8711 TO 716	
60	Isc=7	
70	ABORT 7	
80	CLEAR 716	
90	END IF	
100	OUTPUT @Hp8711; "*CLS"	Clear all the event registers.
110	OUTPUT @Hp8711; "*SRE 0"	Clear the Service Request Enable register.
120	OUTPUT @Hp8711; "*ESE 0"	Clear the Standard Event Status Enable register.
130	OUTPUT @Hp8711; "STAT:PRES"	Preset the remaining status registers.
140	OUTPUT @Hp8711; "STAT:OPER:ENAB 256"	Set operational status register to report to the status byte on positive transition of the averaging bit.
150	OUTPUT @Hp8711; "STAT:OPER:AVER:ENAB 1"	Set averaging status register to report to the operational status register on negative transition of the averaging done bits.
160	OUTPUT @Hp8711; "*SRE 128"	Enable the operational status bit in the status byte to generate an SRQ.

```

170  ON INTR Isc,2 GOSUB Srq_handler      Set up a program branch to handle the
                                            interrupt.

180  ENABLE INTR Isc;2                    Enable an interrupt from the interface
                                            (HP-IB) when the SRQ bit (bit 1 –
                                            decimal value 2) is set.

190  OUTPUT @Hp8711;"SENS1:AVER:COUN 8;*WAI"  Set an averaging factor of 8.

200  OUTPUT @Hp8711;"SENS1:AVER ON;AVER:CLE;*WAI"  Turn on and restart averaging.

210  OUTPUT @Hp8711;"ABOR;:INIT1:CONT ON;*WAI"  Turn on the continuously sweeping
                                            trigger mode.

220  Avg_done=0                          Set a flag indicating that averaging is
                                            not complete.

230  LOOP                                Loop while waiting for an interrupt.

240  DISP "Waiting for average to complete"  Check the flag and exit if averaging is
                                            done.

250  EXIT IF Avg_done=1

260  END LOOP

270  DISP "AVERAGING COMPLETE!"

280  STOP

290  !

300  Srq_handler:!
310  Stb=SPOLL(@Hp8711)                  Determine if the HP 8711 generated the
                                            interrupt.

320  IF BINAND(Stb,128)<>0 THEN        Determine if the operational status
                                            register caused the interrupt.

330  OUTPUT @Hp8711;"STAT:OPER:EVEN?"    Read the operational status event
                                            register.

340  ENTER @Hp8711;Op_event
350  IF BINAND(Op_event,256)<>0 THEN    Determine if the averaging status
                                            register caused the interrupt.

360  Avg_done=1                          Set the flag indicating that averaging is
                                            complete.

370  END IF

380  END IF

390  RETURN                               Return to the main program.

400  END

```

## GRAPHICS — Using graphics and softkeys to create a customized procedure

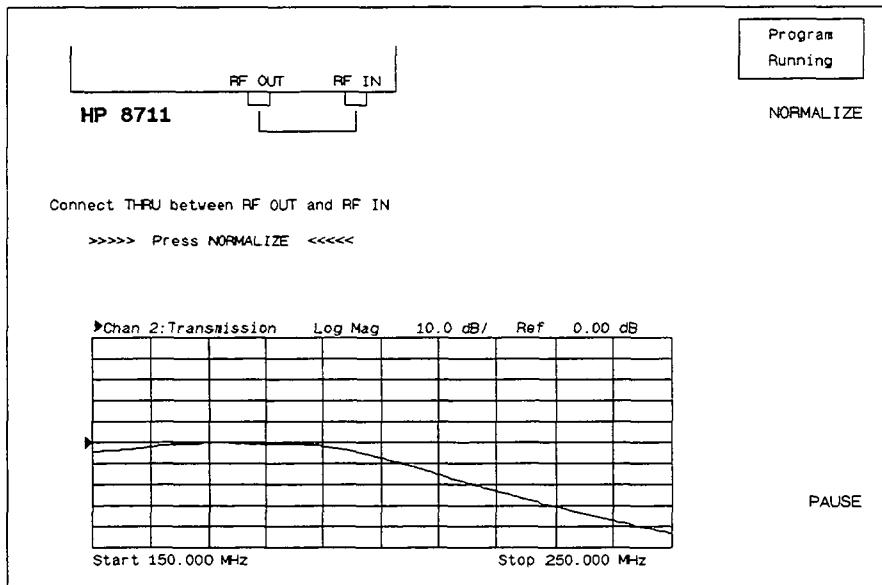
This program demonstrates how to use the HP 8711's user graphics commands to draw setup diagrams. It also demonstrates generating a service request in response to a keyboard interrupt. More information on user graphics commands is available in the previous section, "User Graphics" and in the *HP-IB Command Reference*. Information on generating a service request and using the status reporting structure is in the "Using the Status Registers" section.

Note that this program uses the HP 8711's user graphics commands. If the IBASIC option is installed, graphics may sometimes be more easily implemented using BASIC commands such as POLYGON and RECTANGLE. For further information, see the "BAR" program description in the *HP Instrument BASIC Users Handbook*.

Lines 10-110 are explained in the introduction to the example programs section. They determine which system controller is being used, set flags, and prepare the instrument for remote operation.

Lines 170-240 draw and label a representation of the HP 8711 for a connection diagram. This example is a simple front view from the top.

Lines 250-450 draw the connection needed for a normalization. The operator is prompted to make this connection and to press a softkey on the instrument. A flashing message is used to attract attention.



**GRAPHICS example connection diagram**

Lines 460-580 perform the normalization, erase the prompts (without erasing the whole screen) and prepare for the test.

Lines 590-730 are a branching routine that handles the service request generated interrupts used by the external controller.

```

10  IF SYSTEM$("SYSTEM ID")="HP 8711" THEN
20    ASSIGN @Hp8711 TO 800
30    Internal=1
40    Isc=8
50  ELSE
60    ASSIGN @Hp8711 TO 716
70    Internal=0
80    Isc=7
90    ABORT 7
100   CLEAR 716
110  END IF
120  OUTPUT @Hp8711;"DISP:PROG UPP"1
130  OUTPUT @Hp8711;"DISP:WIND10:GRAP:CLE"2
140  OUTPUT @Hp8711;"SENS2:STAT ON;*WAI"

150  OUTPUT @Hp8711;"ABOR::INIT2:CONT OFF;
:INIT2;*OPC?"1
160  ENTER @Hp8711;Opc
170  OUTPUT @Hp8711;"DISP:WIND10:GRAP:COL 1;
LAB:FONT BOLD"2
180  OUTPUT @Hp8711;"DISP:WIND10:GRAP:
MOVE 45,120;LAB 'HP 8711'"2

190  OUTPUT @Hp8711;"DISP:WIND10:GRAP:
MOVE 30,175;DRAW 30,140;DRAW 480,140;
DRAW 480,175"2
200  OUTPUT @Hp8711;"DISP:WIND10:GRAP:
MOVE 275,140;DRAW 275,130;DRAW 305,130;
DRAW 305,140"2
210  OUTPUT @Hp8711;"DISP:WIND10:GRAP:
MOVE 410,140;DRAW 410,130;DRAW 440,130;
DRAW 440,140"2
220  OUTPUT @Hp8711;"DISP:WIND10:GRAP:LAB:
FONT SMAL"2

230  OUTPUT @Hp8711;"DISP:WIND10:GRAP:
MOVE 250,145;LAB 'RF OUT'"2
240  OUTPUT @Hp8711;"DISP:WIND10:GRAP:
MOVE 395,145;LAB 'RF IN'"2

```

Allocate an IBASIC display partition to show the graphics.

Clear the IBASIC display partition.

Turn on channel 2 for measurements – the upper part of the display is devoted to graphics so the lower part (channel 2) is used for measurements.

Take a single controlled sweep to ensure good data.

Wait for the \*OPC? reply.

Select the bright pen and a bold font.

Draw a label reading 'HP 8711' at 45 pixels to the right and 120 pixels above the origin – the origin is the lower left corner of the current graphics window (the top half of the display).

Draw a box to represent the HP 8711.

Draw a box to represent the RF OUT port.

Draw a box to represent the RF IN port.

Change the text font to small – this is the font the instrument uses to DISP or PRINT things.

Label the RF OUT port.

Label the RF IN port.

```

250 Normalize: !
260  OUTPUT @Hp8711;"DISP:WIND10:GRAP:
    MOVE 290,125;DRAW 290,110;DRAW 425,110;
    DRAW 425,125"2
270  OUTPUT @Hp8711;"DISP:WIND10:GRAP:
    MOVE 1,50;LAB 'Connect THRU between
    RF OUT and RF IN'"2
280  IF Internal=1 THEN
290      ON KEY 1 LABEL "NORMALIZE" RECOVER Norm
300  ELSE
310      Keycode=-1
320      OUTPUT @Hp8711;"DISP:MENU:KEY1
            'NORMALIZE'"
330      OUTPUT @Hp8711;"*CLS;*ESE 0"
340      OUTPUT @Hp8711;"STAT:PRES;DEV:ENAB 1;
            *SRE 4"
350      OUTPUT @Hp8711;"SYST:KEY:QUE:CLE"
360      ON INTR Isc,5 RECOVER Srq
370      ENABLE INTR Isc;2
380  END IF
390  OUTPUT @Hp8711;"DISP:WIND10:GRAP:BUFF OFF"2
400  LOOP
410  OUTPUT @Hp8711;"DISP:WIND10:GRAP:
    MOVE 55,18;LAB '>>>> Press NORMALIZE
    <<<<'"2

```

Draw a THRU connection between the RF OUT and RF IN ports.

Prompt the operator to connect the THRU.

If using the internal IBASIC controller:  
Use the simple ON KEY statement to handle the user interface.

If using an external controller:  
Set a flag for checking on keyboard interrupts.

Label the softkey.

Clear the event status registers.

Preset the status registers, this effects the enable, positive and negative transition registers. Enable the Device Status register to report to the Status Byte on positive transition of bit 0 (Key Pressed). Enable the Status Byte to generate an interrupt when the Device Status register's summary bit reports in.

Clear the key queue to make sure previous key presses don't generate an interrupt.

This branching must be done before the interrupt is enabled. It provides the path for the program to take when an interrupt occurs.

Enable an interrupt to occur when the service request is received.

Turn off the graphics buffer (this is not necessary for the program to run – the purpose is to avoid the error message when the buffer fills up during the following loop).

Wait for a response – flash a message “Press Normalize” on the screen as a prompt. Note there is no exit from this loop other than a keyboard interrupt.

```

420      WAIT .2
430      OUTPUT @Hp8711;"DISP:WIND10:GRAP:
        MOVE 55,18;LAB '           Press NORMALIZE
        ,,,2
440      WAIT .2
450      END LOOP
460 Norm: !
470      IF Keycode<>0 THEN GOTO Normalize
480      OFF KEY
490      OUTPUT @Hp8711;"DISP:WIND10:GRAP:BUFF ON"2
500      OUTPUT @Hp8711;"DISP:WIND10:GRAP:COL 0;
        MOVE 55,18;LAB '>>>>  Press NORMALIZE
        <<<<'2
510      OUTPUT @Hp8711;"DISP:WIND10:GRAP:
        MOVE 1,50;LAB 'Connect THRU between
        RF OUT and RF IN'2
520      OUTPUT @Hp8711;"DISP:MENU:KEY1
        ,,""
530      OUTPUT @Hp8711;"CALC2:MATH (IMPL)"
540      OUTPUT @Hp8711;"INIT2;*WAI"
550      OUTPUT @Hp8711;"TRAC CH2SMEM,CH2SDATA;*WAI"
560      OUTPUT @Hp8711;"CALC2:MATH (IMPL/CH2SMEM)"
570      OUTPUT @Hp8711;"DISP:WIND2:TRAC1 ON;
        TRAC2 OFF"
580      OUTPUT @Hp8711;"DISP:WIND10:GRAP:
        MOVE 290,110;DRAW 425,110;DRAW 425,125;
        COL 1"2
590      STOP
600      !
610 Srq: !
620      Stb=SPOLL(@Hp8711)
630      IF BINAND(Stb,4)<>0 THEN
640          OUTPUT @Hp8711;"STAT:DEV:EVEN?"
650          ENTER @Hp8711;Dev_event

```

If the wrong key was pressed return to the previous message and loop.

The THRU should now be connected and ready to measure.

Turn the graphics buffer back on now that the loop is past.

Select the eraser (pen color 0) and erase the prompts.

Display the active data trace only – turn off any previous normalization.

Take a single sweep on channel 2.

Copy the new data trace into the memory array.

Normalize – display the active data relative to the memory trace.

Display only one of the traces (the normalized trace).

Erase the thru connection and select the bright pen (color 1).

This is the end.

This is the branching routine that handles service request generated interrupts.

Do a serial poll to find out if the analyzer generated the interrupt.

Determine if the Device Status register's summary bit (bit 2 of the Status Byte) has reported in.

If it has, read the Device Status event register.

660        IF BINAND(Dev_event,1)<>0 THEN	Determine if the event was a key press (bit 0 of the Device Status register).
670        OUTPUT @Hp8711;"SYST:KEY?"	If yes, query the analyzer what key has been pressed.
680        ENTER @Hp8711;Keycode	
690        END IF	
700        END IF	
710        ENABLE INTR I <sub>sc</sub>	Re-enable the interrupt in case the wrong key was pressed.
720        GOTO Norm	The thru is now connected – return to that part of the routine and do it.
730        END	This is really the end.

1 This command is only available on instruments that have the IBASIC (1C2) option.

2 Window 10 is the IBASIC display window. Instruments without the IBASIC option (1C2) cannot draw  
to this window. The other graphics windows (1 – channel 1, and 2 – channel 2) are available on all  
instruments. Turning off the data trace and graticule on a measurement channel that is not being used  
results in a clearly visible graphics viewing area.

## **CALKIT — Instrument state file for downloading user-defined calibration kit definitions.**

This instrument state file demonstrates the type of file required to download user-defined calibration kits. To see an example of using this feature, refer to “To Calibrate with Other Cal Kits” in chapter 6 of the *HP 8711 Operating and Programming Manual*.

```
10 !$ Standard Definitions for HP 85054B Precision Type-N Cal Kit.
20 !
30 !$ Definitions for 50 Ohm jack (FEMALE center contact) test
40 !$ ports, plug (MALE center contact) standards.
50 !
60 ! OPEN: $ HP 85054-60027 Open Circuit Plug
70 !     ZO 50.0 $ Ohms
80 !     DELAY 57.993E-12 $ Sec
90 !     LOSS 0.8E+9 $ Ohms/Sec
100 !    CO 88.308E-15 $ Farads
110 !    C1 1667.2E-27 $ Farads/Hz
120 !    C2 -146.61E-36 $ Farads/Hz^2
130 !    C3 9.7531E-45 $ Farads/Hz^3
140 !
150 ! SHORT: $ HP 85054-60025 Short Circuit Plug
160 !     ZO 50.0 $ Ohms
170 !     DELAY 63.078E-12 $ Sec
180 !     LOSS 8.E+8 $ Ohms/Sec
190 !
200 ! LOAD: $ HP 00909-60011 Broadband Load Plug
210 !     ZO 50.0 $ Ohms
220 !     DELAY 0.0 $ Sec
230 !     LOSS 0.0 $ Ohms/Sec
240 !
250 ! THRU: $ HP 85054-60038 Plug to Plug Adapter
260 !     ZO 50.0 $ Ohms
270 !     DELAY 196.0E-12 $ Sec
280 !     LOSS 2.2E+9 $ Ohms/Sec
290 !
300 END
```

**PAGINATION RECORD for PUBLICATION OF: HP-IB Command Reference For the HP  
8711 RF Network Analyzer**  
**PART NUMBER:08711-90110**  
**PRINT DATE: July 1993**  
**PREPARED ON: 6/28/93**

		Title Page (gray index)	9b-1	9b-3	9b-5	9b-7
		copyright page (gray index)	9b-2	9b-4	9b-6	9b-8

9b-9	9b-11	9b-13	9b-15	9b-17	9b-19	9b-21
9b-10	9b-12	9b-14	9b-16	9b-18	9b-20	9b-22

9b-23	9b-25	9b-27	9b-29	9b-31	9b-33	9b-35
9b-24	9b-26	9b-28	9b-30	9b-32	9b-34	9b-36

9b-37	9b-39	9b-41	9b-43	9b-45	9b-47	9b-49
9b-38	9b-40	9b-42	9b-44	9b-46	9b-48	9b-50

9b-51	9b-53	9b-55	(gray index)			
9b-52	9b-54	9b-56	(gray index)			

08711-90121

Model 8711

filename 08711-90121

11/20/11



---

## User's Guide

---

**HP 8711A/11B/12B/13B/14B  
Option H20/K20  
Fault Location Software**

HP part number: 08711-90121  
Printed in USA      March, 1995

**Notice.** The information contained in this document is subject to change without notice. Hewlett-Packard makes no warranty of any kind with regard to this material, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

©Copyright Hewlett-Packard Company 1994, 1995  
All Rights Reserved. Reproduction, adaptation, or translation without prior  
written permission is prohibited, except as allowed under the copyright laws.  
1400 Fountaingrove Parkway, Santa Rosa, CA 95403-1799, USA

---

# Hewlett-Packard Software Product License Agreement and Limited Warranty

**Important**

Please carefully read the License Agreement (located in Chapter 5) before opening the media envelope or operating the equipment. Rights in the software are offered only on the condition that the Customer agrees to all terms and conditions of the License Agreement. Opening the media envelope or operating the equipment indicates your acceptance of these terms and conditions. If you do not agree to the License Agreement, you may return the unopened package for a full refund.

---

## Introduction

The Option H20/K20 Fault Location Software is a set of programs designed to run on any of the following network analyzers:

HP 8711A

HP 8711B

HP 8712B

HP 8713B

HP 8714B

If you ordered option H20, you received a network analyzer and the fault location software. If you ordered option K20, you already own one of the above-mentioned analyzers and you received a kit consisting of the fault location software and two firmware disks (HP 8711A only – see note below). To run the fault location software, your analyzer must have IBASIC (Option 1C2) installed.

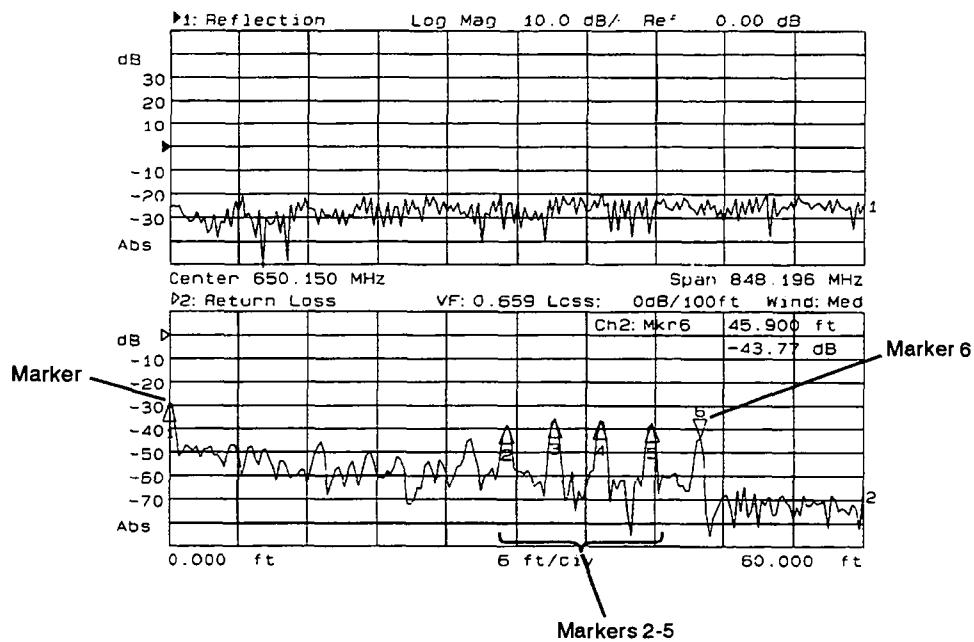
**NOTE**

If you have ordered HP 8711A option K20 (you already had an HP 8711A and you just ordered the fault location software) you will need to load the new firmware that was provided with your shipment before running the software. Refer to Appendix A to perform firmware loading.

The fault location software can help you test and troubleshoot 50 ohm or 75 ohm transmission lines. The software uses the analyzer to perform swept frequency measurements of return loss over the frequency ranges of 300 kHz to 1300 MHz (for HP 8711A/11B/13B) and 300 kHz to 3000 MHz (for HP 8712B/14B). Digital signal processing is performed on the result to convert frequency domain measurements to the distance domain. In the default mode, a split screen display is used, showing both return loss (in dB) versus frequency and return loss (in dB) versus distance. The display units can also be expressed in SWR or reflection coefficient.

The following illustration depicts a typical measurement display. The upper display shows the measurement in the frequency domain. The bottom display shows the response in the distance domain. Six markers identify mismatches along the cable.

- Marker 1 identifies the cable connector.
- Markers 2, 3, 4, and 5 identify barrel connectors separated by 4 feet.
- Marker 6 identifies a 50 ohm termination.



md61ab\_c

---

# Documentation Outline

This guide contains five chapters:

Getting Started	Contains information on how to run the software and how to use the program's main menu.
Making Measurements	Contains basic fault location measurement theory, how to make and interpret measurements, and example measurements.
Key Reference	Contains general key functions of front-panel and softkeys. This reference section is arranged alphabetically by front-panel key (hardkey).
Performance Summary and Characteristics	Contains a summary of software performance and distance range and resolution data.
License Agreement and Limited Warranty	Contains the Hewlett-Packard software license agreement and limited warranty information. Read this chapter before opening the media envelope.
Appendices	Contain information on how to load firmware (for HP 8711A Option K20), a table with typical values of cable loss and velocity factor, in-depth measurement theory, and external programming information.

**Key Conventions.** This manual uses the following conventions:

- |                        |   |
|------------------------|---|
| <b>Front-Panel Key</b> | This represents a key physically located on the instrument (a "hardkey").   |
| <b>Softkey</b>         | This indicates a "softkey," a key whose label is determined by the instrument's firmware, and is displayed on the right side of the instrument's screen next to the eight unlabeled keys. |

---

## Contents

<b>1. Getting Started</b>	
Running the Software . . . . .	1-3
To Run the Software Using the AUTOST Program . . . . .	1-4
To Load the FAULT Program From the Front Panel . . . . .	1-5
To Run the Program After It Has Been Loaded . . . . .	1-5
If an Error Message Appears . . . . .	1-6
The Program Main Menu . . . . .	1-7
Default Setup . . . . .	1-8
Guided Setup . . . . .	1-9
Setup Diagram . . . . .	1-11
PAUSE . . . . .	1-11
Exiting the Program . . . . .	1-12
<b>2. Making Measurements</b>	
Basic Fault Location Measurement Theory . . . . .	2-3
How to Make and Interpret Measurements . . . . .	2-4
Enter the Measurement Parameters . . . . .	2-5
Calibrate the Instrument . . . . .	2-6
Connect the Equipment . . . . .	2-7
Interpret the Measurement . . . . .	2-8
Example Measurements . . . . .	2-11
Example 1: Identify Mismatches Expressed as Return Loss . . . . .	2-12
Example 2: Identify Mismatches as the Magnitude of Reflection . . . . .	2-14
Example 3: Identify Mismatches at Greater Distances . . . . .	2-16
<b>3. Key Reference</b>	
Menumaps . . . . .	3-3
MEAS . . . . .	3-4
SOURCE . . . . .	3-5
CONFIGURE . . . . .	3-6
SYSTEM . . . . .	3-7
Alphabetical Reference . . . . .	3-8
AVG . . . . .	3-9

<b>CAL</b>	3-10
<b>CHAN 1</b>	3-13
<b>CHAN 2</b>	3-14
<b>DISPLAY</b>	3-15
<b>FORMAT</b>	3-16
<b>FREQ</b>	3-17
<b>HARD COPY</b>	3-18
<b>MARKER</b>	3-19
<b>MENU</b>	3-21
<b>POWER</b>	3-22
<b>SAVE RECALL</b>	3-23
<b>SCALE</b>	3-24
<b>SWEET</b>	3-26
<b>SYSTEM OPTIONS</b>	3-27

<b>4. Performance Summary and Characteristics</b>	
General Performance Characteristics	4-3
Distance Range and Resolution	4-4
Range	4-4
Resolution	4-4
Typical Distance Data in Feet	4-5
Distance Data in Meters	4-8
<b>5. License Agreement and Limited Warranty</b>	
License Agreement	5-3
Use	5-3
Copies and Adaptations	5-3
Ownership	5-4
Transfer of Rights in Software	5-4
Sublicensing and Distribution	5-5
Termination	5-5
Updates and Upgrades	5-5
Export	5-5
Limited Warranty	5-6
Software	5-6
Media	5-6
Notice of Warranty Claims	5-7
Limitation of Warranty	5-7
Exclusive Remedies	5-7
Warranty Service	5-7

<b>A. HP 8711A Option K20: Installing the Firmware</b>	
Step 1. Format the Blank Disk . . . . .	A-3
Step 2. Store the Analyzer's Correction Constants To the Disk	A-4
Step 3. Determine the Correct Firmware for Your Analyzer	A-5
Step 4. Load the New Firmware . . . . .	A-6
Step 5. Restore the Correction Constants . . . . .	A-7
<b>B. Cable Loss and Velocity Factor Table</b>	
<b>C. Measurement Theory</b>	
How the Analyzer Converts Frequency Data to Distance	
Data . . . . .	C-3
Start/Stop Distance and Frequency Span Explanation . .	C-4
<b>D. External Programming</b>	
<b>Index</b>	

---

## Figures

2-1. Fault Response to an RF Signal . . . . .	2-3
2-2. Calibrate the Instrument . . . . .	2-6
2-3. Basic Fault Location Measurement Setup . . . . .	2-7
2-5. Channel 2 Display . . . . .	2-9
2-8. Example 1: Identify Mismatches Expressed as Return Loss . . . . .	2-13
2-9. Example 2: Identify Mismatches as the Magnitude of Reflection Coefficient . . . . .	2-15
2-11. Example 3: Identify Mismatches at Greater Distance . . . . .	2-17
2-12. Example 3: Zoom In On Suspected Mismatches . . . . .	2-18

---

## Tables

1-1. Default Setup Parameters . . . . .	1-8
2-1. Guided Setup Parameters Access Keys . . . . .	2-5
4-1. Fault Location Distance Range or Maximum Distance (in feet) Versus Resolution at 201 Points* . . . . .	4-5
4-2. Fault Location Distance Range or Maximum Distance (in meters) Versus Resolution at 201 Points* . . . . .	4-8
B-1. Cable Loss and Velocity Factor Table * . . . . .	B-2
B-1. Cable Loss and Velocity Factor Table (continued) . . . . .	B-3

---

# 1

---

## Getting Started

---

## Getting Started

This chapter will step you through running the fault location software and describe the main menu options available to you after the software is running. Your analyzer must have Option 1C2, IBASIC, installed in order to run the fault location software. To verify that option 1C2 is installed on your analyzer, press **SYSTEM OPTIONS** | **Service** | **Instrument Info** on the analyzer. If the IBASIC option is installed you will see the following message on the display:

**IBASIC Option (1C2):      Installed**

If IBASIC has not been installed on your analyzer, you can order an IBASIC upgrade kit. See the table below for the model/part number to order for your particular analyzer. Contact the nearest Hewlett-Packard sales or service office for information and assistance.

Your Analyzer	IBASIC Kit Model Number	IBASIC Kit Part Number
HP 8711A	HP 86224A	08711-60061
HP 8711B	HP 86224B	08711-60068
HP 8712B	HP 86224B	08711-60068
HP 8713B	HP 86224B	08711-60068
HP 8714B	HP 86224B	08711-60068

**NOTE**

If you have ordered HP 8711A option K20 (you already had an HP 8711A and you just ordered the fault location software) you will need to load the new firmware that was provided with your shipment before running the software. Refer to Appendix A to perform firmware loading.

---

## Running the Software

To run the software you will need the software disk (HP part number 08711-10013). The software disk contains the following three programs:

<b>FAULT</b>	Measures return loss, SWR, or the magnitude of the reflection coefficient simultaneously versus distance and frequency.
<b>AUTOST</b>	Automatically loads at power up with the software disk installed into the disk drive or if located on the analyzer's internal non-volatile RAM disk.
<b>FAULTEXT</b>	This program runs on an external RMB controller or PC. The program FAULT is downloaded to the analyzer and is run by FAULTEXT. Within FAULTEXT, standard SCPI commands are used to communicate parameter changes to the analyzer running the FAULT application. This program demonstrates automated measurement changes.

**NOTE**

The "FAULT" and "AUTOST" programs are identical. They have been renamed for convenience. It is always a good practice to make a backup copy of your software and keep it in a safe place. Then, in the event that the original is damaged or lost, you will still have a working copy of the software.

## To Run the Software Using the AUTOST Program

When you turn on the analyzer, it first searches the internal non-volatile memory and then the 3.5" disk drive for a program named AUTOST, and then runs it.

The easiest way to run the application software is to just insert the fault location software disk into the analyzer's 3.5" disk drive and cycle the power on the analyzer. The program will automatically load and run. The analyzer will immediately be ready to make fault location measurements.

The program can also be automatically run if it has been copied to the analyzer's internal non-volatile memory. To do this, insert the fault location software disk into the analyzer's 3.5" disk drive and complete the following procedure:

1. Press **SAVE RECALL**, **Select Disk**, **Internal 3.5" Disk**.
2. Use the analyzer's front panel knob to select the AUTOST program.
3. Select **Prior Menu**, **File Utilities**, **Copy File**,  
**Copy to NonVol RAM Enter**.

Once the AUTOST program has been copied to the analyzer's internal non-volatile memory, it will automatically start the fault location program every time the power is turned on.

---

## To Load the FAULT Program From the Front Panel

If you do not wish to cycle the power on the analyzer to load the program, you can load and run the fault location software by performing the following steps:

1. Insert the fault location software disk into the analyzer's disk drive.
2. Press **SAVE RECALL**, **Select Disk**, **Internal Disk**.
3. Use the analyzer's front panel knob to select the FAULT program.
4. Press **Prior Menu**, **Programs**, **Recall Program**, **Run**.

---

## To Run the Program After It Has Been Loaded

After the fault location software has been loaded into the analyzer via either of the above procedures, it can be run from the front panel by pressing **SYSTEM OPTIONS**, **IBASIC**, **Run** or **Continue**. Pressing **Run** will rerun the program from scratch and reinitiate the analyzer and its settings. If you have paused the program with the **PAUSE** softkey, pressing **Continue** will resume the program from where it was paused.

---

**CAUTION**

Changes made to the instrument state while the program is paused may result in unspecified behavior. Press **Run** to restart the program and reset the instrument to default fault-location application parameters.

To return the instrument to normal network analyzer operation, press **PRESET**.

## If an Error Message Appears

If an error message appears when you attempt to run the software, the most likely cause is improper firmware.

If you have ordered HP 8711A Option K20 (software and firmware), be sure that the proper firmware has been installed. Refer to Appendix A.

---

## The Program Main Menu

After the software has initialized and the program is running, there are four softkey options available to you:

- **Default Setup**
- **Guided Setup**
- **Setup Diagram**
- **PAUSE**

To access this menu at any time when the fault location software is running, press **BEGIN**.

## Default Setup

Pressing this softkey returns the instrument settings to the fault location program's initial or default state.

When the fault location program is initially run, the analyzer is automatically set to the default state. The default state for the program measures return loss versus frequency on Channel 1 and return loss versus distance on Channel 2.

Other default settings are:

**Table 1-1. Default Setup Parameters**

Parameter	Default Value
Center frequency	650.150 MHz
Stop distance	100 feet
Start distance	0 feet
Velocity factor	0.659
Cable loss	0 dB/100 feet

---

## Guided Setup

When the **Guided Setup** key is pressed, the instrument will prompt you for the basic information required for your measurement, ask if you want to calibrate or not, and show the basic equipment setup diagram. You will be asked to enter the following parameters:

- Center frequency
- Stop distance
- Start distance
- Velocity factor
- Cable loss

You must terminate each entry with either the **ENTER** hardkey or the **Enter** softkey. If you wish to retain the default value for any particular parameter (see Table 1-1) just press **ENTER** or **Enter** instead of inputting a value.

Next, you will be asked whether you want to calibrate the instrument or not. See "Calibrate the Instrument" in Chapter 2 for more information on calibration. To perform an instrument calibration, you must have type-N open, short and load standards, such as those found in Hewlett-Packard calibration kits model numbers HP 85032E (50 ohm) and HP 85036E (75 ohm). If you wish to calibrate, press **1 ENTER** and follow the prompts. Otherwise, just press **ENTER** and the instrument will then show the basic equipment setup diagram on the display.

After your equipment is set up, press **ENTER** or **Enter** to view the measurement screen. At this time you can use markers or limit lines to interpret your measurement. See "Interpret the Measurement" in Chapter 2.

Getting Started

**The Program Main Menu**

**NOTE**

As changes are made to center frequency, stop distance, start distance and velocity factor, a pop-up message will temporarily appear with the following message:

**Changing Frequency Settings...**

**WARNING: Cable Cal May Be Invalid.**

**Please confirm Cable Loss and  
Velocity Factor.**

Frequency span settings are dependent on center frequency, stop and start distance, and velocity factor. Refer to Appendix B for a table of typical loss and velocity factors or refer to information provided by the manufacturer of your cable. Refer to Appendix C for an explanation of the relationship between frequency span and distance range.

---

## Setup Diagram

When this softkey is pressed, the instrument will show the basic equipment setup diagram on the display. Press **ENTER** or **Enter** to return to the normal display.

---

## PAUSE

The program can be paused at any time by pressing the **PAUSE** softkey. This softkey is available at all times when running the fault location software. To resume the program, press **SYSTEM OPTIONS**, **IBASIC**, **Continue**.

---

### CAUTION

Changes made to the instrument state while the program is paused may result in unspecified behavior. Press **Run** to restart the program and reset the instrument to default fault location application parameters.

---

## Exiting the Program

The program can be exited, and the analyzer returned to normal operation by pressing the **PRESET** key.

---

## Making Measurements

---

## Making Measurements

This chapter contains the following information:

- basic fault location measurement theory
- how to make and interpret measurements
- example measurements

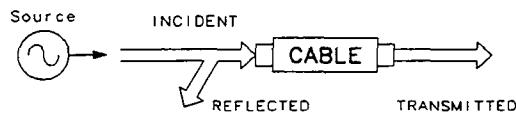
---

## Basic Fault Location Measurement Theory

The fault location software was designed to quickly and easily locate faults, or discontinuities, in either 50 ohm or 75 ohm transmission lines. Refer to Figure 2-1 for the following discussion.

The network analyzer has an RF signal source that produces an incident signal that is used as a stimulus to locate and measure discontinuities in your transmission line or cable. Each fault or discontinuity responds by reflecting a portion of the incident signal and transmitting the remaining signal.

The instrument measures the frequency response of the cable and then transforms the frequency data to distance data; for more measurement theory, refer to Appendix C.



md610

**Figure 2-1. Fault Response to an RF Signal**

The measurement results can be expressed in one of three ways:

<b>Return Loss (RL)</b>	The number of dB that the reflected signal is below the incident signal. Its relationship to the reflection coefficient ( $\rho$ ) is described by the following formula: $RL = -20 \log \rho$ .
<b>Reflection Coefficient (<math>\rho</math>)</b>	The ratio of the reflected voltage wave to the incident voltage wave.
<b>Standing Wave Ratio (SWR)</b>	Any two waves traveling in opposite directions (the incident and reflected for example) cause a "standing wave" to be formed on the transmission line. SWR is defined as the maximum voltage over the minimum voltage of the standing wave. SWR can also be mathematically derived from the reflection coefficient ( $\rho$ ) with the following formula: $SWR = \frac{1 + \rho}{1 - \rho}$

---

## How to Make and Interpret Measurements

After the program is running (see Chapter 1), a typical measurement consists of the following steps:

1. Enter the measurement parameters.
2. Calibrate the instrument.
3. Connect the equipment.
4. Interpret the measurement.

The next few pages explain how to perform each of these steps.

## Enter the Measurement Parameters

Once the program is running and the equipment is set up, use the **Guided Setup** function to enter parameters for your measurement. The guided setup function will prompt you to enter the following parameters:

- Center frequency
- Stop distance
- Start distance
- Velocity factor
- Cable loss

You may also use the instrument hardkeys and softkeys to change instrument settings.

Table 2-1 shows which keys to press to enter the parameters that are asked for in the guided setup. Other parameters may be entered using the instrument keys and are discussed in Chapter 3, "Key Reference."

Table 2-1. Guided Setup Parameters Access Keys

Parameter	Access Keys
Center frequency	[FREQ], <b>Center Freq</b>
Stop distance	[MENU] (or [SCALE]), <b>Stop Distance</b>
Start distance	[MENU] (or [SCALE]), <b>Start Distance</b>
Velocity factor	[CAL], <b>Velocity Factor</b>
Cable loss	[CAL], <b>Cable Loss</b>

### NOTE

Inspect or measure your cable for velocity factor and cable loss per 100 feet specifications. If your cable is not marked with these parameters, you may refer to Table B-1 in Appendix B for some typical numbers. The numbers in Table B-1 are typical only.

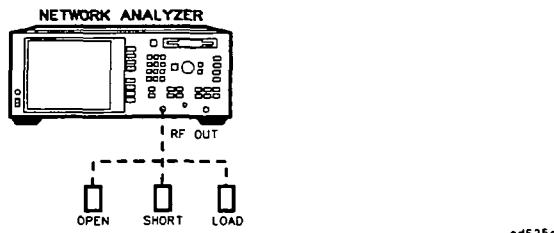
## Calibrate the Instrument

When practical, a calibration should be done at the RF OUT port using open, short, and load calibration standards. If you are using the guided setup function, the instrument will prompt you to perform a calibration. This requires type-N calibration standards. This calibration will correct the instrument and optimize your accuracy. If calibration standards are not available, the default instrument calibration can be used.

**To Use Calibration Standards to Calibrate the**

**Instrument** To calibrate the instrument using calibration standards, follow the prompts in the guided setup function, or perform the following steps:

1. Press **CAL**, **Calibrate Instrument**.
2. Follow the prompts on the analyzer's screen to connect the open, short and load to the instrument's RF OUT port. (See Figure 2-2.)



**Figure 2-2. Calibrate the Instrument**

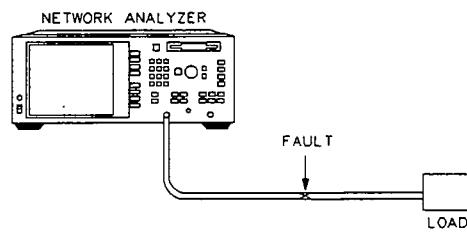
**To Use the Default Calibration**

To use the instrument default calibration, press **CAL**, **Reset Cal to Default**.

---

## Connect the Equipment

The basic equipment setup for fault location measurements is illustrated in Figure 2-3. This diagram is automatically displayed when using the guided setup function, or it can be displayed at any time by pressing **BEGIN Setup Diagram**.



md62a

**Figure 2-3. Basic Fault Location Measurement Setup**

## Making Measurements

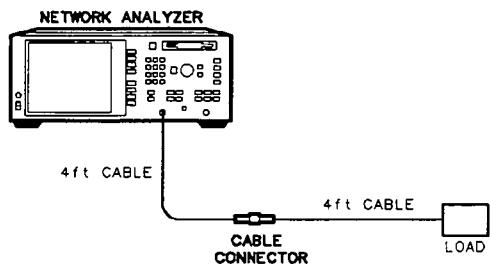
### How to Make and Interpret Measurements

---

## Interpret the Measurement

Once the measurement parameters have been set up, you can use markers to find the peak responses (which indicate faults or discontinuities), or you can place a test limit line on the screen that will indicate whether or not your cable meets a particular specification.

In the following example, two four-foot cables are connected in series to the RF OUT port of the analyzer and are terminated with a 50 ohm load. See Figure 2-4.



md63c

**Figure 2-4.**

Making Measurements  
How to Make and Interpret Measurements

**Using Markers to Find Peak Responses**

Press **MARKER**.

Marker number 1 will automatically be placed on the peak with the highest response. Refer to Figure 2-5. Notice that the display indicates that the response marked by marker number 1 is 4.000 feet from the RF OUT port, and has a return loss of -27.45 dB. Knowing that we connected two four-foot cables together for this example, it can be determined that this discontinuity is created by the connection between the two cables.

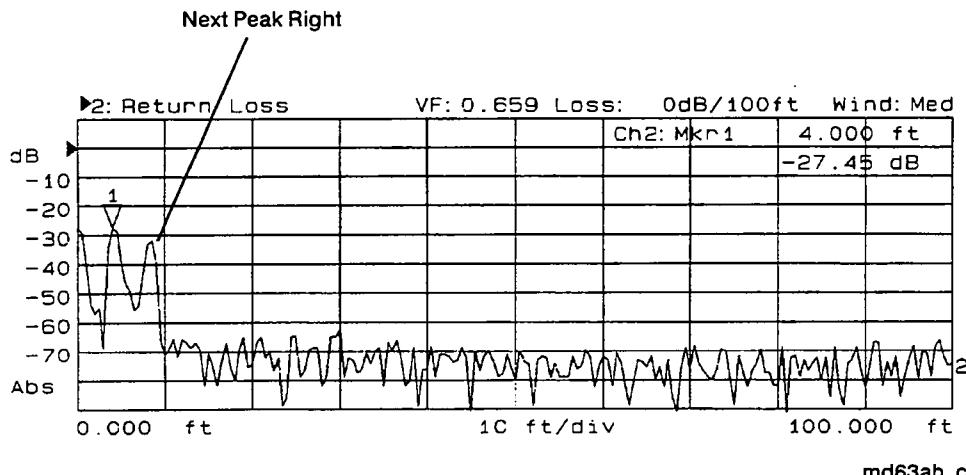


Figure 2-5. Channel 2 Display

Press **Next Peak Right**. When this function is used, the active marker moves to the next highest peak to the right. If you were performing this example, you would notice that marker number 1 has now moved over to the peak indicated as "Next Peak Right" in Figure 2-5. You would also notice that the display would now indicate that the response marked by marker number 1 is 8.000 feet from the RF OUT port. Knowing that we connected two four-foot cables for this example, it can be determined that this response is created by the termination at the end of the two cables.

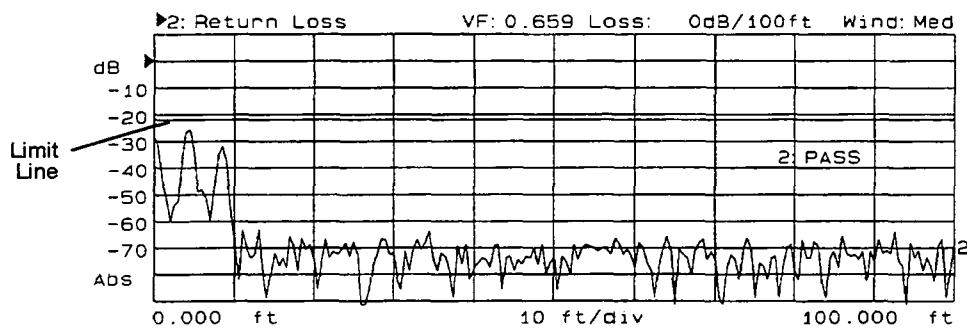
See **MARKER** in Chapter 3 for more information on using markers.

Making Measurements  
**How to Make and Interpret Measurements**

**Using a Limit Line For  
Testing a Cable**

In this example, we have a transmission line return loss specification of  $< -22$  dB. To set the limit line to  $-22$  dB, press **SYSTEM OPTIONS** **Limit Line** **-22** **ENTER**.

Refer to Figure 2-6. Since all responses in this example fall below the test limit line, the display indicates "PASS."



md84ab\_c

**Figure 2-6.**

Press **Limit Line** **-28** **ENTER**.

In this example, setting the limit line to  $-28$  dB return loss would cause the display to indicate "FAIL." This is due to the response at 4 feet exceeding the  $-28$  dB limit. In a real test situation you could quickly and easily determine whether or not your cable meets your specification by correctly setting the limit line to your return loss specification and just looking for the "PASS" or "FAIL" indicator. With a failure you might want to activate a marker (or markers) to determine the magnitude and location of any out-of-specification responses.

---

## Example Measurements

This section contains three measurement example displays with an explanation of the measurement results. These examples may be reproduced by using the test setup diagrams, procedures and explanations with each example.

**NOTE**

Some of the illustrations with the following examples were created by using the **HARD COPY**  
**Plot HPGL to Disk** function. (See **HARD COPY** in Chapter 3.) Whenever the markers are turned on, this marker table becomes part of the graphic that is plotted to the disk.

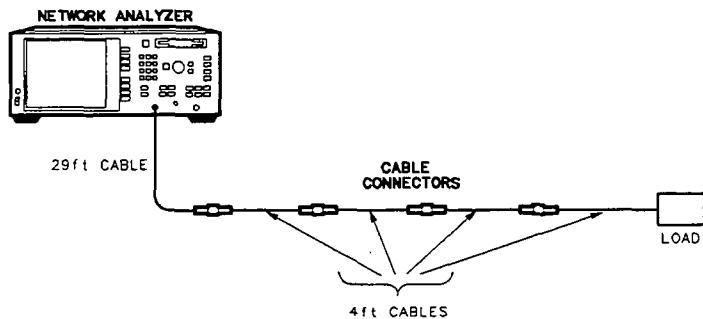
## Making Measurements

### Example Measurements

---

### Example 1: Identify Mismatches Expressed as Return Loss

1. Set up the equipment as shown in Figure 2-7.



md64c

Figure 2-7.

2. Press **BEGIN Default Setup**.
3. Press **SCALE Stop Distance 60 ENTER**.
4. Press **MARKER** and then use the front panel knob and/or up and down arrow keys to place marker 1 all the way to the left side of the display as in Figure 2-8.
5. Press **2** to activate marker 2 and use the front panel knob, the up and down arrow keys, the **Next Peak Left**, or the **Next Peak Right** functions to place marker 2 on the next response as shown in Figure 2-8.
6. Repeat the previous step for markers 3 through 6.

In this example, the first marker displays the return loss at the cable connector. Markers 2, 3, 4, and 5 identify barrel connectors separated by 4 feet. Marker 6 identifies the 50 ohm termination.

Making Measurements  
Example Measurements

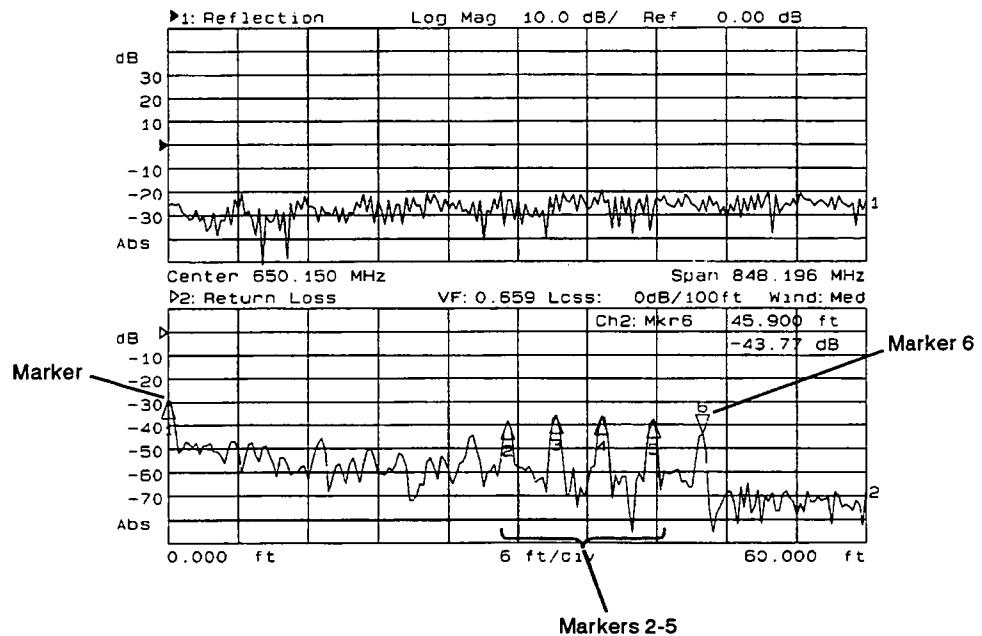


Figure 2-8. Example 1: Identify Mismatches Expressed as Return Loss

## Example 2: Identify Mismatches as the Magnitude of Reflection

### Coefficient

For this example, use the same equipment setup and steps to place markers on responses as instructed in example 1.

Press the following keys:

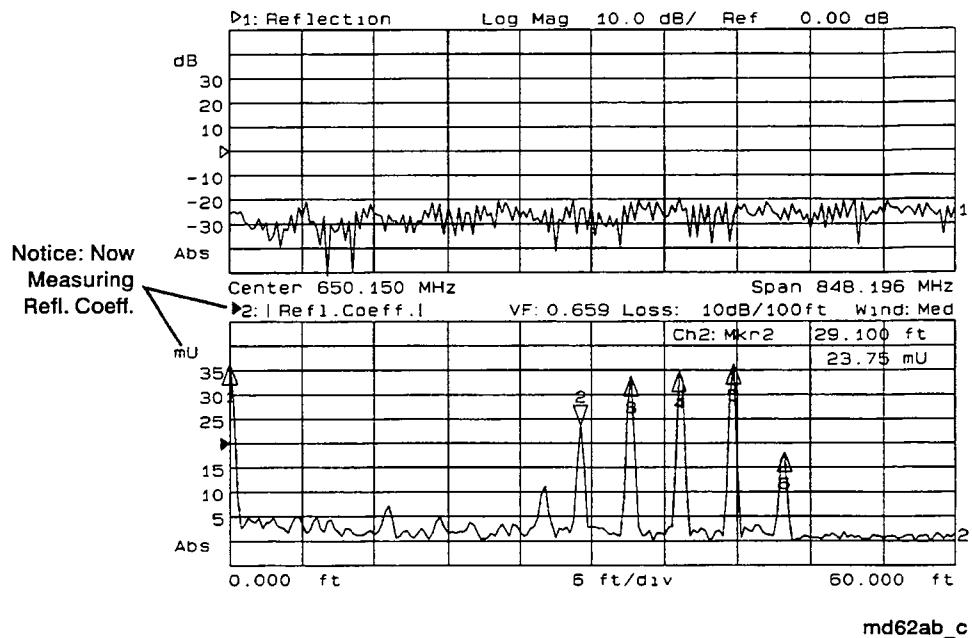
**CHAN 2 Refl. Coeff. Mag.**

**SCALE Autoscale**

This example is very similar to example 1 except that the responses are measured in terms of reflection coefficient rather than return loss. **Autoscale** is used to optimize the viewing of the data trace within the display area.

As in example 1, the response designated by marker 1 identifies the cable connector. Markers 2, 3, 4, and 5 identify barrel connectors separated by 4 feet. Marker 6 identifies the 50 ohm termination.

Making Measurements  
**Example Measurements**



**Figure 2-9. Example 2: Identify Mismatches as the Magnitude of Reflection Coefficient**

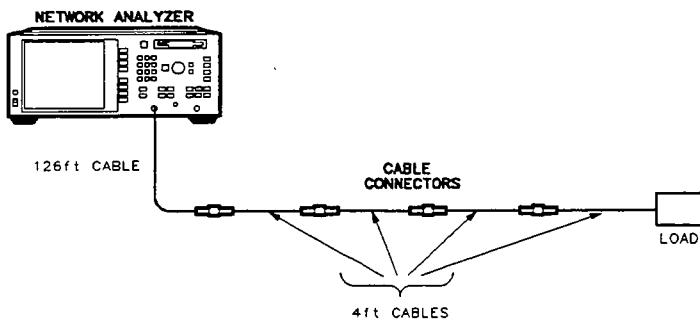
Making Measurements  
**Example Measurements**

---

### Example 3: Identify Mismatches at Greater Distances

In the previous two examples, we were dealing with relatively short lengths of cable ( $\approx$  45 feet). In this example a longer length cable is used.

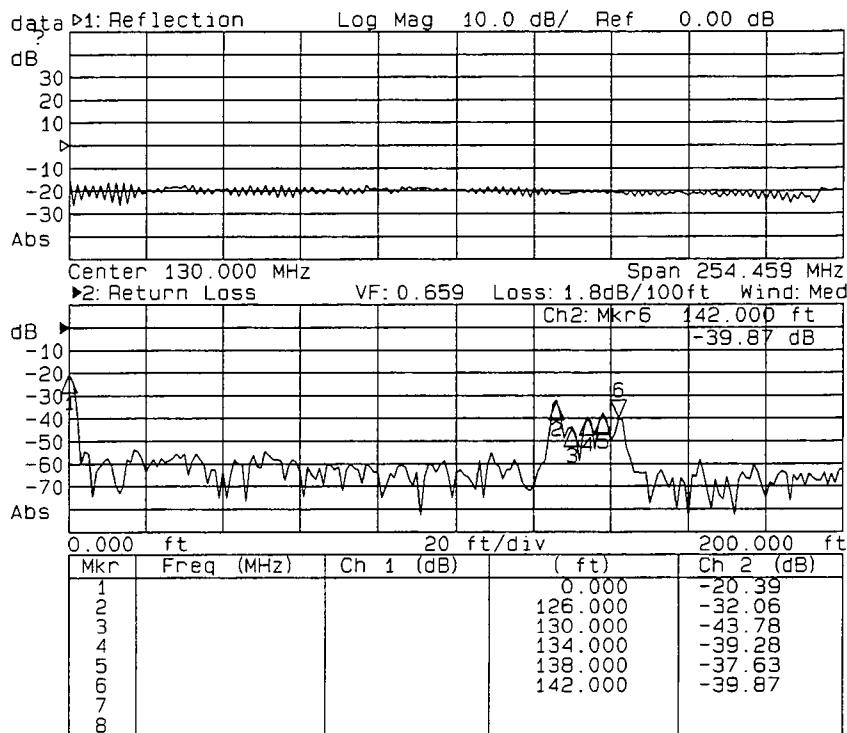
1. Set up the equipment as shown in Figure 2-10.



**Figure 2-10.**

2. Press **BEGIN Default Setup**.
3. Use **Guided Setup** to enter a stop distance of 200 feet, and a cable loss factor of 1.8 dB/100 feet.
4. Use markers to identify the responses. See example 1 for detail on how to place markers.  
The display should now look similar to Figure 2-11.
5. At greater distance the resolution is decreased. In this example, marker 1 identifies the cable connector. Markers 2, 3, 4, and 5 identify barrel connectors. Marker 6 identifies a 50 ohm termination.

Making Measurements  
**Example Measurements**



**Figure 2-11. Example 3: Identify Mismatches at Greater Distance**

## Making Measurements

You can now change the **Stop Distance** and **Start Distance** (in the **SCALE** menu) to zoom in around the responses marked by markers 2 through 6. See Figure 2-12.

Use the **(MARKER)** menu to place markers 1 through 5 on the peak responses. Marker 1 is the first connector. Marker 5 identifies the 50 ohm termination.

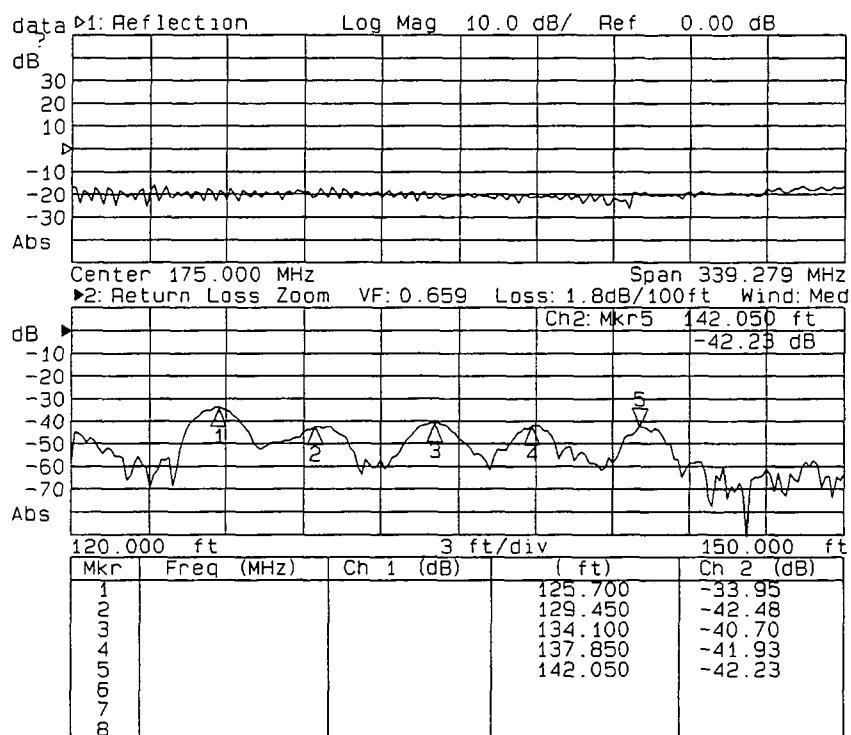


Figure 2-12. Example 3: Zoom In On Suspected Mismatches

**NOTE**

As you change the stop and start distance, a pop-up message will temporarily appear with the following message:

**Changing Frequency Settings...**  
**WARNING: Cable Cal May Be Invalid.**  
**Please confirm Cable Loss and**  
**Velocity Factor.**

Frequency settings are dependent on center frequency, stop and start distance, and velocity factor. Refer to Appendix B for a table of typical loss and velocity factors or refer to information provided by the manufacturer of your cable. Refer to Appendix C for an explanation of the relationship between frequency span and distance range.

---

Key Reference

---

## Key Reference

With the fault location software loaded and running in the analyzer, the hardkeys perform different functions than with normal analyzer operation.

The first part of this chapter contains menu maps showing the softkey functions available under each hardkey. This section is organized by front panel function group.

The second part of this chapter contains an alphabetical reference of each hardkey, along with the functions (softkeys) that are accessed by each key.

---

## Menumaps

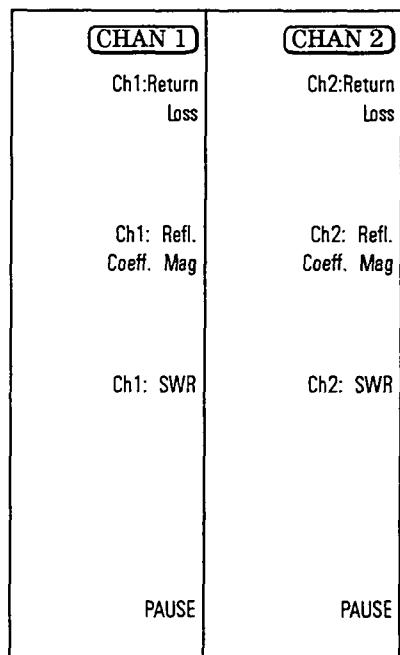
The main softkey functions under each hardkey are depicted on the next few pages. The menumaps in this section are organized by front panel function group.

Pressing a softkey in some cases performs a function, and in other cases may lead to another level of softkeys. Refer to the alphabetical hardkey reference following this section for information on softkey functions, and any lower level softkeys.

**Key Reference**  
**Menumaps**

---

## MEAS



---

## SOURCE

FREQ	SWEEP	POWER	MENU
Center Freq	Continuous	Power Level	101 Points
	Hold		201 Points
	Single		401 Points
			Start Distance
			Stop Distance
PAUSE	PAUSE	PAUSE	PAUSE

**Key Reference  
Menumaps**

## CONFIGURE

SCALE	DISPLAY	CAL	MARKER	FORMAT	AVG
Autoscale	Full Screen	Reset Cal to Default	Active Mkr->Max	Feet	Average Restart
Scale/Div	Split Screen	Calibrate Instrument	Next Peak Right		Average OFF
Reference Level	Marker Screen	Velocity Factor	Next Peak Left	Meters	Average Factor
Reference Position		Cable Loss	Delta Mkr OFF/ON		
Start Distance	Graticule ON/OFF	Calibrate Cable	Active Mkr Off		
Stop Distance		Multi Peak Threshold	All Off		
		Multi Peak Corr OFF/ON	More Mkrs		
PAUSE	PAUSE	PAUSE	PAUSE	PAUSE	PAUSE

---

## SYSTEM

SAVE RECALL	HARD COPY	SYSTEM OPTIONS	
Save Setup/Data	Plot to Default	Window	Minimum
Recall Setup/Data	Plot HPGL to Disk	Limit Line	Medium
Save ASCII Trace Data		Title and Clock On	Maximum
Delete Files		Title and Clock Off	
Format Disk		Change Title	Cancel
PAUSE	PAUSE	PAUSE	PAUSE

---

## Alphabetical Reference

This section is organized alphabetically by hardkey.

**AVG**

Pressing this key displays the average menu.

Averaging reduces random noise by averaging the measurement data from sweep to sweep. In averaging mode, the network analyzer measures each frequency point once per sweep and averages the current and previous trace up to the averaging factor (or number) specified by the user. The instrument computes each data point based on an exponential average of consecutive sweeps weighted by a user-specified averaging factor.

**Average Restart** Restarts averaging of the frequency domain data after it has been turned on by using the **Average OFF/ON** softkey.

**Average** Toggles averaging on and off.  
**OFF/ON**

**Average Factor** Allows entry of new averaging factor. The default is 16 and the maximum is 64.

**NOTE**

Averaging is performed on the frequency domain data only. The distance domain data is not averaged.

**CAL**

Pressing this key displays the calibration menu.

You can improve the accuracy of your measurements with calibration features built into the fault location software. You may correct for directivity, source match, and frequency response with the instrument calibration, or separately correct for cable loss and multiple responses.

**Reset Cal to Default** Selecting this function uses the default instrument calibration.

**Calibrate Instrument** This function guides you through recalibrating the analyzer with open/short/load standards. This calibration will correct for directivity, source match, and frequency response.

**Velocity Factor** This function allows you to enter a new velocity factor. Velocity factor is one of the two required parameters for cable calibration. Velocity factor is the speed of propagation of electrical signals in the cable relative to their speed in a vacuum. The velocity factor of your cable may be imprinted on the cable. If the velocity factor of your cable is not available, you may refer to Table B-1 for some typical values.

**Cable Loss** This function allows you to enter a new cable loss factor. Cable loss is one of the two required parameters for cable calibration. The cable loss of your cable may be imprinted on the cable. If not, you may refer to Table B-1 for some typical values.

**CAUTION**

The values in Table B-1 are typical only. Different manufacturers' cables may vary significantly from the table values.

**Calibrate Cable** This function lets you measure a known length of cable and automatically figures cable loss and velocity factors.

An alternative to entering the velocity factor and cable loss of your cable is to use the algorithm built into the fault location software to determine these values automatically. To use this function, you need a known length of cable of the same type being measured. The cable should be at least 30 feet long.

When this softkey is pressed, you will be prompted to enter the length of your cable and then to connect it, leaving the free end an open circuit. The software will determine values for velocity factor and cable loss based on the known distance down the line where the open circuit appears.

You will then have the option of having the software automatically modify the existing setup with the new values.

**Cable Calibration Note**

Cable loss is nonlinear with frequency. Therefore the apparent loss per 100 feet will change as the frequency range is changed. Velocity factor is also somewhat frequency dependent. You may wish to re-examine these values as the frequency limits are changed.

Key Reference

(CAL)

**Multi Peak Threshold** Allows entry of new multi-peak threshold. The fault location software can automatically compensate for multiple responses along the cable. For example, consider a cable with three successive faults along its length with reflection coefficients of 0.8. Without multi-peak correction, the measured response would be:

Peak	Measured Response	Calculation
Peak #1	0.8	
Peak #2	0.16	$[0.8 * (1 - 0.8)]$
Peak #3	0.032	$[0.8 * ((1 - 0.8) - 0.16)]$

With multi-peak correction, the measured response becomes:

Peak	Measured Response
Peak #1	0.8
Peak #2	0.8
Peak #3	0.8

Each subsequent peak is corrected based on the magnitude of the previous  $n$  peaks. Peak values are searched for above a user-defined threshold.

---

**CAUTION**

There is some uncertainty built into the measurement when multi-peak mode is enabled since the fault location software cannot always distinguish between real responses and response sidelobes which look like responses. To distinguish between side lobes and responses, various window selections should be tried (see SYSTEM OPTIONS), while looking for changes in the responses.

---

**Multi Peak Corr** Turns multi peak correction mode on or off.  
OFF/ON

---

CHAN 1

Pressing this key displays the menu for channel 1 display format.

**Ch1: Return Loss** Displays the channel 1 reflection response in dB. Return loss is the number of dB that the reflected signal is below the incident signal. Return loss can also be defined by the following formula:

$$RL = -20 \log \rho$$

**Ch1: Refl. Coeff. Mag** Displays the channel 1 reflection response as reflection coefficient magnitude. The reflection coefficient is the ratio of the reflected voltage wave to the incident voltage wave.

**Ch1: SWR** Displays the channel 1 reflection response as Standing Wave Ratio. Any two waves traveling in opposite directions (the incident and reflected for example) cause a “standing wave” to be formed on the transmission line. SWR is defined as the maximum voltage over the minimum voltage of the standing wave. SWR can also be mathematically derived from the reflection coefficient ( $\rho$ ) with the following formula:

$$SWR = \frac{1 + \rho}{1 - \rho}$$

Key Reference

---

**CHAN 2**

Pressing this key displays the menu for channel 2 display format.

**Ch2: Return Loss** Displays the channel 2 reflection response in dB. Return loss is the number of dB that the reflected signal is below the incident signal. Return loss can also be defined by the following formula:

$$RL = -20 \log \rho$$

**Ch2: Refl. Coeff. Mag** Displays the channel 2 reflection response as reflection coefficient magnitude. The reflection coefficient is the ratio of the reflected voltage wave to the incident voltage wave.

**Ch2: SWR** Displays the channel 2 reflection response as Standing Wave Ratio. Any two waves traveling in opposite directions (the incident and reflected for example) cause a "standing wave" to be formed on the transmission line. SWR is defined as the maximum voltage over the minimum voltage of the standing wave. SWR can also be mathematically derived from the reflection coefficient ( $\rho$ ) with the following formula:

$$SWR = \frac{1 + \rho}{1 - \rho}$$

---

DISPLAY

Pressing this key accesses the Display menu. These softkeys allow you to customize the display layout to meet your needs.

**Full Screen** Displays both frequency (channel 1) and distance (channel 2) data traces on one full screen.

**Split Screen** Split screen is the default mode. The screen is split into two displays: frequency data (channel 1) is the top screen, and distance data (channel 2) is the bottom screen.

**Marker Screen** Another split screen mode where the top screen is a table containing marker data such as whether a particular marker is in use, and its position and value if it is. In marker screen mode the bottom screen is always channel 2 (distance data).

**Graticule** Turns the display graticule on or off.  
ON/OFF

**Key Reference**

---

**FORMAT**

Allows you to change between displaying distance in feet and meters.

**Feet Displays** distance in units of feet.

**Meters Displays** distance in units of meters.

**FREQ**

Allows entry of center frequency. Center frequency is a parameter that is also asked for in the Guided Setup.

The center frequency should be set within the operating frequency bandwidth of the transmission line being tested. The center frequency can only be set to a value which will not result in the source going beyond its frequency limits.

**Center Frequency** When this softkey is pressed, you will be prompted to enter a new center frequency. You can retain the current center frequency if you wish by just pressing **ENTER**. The fault location software will select the proper frequency span for your measurement.

**See Also**

See Appendix C for an explanation of how the instrument determines the frequency span.

**HARD COPY**

Pressing this key displays the hard copy menu, which allows you to plot your fault location measurement results.

**Plot to Default** Plots the data to the selected output device. See your analyzer's *User's Guide* for information on setting up and configuring a plotter or printer to your analyzer.

**Plot HPGL to Disk** Instead of sending a displayed measurement result to a plotter or printer for a hardcopy output, you can save it as a Hewlett-Packard Graphics Language (HPGL) file to a disk in the network analyzer internal disk drive. The following procedure describes how to plot to a file.

1. Place a formatted disk in the network analyzer disk drive. If your disk is not formatted, press **SAVE RECALL** **Format Disk** to format the disk.
2. Press **HARD COPY** **Plot HPGL to Disk**.

The file is saved with a filename of PLOT#.HGL (where # is a number).

**NOTE**

If any markers are on, the marker table is added to the plot of the screen. See Figure 2-11 in Chapter 2 for an example.

**MARKER**

Pressing this key displays the marker menu and activates marker 1, putting it on the peak response on channel 2. There are eight markers available for use with the fault location software. These markers are only available for use with channel 2 distance data.

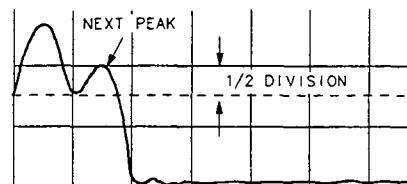
The **MARKER** need not be pressed to activate markers. You can activate a marker at any time by just selecting 1 through 8 on the instrument's keypad.

When a marker is initially selected, it is considered to be "active." Only one marker can be active at any given time. An active marker's triangle will be oriented point-down ( $\nabla$ ) with its corresponding number above the triangle. Any other inactive displayed markers will have their triangles oriented point-up ( $\Delta$ ) with their corresponding numbers below the triangle. The active marker can be manually moved at any time by rotating the instrument's front panel knob or by using the up and down arrow keys.

**Active Mkr-->Max** Places the active marker on the maximum point of the distance display. If all markers are off when this function is selected, it activates marker 1 and places it on the maximum point in the distance display.

**Next Peak Right** Moves the active marker to the next nearest peak to the right. The next peak must be  $> -60$  dB, and the current peak difference to the next peak difference must be at least 1/2 of a vertical division. See Figure 3-1.

**Next Peak Left** Moves the active marker to the next nearest peak to the left. The next peak must be  $> -60$  dB, and the current peak difference to the next peak difference must be at least 1/2 of a vertical division. See Figure 3-1.



md660

**Figure 3-1.**

Key Reference

**MARKER**

**Delta Mkr OFF/ON** Toggles the marker delta function on and off. The delta function makes the active marker the delta marker or reference point. The reference point is indicated by a small delta symbol below the marker icon. In delta marker mode, marker values are the offset values from the delta marker. For example, assume marker 1 is the delta marker: if marker 2 reads 2.5 feet and -9.3 dB, it is 2.5 feet away from and 9.3 dB below marker 1.

**Active Mkr Off** Turns only the active marker off.

**All Off** Turns all markers off.

**More Mkrs** Briefly displays the following message: "The number pad selects markers 1 through 8. To move markers, use the front panel knob or the up and down arrow keys."

**See Also**

For an example of marker usage, see "Interpret the Measurement" in Chapter 2.

## [MENU]

This menu allows you to select the number of data trace points. Fewer trace points provides less resolution but faster sweep speeds and vice versa. This menu also allows entry of the start and stop distance.

**101 Points** Selects 101 data tracepoints.

**201 Points** Selects 201 data tracepoints. This is the default number of tracepoints.

**401 Points** Selects 401 data tracepoints.

**Start Distance** Allows entry of a new start distance. Start distance is a parameter that is also asked for in the Guided Setup. When this softkey is pressed, you will be prompted to enter a new start distance. You can retain the current start distance if you wish by just pressing [ENTER].

**Stop Distance** Allows the entry of a new stop distance. Stop distance is a parameter that is also asked for in the Guided Setup. When this softkey is pressed, you will be prompted to enter a new stop distance. The maximum allowable entry for stop distance is 9999.999 feet or 9999.99 meters. You can retain the current stop distance if you wish by just pressing [ENTER].

**NOTE**

Changing start and/or stop distance will change the frequency span. See Appendix C for detailed information on the relationship between distance range and frequency span.

**POWER**

Pressing this key allows you to change the power level.

**Power Level** Allows you to set the power level at the output port. The default power level is 0 dBm.

**SAVE RECALL**

Pressing this key displays the Save Recall menu.

**Save Setup/Data** Saves the current instrument setup and data to a formatted disk in the internal disk drive. You will be prompted to enter a file name.

**Recall Setup/Data** Recalls setup and data from an internal floppy disk to the display. Use the front panel knob or the up and down arrow keys to select a file to recall.

**Save ASCII Trace Data** Saves trace data in ASCII format. This data is spreadsheet compatible.

**Delete Files** Allows you to delete files from the disk. Use the front panel knob or the up and down arrow keys to select a file to delete.

**Format Disk** Formats a floppy disk in the instrument's disk drive in DOS format.

**SCALE**

Pressing this key displays the scale menu.

**Autoscale** Scales the data trace vertically to fit within the graticule area of the display on the active channel only.

**Scale/Div** Allows you to enter the appropriate desired units per vertical display division for the active channel. For example, if scale/div is 10 dB, each graticule line is 10 dB higher than the one below it.

**Reference Level** Allows you to set the value of the reference line on the active channel. The line is indicated on the display by a small triangle (►) at the left edge of the graticule. The following table shows the minimum, maximum and default reference level values for each measurement mode.

Measurement Mode	Minimum	Maximum	Default
Return Loss (dB)	-100	0	10
Reflection Coefficient	0	.5	10
SWR	-500	1	500

**Reference Position** Allows you to set the position of the reference line on the active channel.

**Position** Setting the reference position to 10 puts the line on the top graticule of the display, setting the position to 0 puts it on the bottom graticule. The default position is 9.

**Start Distance** Allows entry of a new start distance. Start distance is a parameter that is also asked for in the Guided Setup. When this softkey is pressed, you will be prompted to enter a new start distance. You can retain the current start distance if you wish by just pressing **ENTER**.

**Stop Distance** Allows the entry of a new stop distance. Stop distance is a parameter that is also asked for in the Guided Setup. When this softkey is pressed, you will be prompted to enter a new stop distance. The maximum allowable entry for stop distance is 9999.999 feet or 9999.99 meters. You can retain the current stop distance if you wish by just pressing **ENTER**.

**NOTE**

Changing start and/or stop distance will change the frequency span. See Appendix C for detailed information on the relationship between distance range and frequency span.

**SWEEP**

Pressing this key displays the sweep menu.

**Continuous** This is the default sweep mode. The analyzer begins its next sweep at the conclusion of the current sweep.

**Hold** Stops the current sweep and holds it until **Continuous** or **Single** is selected.

**Single** Triggers one complete sweep if **Hold** has been selected, and holds until retriggered with either **Continuous** or **Single**. Has no effect if instrument is currently in **Continuous** mode. (**Hold** must be pressed before **Single** can be used.)

**SYSTEM OPTIONS**

This menu allows you to select windowing options, set test limit lines, and configure the display with a title and date and time stamp.

**Window** The fault location software provides a windowing feature that makes fault measurements more useful for isolating and identifying individual responses. Windowing is needed because of the abrupt transitions in a frequency domain response at the start and stop frequencies. The band limiting of a frequency domain response causes overshoot and ringing in the distance domain response. This gives a non-windowed impulse stimulus to have a  $\sin(wt)/wt$  shape where  $w = \pi/(frequency\ span)$ . This has two effects that limit the usefulness of the fault location measurement:

1. Finite impulse width (or rise time): This limits the ability to resolve between two closely spaced responses. The effects of the finite impulse are improved as the frequency span is increased.
2. Sidelobes: The impulse sidelobes limit the dynamic range of the time domain measurement by hiding low-level responses within the sidelobes of higher level responses. The effects of sidelobes can be improved by windowing.

Windowing improves the dynamic range of a distance domain measurement by filtering the frequency domain data prior to converting to the distance domain, producing an impulse response that has lower sidelobes. This makes it much easier to see distance domain responses that are different in magnitude. This sidelobe reduction is achieved, however, at the expense of increased impulse width.

**Key Reference**

**SYSTEM OPTIONS**

The following options are available under the Window menu:

**Minimum**

This is essentially no window. It gives the highest sidelobes and the narrowest impulse response.

**Medium**

This is the default selection. Selecting medium provides reduced sidelobes.

**Maximum**

This selection provides minimum sidelobes but the widest impulse response.

**Cancel**

Returns to the **SYSTEM OPTIONS** menu.

**Limit Line** Allows entry of a test limit line. When testing cables you may have a particular specification you are testing to. For example, you may have a specification that the return loss on your cable must be  $-30$  dB or better. If you set the limit line to  $-30$ , you can just connect the cable and immediately see whether your cable meets your specification or not. The analyzer will indicate "PASS" if all responses are below the limit line, and will indicate "FAIL" if any responses fall above the limit line.

**See Also**

For an example of limit line usage, see "Interpret the Measurement" in Chapter 2.

**Title and Clock** Displays user-entered title on channel 2 display and date/time information on  
On channel 1 display.

**Title and Clock** Turns off display of title and date/time information.  
Off

**Change Title** Allows user to enter or change a title. Use the front panel knob and/or up  
and down arrow keys to select characters or words for your title. Press  
**Select Char/Word**, after each character or word selection and press  
**ENTER** when done selecting characters and/or words.

---

## **Performance Summary and Characteristics**

---

## Performance Summary and Characteristics

This chapter contains performance data for the fault location measurement software. For complete specification and characteristic information for your analyzer, see your analyzer's *User's Guide*.

---

## General Performance Characteristics

<b>Measurements</b>	Return loss [dB] versus distance Reflection coefficient magnitude versus distance SWR versus distance
<b>Dynamic Range</b>	40 dB (based on system directivity after calibration)
<b>Windowing</b>	Minimum, medium and maximum windows are available for optimizing distance response data
<b>Amplitude Accuracy<sup>1</sup></b>	±2.5 dB typical (minimum windowing) ±1.2 dB typical (medium windowing) ±0.4 dB typical (maximum windowing)
<b>Data Correction</b>	Data is normalized to the open/short/load response at the output port. Data correction for line losses and preceding mismatches is also available.
<b>Measurement and Data Storage</b>	You may store and retrieve setups and data onto or from a data disk. Typically, you may store as many as 40 setups onto a disk. You may also store data in a format compatible with popular PC-based spreadsheet software. Data may also be stored in HPGL graphics format.
<b>Markers</b>	Eight independently controlled markers can be used to display return loss, reflection coefficient, or SWR versus distance.
<b>Limit Lines</b>	Limit lines may be entered for comparison to specification limits and pass/fail testing.
<b>Remote Programming</b>	The analyzer running the fault location application can be controlled from an external controller through the IEEE 488.2 HP-IB port. Use standard SCPI program subsystem commands to control the fault location task. See Appendix D for an external program listing.
<b>Hard Copy</b>	The analyzer can be configured to output print or plot data to the serial port, the parallel port, the HP-IB port, or to a file. The data can be either a graph or a tabular listing of data points. Tabular listings include limit-line information and pass/fail margins, point by point.
<b>Fault Range</b>	Up to 9999.99 meters. (See Table 4-1 and Table 4-2.)
<b>Resolution</b>	Down to 0.195% of range. (See Table 4-1 and Table 4-2.)

<sup>1</sup> Inaccurate cable loss factor and/or multiple fault correction may introduce additional error uncertainties.

---

## Distance Range and Resolution

Resolution improves as the range is shortened and as the number of measurement points are increased. (See the following tables and graphs.) Distance is displayed in feet or meters. Typical range is limited by transmission line losses.

---

### Range

Maximum range is a function of the velocity factor ( $V_f$ ), frequency bandwidth (BW), the velocity of light in a vacuum ( $c = 2.99796 \times 10^8$  mtrs/sec), and the number of measurement points (NP), and is determined using the following formula:

$$Range = \frac{V_f(c)(NP - 1)}{2(BW)}$$

---

### Resolution

Maximum resolution is a function of the velocity factor ( $V_f$ ), frequency bandwidth (BW), sampling factor ( $N_s$ : 128 for 101 points, 256 for 201 points, and 512 for 401 points), the velocity of light in a vacuum ( $c = 2.99796 \times 10^8$  mtrs/sec), and the number of measurement points (NP), and is determined using the following formula:

$$Resolution = \frac{V_f(c)(NP - 1)}{2(BW)(N_s)} = \frac{Range}{N_s}$$

## Typical Distance Data in Feet

**Table 4-1.**  
**Fault Location Distance Range or Maximum Distance (in feet)**  
**Versus Resolution at 201 Points\***

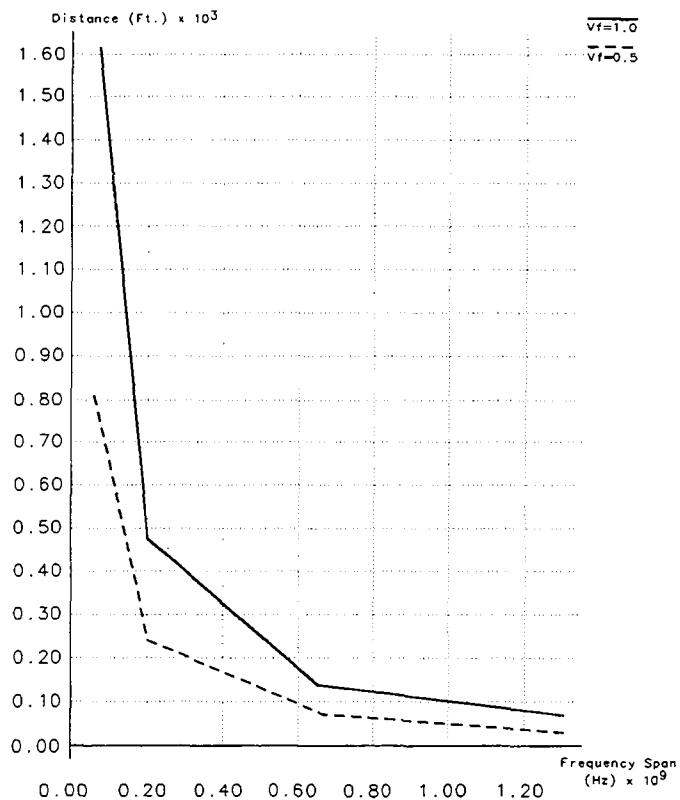
Frequency Span (MHz)	Distance Range (feet)	Resolution (feet)	Frequency Span (MHz)	Distance Range (feet)	Resolution (feet)
<b>Velocity Factor = 0.5</b>					
1300	37.18	0.15	1300	59.49	0.23
650	74.36	0.29	650	118.97	0.46
200	241.67	0.94	200	386.67	1.51
60	805.55	3.15	60	1288.88	5.03
20	2416.66	9.44	20	3866.65	15.10
<b>Velocity Factor = 0.6</b>					
1300	44.62	0.17	1300	66.92	0.26
650	89.23	0.35	650	133.85	0.52
200	290.00	1.13	200	435.00	1.70
60	966.66	3.78	60	1450.00	5.66
20	2899.99	11.33	20	4349.99	16.99
<b>Velocity Factor = 0.7</b>					
1300	52.05	0.20	1300	74.36	0.29
650	104.10	0.41	650	148.72	0.58
200	338.33	1.32	200	483.33	1.89
60	1127.77	4.41	60	1611.11	6.29
20	3383.32	13.22	20	4833.32	18.88
<b>Velocity Factor = 1.0</b>					

\* See Appendix B for typical coaxial cable characteristics including velocity factor.

Performance Summary and  
Characteristics  
**Distance Range and Resolution**

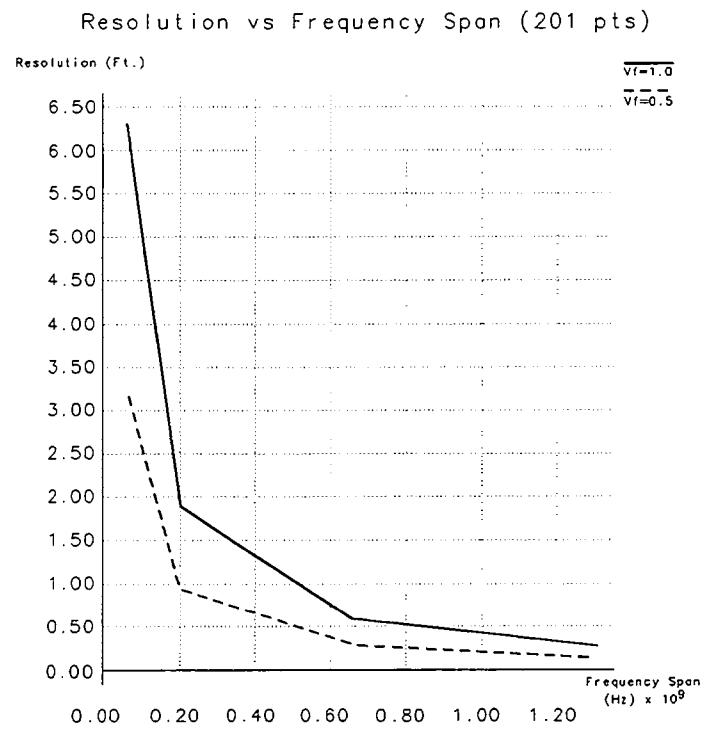
The following two graphs are plots of maximum distance versus frequency span and resolution versus frequency span using data from Table 4-1. Please note that data is plotted only for velocity factors of 0.5 and 1.0.

Max Distance vs Frequency Span (201 pts)



md67a

Performance Summary and  
Characteristics  
**Distance Range and Resolution**



md68a

**Performance Summary and  
Characteristics**  
**Distance Range and Resolution**

---

## Distance Data in Meters

**Table 4-2.**  
**Fault Location Distance Range or Maximum Distance (in meters)**  
**Versus Resolution at 201 Points\***

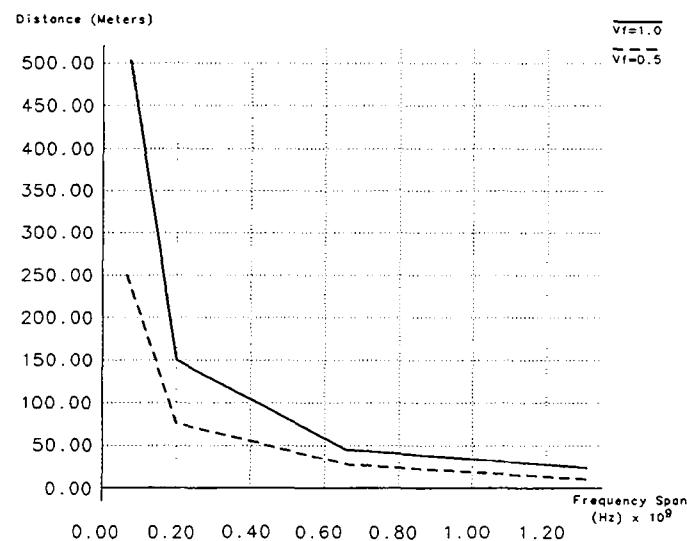
Frequency Span (MHz)	Distance Range (meters)	Resolution (meters)	Frequency Span (MHz)	Distance Range (meters)	Resolution (meters)
<b>Velocity Factor = 0.5</b>			<b>Velocity Factor = 0.8</b>		
1300	11.59	0.05	1300	18.54	0.07
650	23.18	0.09	650	37.08	0.14
200	75.32	0.29	200	120.52	0.47
60	251.08	0.98	60	401.72	1.57
20	753.23	2.94	20	1205.17	4.71
<b>Velocity Factor = 0.6</b>			<b>Velocity Factor = 0.9</b>		
1300	13.91	0.05	1300	20.86	0.08
650	27.81	0.11	650	41.72	0.16
200	90.39	0.35	200	135.58	0.53
60	301.29	1.18	60	451.94	1.77
20	903.87	3.53	20	1355.81	5.30
<b>Velocity Factor = 0.7</b>			<b>Velocity Factor = 1.0</b>		
1300	16.22	0.06	1300	23.18	0.09
650	32.45	0.13	650	46.35	0.18
200	105.45	0.41	200	150.65	0.59
60	351.51	1.37	60	502.15	1.96
20	1054.52	4.12	20	1506.46	5.88

\* See Appendix B for typical coaxial cable characteristics including velocity factor.

Performance Summary and  
Characteristics  
**Distance Range and Resolution**

The following two graphs are plots of maximum distance versus frequency span and resolution versus frequency span using data from Table 4-2. Please note that data is plotted only for velocity factors of 0.5 and 1.0.

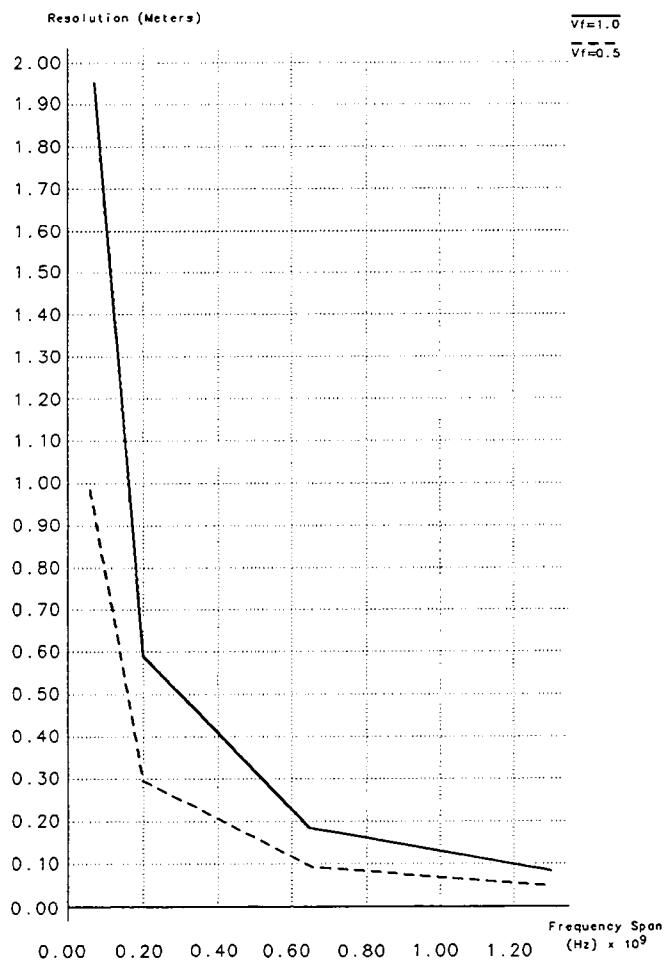
Max Distance vs Frequency Span at 201 points.



md69a

Performance Summary and  
Characteristics  
**Distance Range and Resolution**

Resolution vs Frequency Span (201 pts)



md610a

---

**License Agreement and  
Limited Warranty**

---

## License Agreement and Limited Warranty

**Important**

Please carefully read this License Agreement before opening the media envelope or operating the equipment. Rights in the software are offered only on the condition that the Customer agrees to all terms and conditions of the License Agreement. Opening the media envelope or operating the equipment indicates your acceptance of these terms and conditions. If you do not agree to the License Agreement, you may return the unopened package for a full refund.

---

## **License Agreement**

In return for payment of the applicable fee, Hewlett-Packard grants the Customer a license in the software, until terminated, subject to the following:

---

### **Use**

- Customer may use the software on one instrument.
- Customer may not reverse assemble or decompile the software.

---

### **Copies and Adaptations**

- Customer may make copies or adaptations of the software:
  - For archival purposes, or
  - When copying or adaptation is an essential step in the use of the software with a computer so long as the copies and adaptations are used in no other manner.
- Customer has no other rights to copy unless they acquire an appropriate license to reproduce which is available from Hewlett-Packard for some software.
- Customer agrees that no warranty, free installation, or free training is provided by Hewlett-Packard for any copies or adaptations made by Customer.
- All copies and adaptations of the software must bear the copyright notices(s) contained in or on the original.

## **Ownership**

- Customer agrees that they do not have any title or ownership of the software, other than ownership of the physical media.
- Customer acknowledges and agrees that the software is copyrighted and protected under the copyright laws.
- Customer acknowledges and agrees that the software may have been developed by a third party software supplier named in the copyright notice(s) included with the software, who shall be authorized to hold the Customer responsible for any copyright infringement or violation of this License Agreement.

---

## **Transfer of Rights in Software**

- Customer may transfer rights in the software to a third party only as part of the transfer of all their rights and only if Customer obtains the prior agreement of the third party to be bound by the terms of this License Agreement.
- Upon such a transfer, Customer agrees that their rights in the software are terminated and that they will either destroy their copies and adaptations or deliver them to the third party.
- Transfer to a U.S. government department or agency or to a prime or lower tier contractor in connection with a U.S. government contract shall be made only upon their prior written agreement to terms required by Hewlett-Packard.

## **Sublicensing and Distribution**

- Customer may not sublicense the software or distribute copies or adaptations of the software to the public in physical media or by telecommunication without the prior written consent of Hewlett-Packard.

## **Termination**

- Hewlett-Packard may terminate this software license for failure to comply with any of these terms provided Hewlett-Packard has requested Customer to cure the failure and Customer has failed to do so within thirty (30) days of such notice.

## **Updates and Upgrades**

- Customer agrees that the software does not include future updates and upgrades which may be available from HP under a separate support agreement.

## **Export**

- Customer agrees not to export or re-export the software or any copy or adaptation in violation of the U.S. Export Administration regulations or other applicable regulations.

---

## Limited Warranty

---

### Software

Hewlett-Packard warrants for a period of 1 year from the date of purchase that the software product will execute its programming instructions when properly installed on the instrument indicated on this package. Hewlett-Packard does not warrant that the operation of the software will be uninterrupted or error free. In the event that this software product fails to execute its programming instructions during the warranty period, customer's remedy shall be to return the media to Hewlett-Packard for replacement. Should Hewlett-Packard be unable to replace the media within a reasonable amount of time, Customer's alternate remedy shall be a refund of the purchase price upon return of the product and all copies.

---

### Media

Hewlett-Packard warrants the media upon which this product is recorded to be free from defects in materials and workmanship under normal use for a period of 1 year from the date of purchase. In the event any media prove to be defective during the warranty period, Customer's remedy shall be to return the media to Hewlett-Packard for replacement. Should Hewlett-Packard be unable to replace the media within a reasonable amount of time, Customer's alternate remedy shall be a refund of the purchase price upon return of the product and all copies.

---

## Notice of Warranty Claims

Customer must notify Hewlett-Packard in writing of any warranty claim not later than thirty (30) days after the expiration of the warranty period.

---

## Limitation of Warranty

Hewlett-Packard makes no other express warranty, whether written or oral, with respect to this product. Any implied warranty of merchantability or fitness is limited to the 1 year duration of this written warranty. This warranty gives specific legal rights, and Customer may also have other rights which vary from state to state, or province to province.

---

## Exclusive Remedies

The remedies provided above are Customer's sole and exclusive remedies. In no event shall Hewlett-Packard be liable for any direct, indirect, special, incidental, or consequential damages (including lost profit) whether based on warranty, contract, tort, or any other legal theory.

---

## Warranty Service

Warranty service may be obtained from the nearest Hewlett-Packard sales office or other location indicated in the owner's manual or service booklet.

---

A

---

---

**HP 8711A Option K20:  
Installing the Firmware**

---

## HP 8711A Option K20: Installing the Firmware

This procedure is necessary only if you have ordered HP 8711A option K20.

**NOTE**

If you have a "B" model analyzer, or if you ordered option H20, you received an analyzer already configured with the correct firmware. Do not perform this procedure if you ordered option H20, or if your analyzer is an HP 8711B/12B/13B/14B.

To load the firmware you will need the appropriate firmware disk (see "Step 3. Determine the Correct Firmware for Your Analyzer") as well as a blank disk to store the instrument's correction constants on during firmware installation.

To correctly install the firmware, the following steps should be performed in the order they appear.

---

**CAUTION**

It is extremely important that the analyzer's correction constants are stored to a disk before the new firmware is loaded.

---

## Step 1. Format the Blank Disk

This section may be omitted if the blank disk you have is already formatted in either DOS or LIF format, *or* if you have the correction constant data for your instrument already available on a disk.

1. Make sure the analyzer's internal disk drive is empty and cycle the power.
2. Label the blank disk with "CC DATA" and the serial number of the analyzer.
3. Insert the disk into the analyzer's internal disk drive, making sure first that the disk is not write-protected.
4. Format the disk by selecting the following: **SAVE RECALL**,  
**File Utilities**, **Format Disk**, **Format Int Disk**, **YES**.
5. Formatting will take approximately 2.5 minutes.

---

## Step 2. Store the Analyzer's Correction Constants To the Disk

This step may be omitted if you have current correction constant data already available on a disk.

1. With a formatted disk in the disk drive, press: **SYSTEM OPTIONS**, **Service**, **Update Corr Const**, **Store CC to Disk**.
2. When done, remove this disk and set it aside. This is a copy of the correction constants unique to your analyzer. They will need to be reloaded into the analyzer after the new firmware is installed.

---

### Step 3. Determine the Correct Firmware for Your Analyzer

Two firmware disks were supplied with your shipment:

Revision A.03.01 — HP part number 08711-10017

Revision A.02.03 — HP part number 08711-10014

You must use the following procedure to determine which revision of firmware to install in your analyzer.

1. If your analyzer has a serial number ending with 802 or greater, you will be using Revision A.03.01 (part number 08711-10017). Obtain this disk and proceed with "Step 4. Load the New Firmware."
2. Analyzers prior to serial number 802 were shipped with 1.5 MB of flash EPROM which is incompatible with firmware revisions A.03.00 and above. However, if you have one of these analyzers, it *may* contain sufficient EPROM to run the newer version of firmware if the analyzer has ever been upgraded or had its A2 CPU board replaced.
3. To determine the amount of flash EPROM contained in your particular analyzer, cycle the power on the analyzer and watch the display.
4. During the power-up routine the analyzer should display the heading "Program ROM." If a value of 2 MB is indicated, then you will be using Revision A.03.01 (part number 08711-10017).
5. If any other value (or no value) is indicated, you need to use Revision A.02.03 (part number 08711-10014).

---

## Step 4. Load the New Firmware

1. Insert the firmware disk (determined in Step 3, above) into the disk drive and cycle the power on the analyzer.
2. Upon power-up you will see a message indicating that a firmware disk has been found.

---

**CAUTION**

Do not proceed unless you have completed the previous procedure entitled "Step 2. Store the Analyzer's Correction Constants To the Disk," or you are certain that you already have the instrument's correction constant data available on a disk.

3. Press **BEGIN**.
4. The new firmware will now automatically load. The process will take about five minutes.

---

## Step 5. Restore the Correction Constants

1. Remove the firmware disk from the analyzer's disk drive.
2. Cycle the power on the analyzer.
3. Insert the "CC DATA" disk into the analyzer's disk drive.
4. When prompted, press **Install CC From Disk**.
5. Remove the disk and cycle the power on the analyzer.

The firmware installation is now complete.

---

**B**

---

**Cable Loss and Velocity  
Factor Table**

# Cable Loss and Velocity Factor Table

**Table B-1. Cable Loss and Velocity Factor Table \***

RG/U	Relative Velocity	Coaxial Cable Nominal Loss Characteristics dB per Hundred Feet, Frequency in GHz #									
		.01	.05	.10	.20	.40	1	3	5	10	
5, 5A, 5B, 6A, 6B, 212	.659	.80	1.40	2.90	4.30	6.40	11.00	22.00	30.00	52.00	
7	.659	.66	1.50	2.20	3.20	4.60	9.00	19.00	28.00	47.00	
8, 8A, 10A, 70, 213, 215	.659	.66	1.50	2.20	3.20	4.60	9.00	19.00	28.00	47.00	
9, 9A, 9B, 214	.659	.66	1.50	2.20	3.20	4.60	9.00	19.00	28.00	47.00	
11, 11A, 12, 12A, 13, 13A, 216	.659	.66	1.50	2.20	3.20	4.60	9.00	19.00	28.00	-	
14, 14A, 74, 74A, 217, 224, 293, 293A, 388	.659	.41	1.00	1.40	2.10	3.10	5.80	13.00	19.00	31.00	
17, 17A, 18, 18A, 84A, 85A, 177, 218, 219, 295	.659	.23	.56	.81	1.20	1.90	3.80	9.00	13.50	-	
19, 19A, 20, 20A, 147, 220, 221	.659	.17	.43	.63	.94	1.50	3.00	7.00	-	-	
21, 21A, 222	.659	4.40	9.40	12.90	18.20	26.50	44.00	87.00	-	-	
22, 22B, 111, 111A	.659	1.20	2.80	4.20	6.30	9.60	-	-	-	-	
29	.659	1.35	3.00	4.30	6.00	8.80	16.50	36.00	51.00	85.00	
34, 34A, 34B	.659	.32	.90	1.40	2.10	3.30	5.80	16.00	28.00	-	
35, 35A, 35B, 164	.659	.24	.60	.90	1.30	2.00	3.70	8.90	15.00	-	
54, 54A	.659	.90	2.20	3.30	4.60	6.90	13.10	26.20	35.00	-	
55, 55A, 55B, 223	.659	1.35	3.00	4.30	6.00	8.80	16.50	36.00	51.00	85.00	
57, 57A, 130, 131, 294, 294A	.659	.65	1.60	2.40	3.60	5.20	10.00	21.20	-	-	
58, 58B	.659	1.20	3.10	4.60	7.00	10.00	17.50	38.00	-	-	
58A, 58C	.659	1.40	3.30	4.90	7.30	11.00	20.00	41.00	-	-	

\* Times Wire and Cable, *RF Transmission Line Catalog and Handbook*. Catalog TL-6, 1972.

# CONDITIONS: Ambient 20 °C

Cable Loss and Velocity Factor Table

**Table B-1. Cable Loss and Velocity Factor Table (continued)**

RG/U	Relative Velocity	.01	.05	.10	.20	.40	1	3	5	10
59, 59A, 59B	.659	1.10	2.30	3.30	4.70	6.70	11.50	25.50	41.00	-
62, 62A, 71, 71A, 71B	.84	.90	1.90	2.80	3.70	5.20	8.50	18.40	29.50	-
63, 63B, 79, 79B	.84	.50	1.10	1.50	2.30	3.40	5.70	12.20	20.90	-
87A, 115, 115A, 116, 165, 166, 225, 227, 393, 397	.695	.60	1.40	2.10	3.10	4.50	7.50	14.00	21.00	35.00
94	.695	.60	1.30	2.00	2.90	4.20	7.10	13.00	19.00	33.00
94A, 226	.695	.40	1.00	1.50	2.10	3.00	5.00	10.00	15.00	27.00
108, 108A	.659	2.30	5.20	7.50	11.00	16.00	26.20	54.00	-	-
117, 117A, 118, 211, 211A, 228, 228A	.695	.25	.61	.90	1.40	2.00	3.40	7.50	11.50	-
119, 120	.695	.50	1.05	1.60	2.20	3.10	5.10	10.20	15.20	27.30
122	.659	1.60	4.40	6.90	11.00	16.60	29.20	57.20	89.00	-
140, 141, 141A, 142, 142B, 159, 302, 303, 400, 402	.695	1.20	2.70	3.90	5.50	8.00	13.00	26.00	36.00	62.00
143, 143A, 304, 401	.695	.85	1.80	2.50	3.80	5.70	9.70	18.10	26.10	40.70
144	.695	.38	1.00	1.60	2.30	3.80	7.00	15.10	-	-
161, 179, 179A, 179B, 187, 187A	.695	5.00	7.20	9.80	12.70	15.80	25.00	43.00	62.50	135.00
174, 174A	.659	3.80	6.50	8.90	12.00	17.50	31.00	64.30	97.00	185.00
178, 178A, 178B, 196, 196A, 403, 404	.695	5.30	10.00	13.30	20.00	27.50	45.00	78.00	115.00	172.00
180, 180A, 180B, 195, 195A	.695	3.10	4.20	5.10	7.30	10.40	16.50	36.00	49.00	89.00
183	.91	.18	.38	.53	.78	1.20	1.90	3.70	5.00	-
188, 188A, 316	.695	3.80	7.90	11.50	15.00	20.00	30.00	58.00	79.00	133.00
235	.695	.60	1.40	2.10	3.10	4.50	7.50	14.00	21.00	35.00
306, 306A, 336	.80	.15	.33	.52	.80	1.30	2.30	5.20	7.80	-
307, 307A	.80	1.20	2.70	3.80	5.40	7.50	12.00	-	-	-
323, 324, 332, 333, 376	.80	.15	.32	.50	.75	1.20	2.10	4.70	6.50	-
334, 335	.80	.25	.60	.85	1.20	1.90	3.50	7.00	10.00	18.00
360	.80	.18	.40	.60	.90	1.50	2.50	5.30	7.50	-

---

C

---

## Measurement Theory

---

## Measurement Theory

This section explains how the analyzer converts frequency domain data to distance domain data. It also explains the relationship between start distance, stop distance and frequency span.

---

## How the Analyzer Converts Frequency Data to Distance Data

The analyzer performs swept frequency measurements of return loss, then performs an inverse Fourier Transform on the result to convert to the distance domain. Other cable corrections are done to correct for cable loss and multiple faults.

Fundamentally, the analyzer has a tuned receiver, measuring at discrete frequencies. In order to use the analyzer's basic high performance to make distance domain measurements, the frequency domain data has to be converted to the distance domain.

The way to do this is to build the impulse out of the frequencies measured. The process is one of measuring many different frequencies and adding the scaled responses back together.

For example, an impulse can be synthesized from three sine waves or three measured points. The first point is  $F_0$ , the fundamental frequency of an impulse train spectrum. The impulse train will repeat at one over the frequency of the fundamental. We then add in the first harmonic ( $2 * F_0$ ). The impulse will start to become defined. As the second harmonic ( $3 * F_0$ ) is added in, the increasing bandwidth makes a narrower pulse. The more points used, the more bandwidth included, and the narrower the impulse will become.

The process of adding sine waves is expressed mathematically by the inverse discrete Fourier transform (inverse DFT).

You can use this information to build an impulse response. Suppose that before adding the sine waves back together, we send each one out to a device under test, and measure the magnitude and phase offset of the return sine wave. As long as the device is linear, it does not matter if we send the whole impulse at once, or if we only send one spectral piece at a time: we will still get the same response.

---

## Start/Stop Distance and Frequency Span Explanation

When the analyzer is set up for a measurement, you determine the center frequency, and start and stop distances for the measurement. The fault location software uses the distance range (start distance minus stop distance) to determine the frequency span, which in turn, determines the start and stop frequencies. Note that the only user-chosen frequency parameter is center frequency. Changes to the distance range do not affect the user-chosen center frequency setting.

The fault location software will attempt to set the frequency span to the setting required for the distance range. The maximum setting for the frequency span cannot exceed a "legal" setting. For instance, the start frequency cannot be lower than the analyzer's low frequency limit, and the stop frequency cannot be higher than the analyzer's high frequency limit.

When the distance range requires a larger span than can be legally set, the frequency span is set to the maximum legal setting and extra processing is done on the frequency domain data to provide the response in the distance domain. This extra processing is called Zoom mode. The "Zoom" message will appear in the Channel 2 annotation on the screen.

Zoom mode is a way to create a distance domain response with arbitrary start and stop distances for a given frequency span. Because Zoom mode requires more processing, sweep update rates will slow down in this mode. The software will generally attempt to set frequency span to a legal setting for the requested distance range before switching to Zoom mode.

---

D

---

## External Programming

---

## External Programming

The BASIC program, filename "FAULTEXT," runs on a Hewlett-Packard workstation equipped with HP BASIC, or a personal computer equipped with HP Instrument BASIC (IBASIC) for Windows<sup>TM</sup>. The program allows the external computer to control the analyzer in order to perform fault location measurements remotely over Hewlett-Packard Interface Bus (HP-IB). Using this configuration, the external computer can perform fault location measurements, and query the measurement data for statistical or archival purposes. This program is provided on the software disk you received with your shipment (HP part number 08711-10013).

See the programming information provided in your analyzer's *Programming Guide* for information on programming.

---

---

## **Index**

---

# Index

**1** 101 points, 3-21

**2** 201 points, 3-21

**4** 401 points, 3-21

**A** activating markers, 3-19

active marker, 3-19

active mkr-->max, 3-19

active mkr off, 3-20

agreement

software license, 5-2

all off, 3-20

analyzer calibration, 2-6, 3-10

ASCII data

how to save, 3-23

autoscale, 3-24

AUTOST program, 1-3, 1-4

average factor, 3-9

averaging, 3-9

averaging, how it works, 3-9

**AVG**, 3-9

**B** blank disk

how to format, A-3

blinking the graticule, 3-15

**C** cable calibration, 3-10, 3-11

cable loss, 3-10

cable loss table, B-2

cable parameters

typical, B-2

cable specifications, 2-5

**CAL**, 3-10

calibrate cable, 3-11

calibrate instrument, 3-10

calibration, 2-6

calibration menu, 3-10

cancel, 3-28

center frequency, 3-17  
[CHAN 1], 3-13  
[CHAN 2], 3-14  
change title, 3-29  
changing distance units, 3-16  
changing frequency, 3-17  
continue, 1-11  
continuous, 3-26  
control  
    external, D-2  
copying correction constants, A-4  
correction, 2-6, 3-10  
    multi peak, 3-12  
correction constants, A-4  
    to restore, A-7

D data  
    ASCII, 3-23  
    how to recall, 3-23  
    how to save, 3-23  
    scaling, 3-24  
data correction, 4-3  
data tracepoints, 3-21  
date/time display, 3-29  
default calibration, 2-6, 3-10  
default reference levels, 3-24  
default setup, 1-8  
    Default Setup, 1-8  
default state, 1-8  
default values for parameters, 1-8  
delete files, 3-23  
delta mkr off/on, 3-20  
diagram  
    setup, 1-11  
diagram setup, 2-7  
directivity calibration, 3-10  
disk  
    format, 3-18  
    formatting, A-3  
    software, 1-3  
    to format, 3-23  
[DISPLAY], 3-15  
display date/time and title, 3-29  
displaying the marker table, 3-15  
distance  
    start, 3-21, 3-24  
    stop, 3-21, 3-25  
distance units, 3-16

distance vs bandwidth (in feet)  
graph, 4-6  
distance vs bandwidth (in meters)  
graph, 4-9  
distance vs resolution (in feet)  
table, 4-5  
distance vs resolution (in meters)  
table, 4-8  
documentation outline, vi  
dynamic range, 3-27, 4-3

**E** equipment setup, 1-11, 2-7  
example measurements, 2-11-19  
exiting the program, 1-12  
external control, D-2  
external programming, D-2

**F** FAULTEXT program, 1-3, D-2  
fault location theory, 2-3  
FAULT program, 1-3  
feet, 3-16  
file  
plots, 3-18  
to delete, 3-23  
firmware  
choosing the correct disk, A-5  
firmware installation, A-2-7  
**FORMAT**, 3-16  
format disk, 3-23  
formatting a disk, 3-18, A-3  
**[FREQ]**, 3-17  
frequency response calibration, 3-10  
frequency span, 3-17  
front panel key conventions, vi

**G** graticule on/off, 3-15  
guided setup, 1-9, 2-5  
parameter default values, 1-8  
**Guided Setup**, 1-9  
guided setup parameters, 2-5

- H** H20 option, iv  
hard copy, 4-3  
**[HARD COPY]**, 3-18  
hardkey conventions, vi  
hardkeys, 3-2-29  
hold, 3-26
- I** IBASIC option, 1-2  
indicator  
    reference line, 3-24  
installing the firmware, A-2-7  
instrument calibration, 2-6, 3-10  
instrument setup  
    how to save, 3-23  
instrument setups  
    how to recall, 3-23  
interpreting measurements, 2-8
- K** K20 option, iv  
key conventions, vi  
keys, 3-2-29
- L** limited warranty, 5-6  
limit line, 3-28  
    example, 2-10  
loading the program, 1-5  
loss  
    cable, 3-10
- M** making measurements, 2-4  
marker  
    example, 2-9  
**[MARKER]**, 3-19  
marker data, 3-15  
marker menu, 3-19  
marker offsets, 3-20  
marker reference, 3-20  
markers  
    moving manually, 3-19  
marker screen, 3-15  
marker to maximum, 3-19  
maximum, 3-28  
maximum window, 3-28  
measurement basics, 2-3  
measurement examples, 2-11-19  
measurement interpretation, 2-8

measurements  
    how to make, 2-4  
measurement setup, 2-7  
medium, 3-28  
medium window, 3-28  
**[MENU]**, 3-21  
menumaps, 3-3  
meters, 3-16  
minimum, 3-28  
minimum window, 3-28  
more mkrs, 3-20  
moving markers, 3-19  
multi peak threshold, 3-12  
multiple responses, 3-12

**N** next peak  
    example, 2-9  
next peak left, 3-19  
next peak right, 3-19  
noise reduction, 3-9  
normal operation, 1-12  
number of points, 3-21

**O** offset marker, 3-20  
operation  
    normal, 1-12  
option 1C2, 1-2  
option H20, iv  
option K20, iv  
options, iv  
outline  
    documentation, vi  
output power, 3-22

**P** pass/fail line  
    example, 2-10  
pass/fail testing, 3-28  
pause, 1-11  
**[PAUSE]**, 1-11  
pausing the program, 1-11  
performance characteristics, 4-3  
plot files, 3-18  
plot HPGL to disk, 3-18  
plotting data, 3-18  
plot to default, 3-18  
**[POWER]**, 3-22  
power level, 3-22

preset, 1-12  
printing data, 3-18  
program  
    exiting, 1-12  
    to load from front panel, 1-5  
    to run, 1-5  
program defaults, 1-8  
programs on disk, 1-3

**R** random noise reduction, 3-9  
recalibrating, 3-10  
recall setup/data, 3-23  
reference level, 3-24  
reference level defaults, 3-24  
reference line, 3-24  
    positioning, 3-24  
reference line indicator, 3-24  
reference marker, 3-20  
reference position, 3-24  
refl. coeff. mag, 3-13, 3-14  
reflection coefficient, 2-3  
reflection coefficient magnitude, 3-13, 3-14  
remote programming, 4-3  
reset cal to default, 3-10  
resolution vs bandwidth (in feet)  
    graph, 4-7  
resolution vs bandwidth (in meters)  
    graph, 4-10  
resume the program, 1-11  
return loss, 2-3, 3-13, 3-14  
running the program, 1-5  
running the software, 1-3-6

**S** save ASCII trace data, 3-23  
    **SAVE RECALL**, 3-23  
save setup/data, 3-23  
saving data to disk, 3-18  
    **SCALE**, 3-24  
scale/div, 3-24  
setup  
    default, 1-8  
    equipment, 1-11, 2-7  
    guided, 1-9, 2-5  
setup diagram, 1-11, 2-7  
    **Setup Diagram**, 1-11  
setups  
    how to recall, 3-23

- how to save, 3-23
- sidelobes, 3-27
- single, 3-26
- single sweep, 3-26
- softkey convention, vi
- softkeys, 3-2-29
- software
  - to exit, 1-12
  - to pause, 1-11
  - to run, 1-3
- software copying, 5-3
- software defaults, 1-8
- software disk, 1-3
- software license agreement, 5-2
- software product license agreement, 5-2-5
- software programs, 1-3
- software updates and upgrades, 5-5
- source match calibration, 3-10
- span
  - frequency, 3-17
- split screen, 3-15
- spreadsheet data, 3-23
- standing wave ratio, 2-3, 3-13, 3-14
- start distance, 3-21, 3-24
- stop distance, 3-21, 3-25
- stop sweep, 3-26
- storing correction constants, A-4
- SWEET, 3-26
- sweep trigger menu, 3-26
- SWR, 2-3, 3-13, 3-14
- SYSTEM OPTIONS, 3-27

**T** test limit line, 3-28  
    example, 2-10  
theory, 2-3  
time/date display, 3-29  
title

- how to change, 3-29

  
title and clock off, 3-29  
title and clock on, 3-29  
title display, 3-29  
tracepoints

- selecting, 3-21

  
trigger

- single, 3-26
- triggering, 3-26

  
turning markers off, 3-20  
typical cable loss table, B-2

typical velocity factor table, B-2

**U** using markers, 3-19  
example, 2-9

**V** velocity factor, 3-10  
definition, 3-10  
velocity factor table, B-2

**W** warranty, 5-7  
limited, 5-6  
window, 3-27  
maximum, 3-28  
medium, 3-28  
minimum, 3-28  
*windowing*, 3-27, 4-3