Programmer's Guide

HP 8753D Network Analyzer
Including Option 011
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.
Warranty

This Hewlett-Packard instrument product is warranted against defects in material and workmanship for a period of one year from date of shipment. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products which prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard. Buyer shall prepay shipping charges to Hewlett-Packard and Hewlett-Packard shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to Hewlett-Packard from another country.

Hewlett-Packard warrants that its software and firmware designated by Hewlett-Packard for use with an instrument will execute its programming instructions when properly installed on that instrument. Hewlett-Packard does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error-free.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HEWLETT-PACKARD SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Exclusive Remedies

THE REMEDIES PROVIDED HEREIN ARE BUYER’S SOLE AND EXCLUSIVE REMEDIES. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.
Assistance

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products. For any assistance, contact your nearest Hewlett-Packard Sales and Service Office.

Statement of Compliance

This instrument has been designed and tested in accordance with IEC Publication 348, Safety Requirements for Electronic Measuring Apparatus, and has been supplied in a safe condition. The instruction documentation contains information and warnings which must be followed by the user to ensure safe operation and to maintain the instrument in a safe condition.

Safety Notes

The following safety notes are used throughout this manual. Familiarize yourself with each of the notes and its meaning before operating this instrument.

<table>
<thead>
<tr>
<th>Warning</th>
<th>No operator serviceable parts inside. Refer servicing to qualified personnel. To prevent electrical shock, do not remove covers.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warning</td>
<td>If this instrument is used in a manner not specified by Hewlett-Packard Co., the protection provided by the instrument may be impaired.</td>
</tr>
<tr>
<td>Warning</td>
<td>For continued protection against fire hazard replace line fuse only with same type and rating ([F 6.3A/250V]). The use of other fuses or material is prohibited.</td>
</tr>
<tr>
<td>Warning</td>
<td>This is a Safety Class I product (provided with a protective earthing ground incorporated in the power cord). The mains plug shall only be inserted in a socket outlet provided with a protective earth contact. Any interruption of the protective conductor, inside or outside the instrument, is likely to make the instrument dangerous. Intentional interruption is prohibited.</td>
</tr>
<tr>
<td>Warning</td>
<td>If this instrument is to be energized via an external autotransformer for voltage reduction, make sure that its common terminal is connected to a neutral (earthed pole) of the power supply.</td>
</tr>
<tr>
<td>Warning</td>
<td>The power cord is connected to internal capacitors that may remain live for 10 seconds after disconnecting the plug from its power supply.</td>
</tr>
<tr>
<td><strong>Caution</strong></td>
<td>Always use the three-prong ac power cord supplied with this instrument. Failure to ensure adequate earth grounding by not using this cord may cause instrument damage.</td>
</tr>
<tr>
<td><strong>Caution</strong></td>
<td>This instrument has autoranging line voltage input; be sure the supply voltage is within the specified range.</td>
</tr>
<tr>
<td><strong>Caution</strong></td>
<td>Ventilation Requirements: When installing the instrument in a cabinet, the convection into and out of the instrument must not be restricted. The ambient temperature (outside the cabinet) must be less than the maximum operating temperature of the instrument by 4 °C for every 100 watts dissipated in the cabinet. If the total power dissipated in the cabinet is greater than 800 watts, then forced convection must be used.</td>
</tr>
</tbody>
</table>

⚠ The instruction documentation symbol. The product is marked with this symbol when it is necessary for the user to refer to the instructions in the documentation.

“CE” The CE mark is a registered trademark of the European Community. (If accompanied by a year, it is when the design was proven.)

“ISM1-A” This is a symbol of an Industrial Scientific and Medical Group 1 Class A product.

“CSA” The CSA mark is a registered trademark of the Canadian Standards Association.
How to Use This Guide

This guide uses the following conventions:

Front-Panel Key This represents a key physically located on the instrument.

Softkey This represents a "softkey," a key whose label is determined by the instrument’s firmware.

Screen Text This represents text displayed on the instrument’s screen.
The **Installation and Quick Start Guide** familiarizes you with the HP 8753D/Option 011 network analyzer's front and rear panels, electrical and environmental operating requirements, as well as procedures for installing, configuring, and verifying the operation of the analyzer.

The **User's Guide** shows how to make measurements, explains commonly-used features, and tells you how to get the most performance from your analyzer.

The **Quick Reference Guide** provides a summary of all available user features.

The **Programmer's Guide** provides programming information including: an HP-IB command reference, an HP-IB programming reference, as well as programming examples.

The **System Verification and Test Guide** provides the system verification and performance tests and the Performance Test Record for your HP 8753D/Option 011 network analyzer.
# Contents

1. **HP-IB Command Reference**  
   HP-IB Capabilities ........................................ 1-2  
   Held Commands ............................................ 1-2  
   Operation Complete ....................................... 1-3  
   Command Interrogate .................................... 1-3  
   Identification ............................................ 1-4  
   Output Queue .............................................. 1-4  
   Units ....................................................... 1-4  
   Input Data ................................................ 1-5  
   Input Syntax .............................................. 1-5  
   General Structure ....................................... 1-5  
   Valid Characters ......................................... 1-6  
   Programming Data ........................................ 1-7  
   Key Select Codes for the HP 8753D/Option 011 Network Analyzer ........................................ 1-8  
   HP-IB Only Commands .................................... 1-29  
   Calibration ................................................. 1-34  
   Display Graphics ......................................... 1-36  
   User Graphics Units .................................... 1-36  
   HP-GL subset: ............................................. 1-36  
   Accepted but ignored HP-GL commands: ............... 1-37  
   Useful Tables and Figures ................................ 1-38  
   Disk File Names ........................................... 1-40  
   Alphabetical Code Listing ............................... 1-42  

2. **HP-IB Programming Reference**  
   Where to Look for More Information .................... 2-2  
   HP-IB Operation .......................................... 2-2  
   Device Types .............................................. 2-2  
   Talker ....................................................... 2-2  
   Listener .................................................... 2-3  
   Controller .................................................. 2-3  
   HP-IB Bus Structure ....................................... 2-3  
   Data Bus ..................................................... 2-4  
   Handshake Lines .......................................... 2-4  
   Control Lines .............................................. 2-4  
   HP-IB Requirements ....................................... 2-5  
   HP-IB Operational Capabilities ......................... 2-5  
   HP-IB Status Indicators .................................. 2-6  
   Bus Device Modes ......................................... 2-6  
   System-Controller Mode .................................. 2-7  
   Talker/Listener Mode ..................................... 2-7  
   Pass-Control Mode ........................................ 2-7  
   Analyzer Bus Modes ....................................... 2-7  
   Setting HP-IB Addresses .................................. 2-8  
   Response to HP-IB Meta-Messages (IEEE-488 Universal Commands) ..................................... 2-8  
   Abort ......................................................... 2-8
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Clear</td>
<td>2-8</td>
</tr>
<tr>
<td>Local</td>
<td>2-8</td>
</tr>
<tr>
<td>Local Lockout</td>
<td>2-9</td>
</tr>
<tr>
<td>Parallel Poll</td>
<td>2-9</td>
</tr>
<tr>
<td>Pass Control</td>
<td>2-9</td>
</tr>
<tr>
<td>Remote</td>
<td>2-9</td>
</tr>
<tr>
<td>Serial Poll</td>
<td>2-9</td>
</tr>
<tr>
<td>Trigger</td>
<td>2-9</td>
</tr>
<tr>
<td>Analyzer Command Syntax</td>
<td>2-10</td>
</tr>
<tr>
<td>Code Naming Convention</td>
<td>2-10</td>
</tr>
<tr>
<td>Valid Characters</td>
<td>2-11</td>
</tr>
<tr>
<td>Units and Terminators</td>
<td>2-11</td>
</tr>
<tr>
<td>Command Formats</td>
<td>2-11</td>
</tr>
<tr>
<td>General Structure:</td>
<td>2-11</td>
</tr>
<tr>
<td>Syntax Types</td>
<td>2-12</td>
</tr>
<tr>
<td>Analyzer Operation</td>
<td>2-13</td>
</tr>
<tr>
<td>Held Commands</td>
<td>2-13</td>
</tr>
<tr>
<td>Operation Complete</td>
<td>2-13</td>
</tr>
<tr>
<td>Reading Analyzer Data</td>
<td>2-14</td>
</tr>
<tr>
<td>Output Queue</td>
<td>2-14</td>
</tr>
<tr>
<td>Command Interrogate</td>
<td>2-14</td>
</tr>
<tr>
<td>Output Syntax</td>
<td>2-15</td>
</tr>
<tr>
<td>Marker data</td>
<td>2-16</td>
</tr>
<tr>
<td>Array-Data Formats</td>
<td>2-18</td>
</tr>
<tr>
<td>Trace-Data Transfers</td>
<td>2-19</td>
</tr>
<tr>
<td>Frequency-Related Arrays</td>
<td>2-20</td>
</tr>
<tr>
<td>Data-Processing Chain</td>
<td>2-21</td>
</tr>
<tr>
<td>Data Arrays</td>
<td>2-21</td>
</tr>
<tr>
<td>Data Levels</td>
<td>2-22</td>
</tr>
<tr>
<td>Learn String and Calibration-Kit String</td>
<td>2-23</td>
</tr>
<tr>
<td>Error Reporting</td>
<td>2-24</td>
</tr>
<tr>
<td>Status Reporting</td>
<td>2-24</td>
</tr>
<tr>
<td>The Status Byte</td>
<td>2-26</td>
</tr>
<tr>
<td>The Event-Status Register and Event-Status Register B</td>
<td>2-27</td>
</tr>
<tr>
<td>Error Output</td>
<td>2-27</td>
</tr>
<tr>
<td>Measurement Process</td>
<td>2-28</td>
</tr>
<tr>
<td>Step 1. Setting Up the Instrument</td>
<td>2-28</td>
</tr>
<tr>
<td>Step 2. Calibrating the Test Setup</td>
<td>2-28</td>
</tr>
<tr>
<td>Step 3. Connecting the Device under Test</td>
<td>2-28</td>
</tr>
<tr>
<td>Step 4. Taking the Measurement Data</td>
<td>2-29</td>
</tr>
<tr>
<td>Step 5. Post-Processing the Measurement Data</td>
<td>2-29</td>
</tr>
<tr>
<td>Step 6. Transferring the Measurement Data</td>
<td>2-29</td>
</tr>
<tr>
<td>Programming Examples</td>
<td>2-30</td>
</tr>
<tr>
<td>Program Information</td>
<td>2-31</td>
</tr>
<tr>
<td>Analyzer Features Helpful in Developing Programming Routines</td>
<td>2-31</td>
</tr>
<tr>
<td>Analyzer-Debug Mode</td>
<td>2-31</td>
</tr>
<tr>
<td>User-Controllable Sweep</td>
<td>2-31</td>
</tr>
<tr>
<td>Example 1: Measurement Setup</td>
<td>2-32</td>
</tr>
<tr>
<td>Example 1A: Setting Parameters</td>
<td>2-32</td>
</tr>
<tr>
<td>Tutorial Program Explanation</td>
<td>2-32</td>
</tr>
<tr>
<td>Initializing the System</td>
<td>2-32</td>
</tr>
<tr>
<td>Main Program</td>
<td>2-33</td>
</tr>
<tr>
<td>Running the Program</td>
<td>2-34</td>
</tr>
<tr>
<td>Example 1B: Verifying Parameters</td>
<td>2-35</td>
</tr>
<tr>
<td>Tutorial Program Explanation</td>
<td>2-35</td>
</tr>
</tbody>
</table>
Initializing the System ........................................ 2-35
Main Program .................................................... 2-36
Running the Program .......................................... 2-36
Example 2: Measurement Calibration ........................ 2-37
  Calibration kits ............................................... 2-37
  Example 2A: $S_1$ 1-Port Measurement Calibration .... 2-37
    Tutorial Program Explanation ........................... 2-38
    Initializing the System ................................ 2-38
    Main Program ............................................ 2-38
    Main Program Resumes .................................. 2-39
    Running the Program .................................... 2-41
  Example 2B: Full 2-Port Measurement Calibration .... 2-41
    Tutorial Program Explanation ........................... 2-41
    Initializing the System ................................ 2-41
    Reflection Portion of the Full 2-Port Measurement Calibration Program 2-42
      Resumes ............................................... 2-42
    Transmission Portion of the Full 2-Port Measurement Calibration Program
      Resumes ............................................... 2-44
      Isolation Portion of the Full 2-Port Measurement Calibration Program Resumes 2-45
      Running the Program .................................... 2-47
Example 3: Measurement Data Transfer ...................... 2-48
  Trace-Data Formats and Transfers ......................... 2-48
  Example 3A: Data Transfer Using Markers ................ 2-49
    Tutorial Program Explanation ........................... 2-49
    Initializing the System ................................ 2-49
    Main Program ............................................ 2-50
    Running the Program .................................... 2-50
  Example 3B: Data Transfer Using FORM 4 (ASCII Transfer) 2-51
    Tutorial Program Explanation ........................... 2-51
    Initializing the System ................................ 2-52
    Main Program ............................................ 2-52
    Running the Program .................................... 2-52
  Example 3C: Data Transfer Using Floating-Point Numbers 2-54
    Tutorial Program Explanation ........................... 2-54
    Initializing the System ................................ 2-55
    Main Program ............................................ 2-55
    Running the Program .................................... 2-55
  Example 3D: Data Transfer Using Frequency Array Information 2-56
    Tutorial Program Explanation ........................... 2-56
    Initializing the System ................................ 2-57
    Main Program ............................................ 2-57
    Running the Program .................................... 2-58
Example 3E: Data Transfer Using FORM 1 (Internal Binary Format) 2-60
  Tutorial Program Explanation ........................... 2-60
  Initializing the System ................................ 2-60
  Main Program ............................................ 2-61
  Running the Program .................................... 2-61
Example 4: Measurement Process Synchronization .............. 2-62
  Status Reporting .......................................... 2-62
Example 4A: Using the Error Queue ......................... 2-63
  Tutorial Program Explanation ........................... 2-63
  Initializing the System ................................ 2-63
  Main Program ............................................ 2-64
  Running the Program .................................... 2-64
Example 4B: Generating Interrupts ........................................... 2-65
  Tutorial Program Explanation ........................................... 2-66
  Initializing the System .................................................. 2-66
  Main Program .............................................................. 2-66
  Running the Program ..................................................... 2-67
Example 4C: Power Meter Calibration ........................................ 2-69
  Tutorial Program Explanation .......................................... 2-70
  Initializing the System .................................................. 2-70
  Main Program .............................................................. 2-71
  Running the Program ..................................................... 2-73
Example 5: Network Analyzer System Setups ................................ 2-74
  Storing and Recalling Instrument States ............................... 2-74
Example 5A: Using the Learn String ........................................ 2-74
  Tutorial Program Explanation .......................................... 2-75
  Initializing the System .................................................. 2-75
  Main Program .............................................................. 2-75
  Running the Program ..................................................... 2-75
Example 5B: Reading Calibration Data ....................................... 2-76
  Tutorial Program Explanation .......................................... 2-77
  Initializing the System .................................................. 2-77
  Main Program .............................................................. 2-78
  Running the Program ..................................................... 2-79
Example 5C: Saving and Restoring the Analyzer Instrument State ...... 2-79
  Tutorial Program Explanation .......................................... 2-80
  Initializing the System .................................................. 2-80
  Main Program .............................................................. 2-81
  Running the Program ..................................................... 2-82
Example 6: Limit-Line Testing ................................................ 2-83
  Using List-Frequency Mode .............................................. 2-83
Example 6A: Setting Up a List-Frequency Sweep ......................... 2-83
  Tutorial Program Explanation .......................................... 2-84
  Initializing the System .................................................. 2-84
  Main Program .............................................................. 2-84
  Main Program Resumes ................................................... 2-85
  Running the Program ..................................................... 2-86
Example 6B: Selecting a Single Segment from a Table of Segments .... 2-86
  Tutorial Program Explanation .......................................... 2-87
  Initializing the System .................................................. 2-87
  Running the Program ..................................................... 2-89
Example 6C: Setting Up Limit Lines ......................................... 2-89
  Tutorial Program Explanation .......................................... 2-90
  Initializing the System .................................................. 2-90
  Main Program .............................................................. 2-90
  Main Program Resumes ................................................... 2-92
  Running the Program ..................................................... 2-92
Example 6D: Performing PASS/FAIL Tests While Tuning .................. 2-93
  Tutorial Program Explanation .......................................... 2-94
  Initializing the System .................................................. 2-94
  Main Program .............................................................. 2-94
  Running the Program ..................................................... 2-95
Example 7: Report Generation ................................................ 2-96
Example 7A1: Operation Using Talker/Listener Mode ...................... 2-96
  Tutorial Program Explanation .......................................... 2-96
  Initializing the System .................................................. 2-97
  Main Program .............................................................. 2-97
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running the Program</td>
<td>2-98</td>
</tr>
<tr>
<td>Example 7A2: Controlling Peripherals Using Pass-Control Mode</td>
<td>2-98</td>
</tr>
<tr>
<td>Tutorial Program Explanation</td>
<td>2-99</td>
</tr>
<tr>
<td>Initializing the System</td>
<td>2-99</td>
</tr>
<tr>
<td>Main Program</td>
<td>2-99</td>
</tr>
<tr>
<td>Running the Program</td>
<td>2-101</td>
</tr>
<tr>
<td>Example 7A3: Printing with the Serial Port</td>
<td>2-101</td>
</tr>
<tr>
<td>Tutorial Program Explanation</td>
<td>2-102</td>
</tr>
<tr>
<td>Initializing the System</td>
<td>2-102</td>
</tr>
<tr>
<td>Main Program</td>
<td>2-102</td>
</tr>
<tr>
<td>Running the Program</td>
<td>2-103</td>
</tr>
<tr>
<td>Example 7B1: Plotting to a File and Transferring the File Data to a Plotter</td>
<td>2-103</td>
</tr>
<tr>
<td>Tutorial Program Explanation</td>
<td>2-104</td>
</tr>
<tr>
<td>Initializing the System</td>
<td>2-104</td>
</tr>
<tr>
<td>Main Program</td>
<td>2-104</td>
</tr>
<tr>
<td>Running the Program</td>
<td>2-105</td>
</tr>
<tr>
<td>Utilizing PC-Graphics Applications Using the Plot File</td>
<td>2-105</td>
</tr>
<tr>
<td>LIF2DOS</td>
<td>2-106</td>
</tr>
<tr>
<td>Example 7B2: Reading Plot Files From a Disk</td>
<td>2-106</td>
</tr>
<tr>
<td>Running the Program</td>
<td>2-107</td>
</tr>
<tr>
<td>Example 7C: Reading ASCII Disk Files to the Instrument Controller's Disk File</td>
<td>2-108</td>
</tr>
<tr>
<td>(HP 8753D standard only)</td>
<td>2-108</td>
</tr>
<tr>
<td>Tutorial Program Explanation</td>
<td>2-110</td>
</tr>
<tr>
<td>Initializing the System</td>
<td>2-110</td>
</tr>
<tr>
<td>Main Program</td>
<td>2-110</td>
</tr>
</tbody>
</table>

3. **HP BASIC Programming Examples**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>3-1</td>
</tr>
<tr>
<td>Required Equipment</td>
<td>3-1</td>
</tr>
<tr>
<td>Optional Equipment</td>
<td>3-2</td>
</tr>
<tr>
<td>System Setup and HP-IB Verification</td>
<td>3-2</td>
</tr>
<tr>
<td>HP 8753D/Option 011 Network Analyzer Instrument Control Using BASIC</td>
<td>3-4</td>
</tr>
<tr>
<td>Command Structure in BASIC</td>
<td>3-4</td>
</tr>
<tr>
<td>Command Interrogate</td>
<td>3-6</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-6</td>
</tr>
<tr>
<td>Held Commands</td>
<td>3-7</td>
</tr>
<tr>
<td>Operation Complete</td>
<td>3-7</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-8</td>
</tr>
<tr>
<td>Preparing for Remote (HP-IB) Control</td>
<td>3-8</td>
</tr>
<tr>
<td>I/O Paths</td>
<td>3-9</td>
</tr>
<tr>
<td>BASIC Programming Examples</td>
<td>3-10</td>
</tr>
<tr>
<td>Example 1: Measurement Setup</td>
<td>3-10</td>
</tr>
<tr>
<td>Example 1A: Setting Parameters</td>
<td>3-10</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-11</td>
</tr>
<tr>
<td>Example 1B: Verifying Parameters</td>
<td>3-12</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-13</td>
</tr>
<tr>
<td>Example 2: Calibration</td>
<td>3-14</td>
</tr>
<tr>
<td>Performing a Measurement Calibration</td>
<td>3-14</td>
</tr>
<tr>
<td>Example 2A: S11 1-Port Calibration</td>
<td>3-14</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-16</td>
</tr>
<tr>
<td>Example 2B: Full 2-Port Measurement Calibration</td>
<td>3-16</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-19</td>
</tr>
<tr>
<td>Example 3: Measurement Data Transfer</td>
<td>3-20</td>
</tr>
<tr>
<td>Example 3A: Data Transfer Using Markers</td>
<td>3-20</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-21</td>
</tr>
<tr>
<td>Example 3B: Data Transfer Using FORM 4 (ASCII Transfer)</td>
<td>3-22</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-25</td>
</tr>
<tr>
<td>Example 3C: Data Transfer Using Floating-Point Numbers</td>
<td>3-25</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-27</td>
</tr>
<tr>
<td>Example 3D: Data Transfer Using Frequency-Array Information</td>
<td>3-27</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-29</td>
</tr>
<tr>
<td>Example 3E: Data Transfer Using FORM 1, Internal-Binary Format</td>
<td>3-29</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-30</td>
</tr>
<tr>
<td>Example 4: Measurement Process Synchronization</td>
<td>3-31</td>
</tr>
<tr>
<td>Status Reporting</td>
<td>3-31</td>
</tr>
<tr>
<td>Example 4A: Using the Error Queue</td>
<td>3-31</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-32</td>
</tr>
<tr>
<td>Example 4B: Generating Interrupts</td>
<td>3-33</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-36</td>
</tr>
<tr>
<td>Example 4C: Power Meter Calibration</td>
<td>3-36</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-39</td>
</tr>
<tr>
<td>Example 5: Network Analyzer System Setups</td>
<td>3-40</td>
</tr>
<tr>
<td>Saving and Recalling Instrument States</td>
<td>3-40</td>
</tr>
<tr>
<td>Example 5A: Using the Learn String</td>
<td>3-40</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-41</td>
</tr>
<tr>
<td>Example 5B: Reading Calibration Data</td>
<td>3-42</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-44</td>
</tr>
<tr>
<td>Example 5C: Saving and Restoring the Analyzer Instrument State</td>
<td>3-45</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-47</td>
</tr>
<tr>
<td>Example 6: Limit-Line Testing</td>
<td>3-48</td>
</tr>
<tr>
<td>Using List-Frequency Mode</td>
<td>3-48</td>
</tr>
<tr>
<td>Example 6A: Setting Up a List-Frequency Sweep</td>
<td>3-48</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-50</td>
</tr>
<tr>
<td>Example 6B: Selecting a Single Segment from a Table of Segments</td>
<td>3-51</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-53</td>
</tr>
<tr>
<td>Using Limit Lines to Perform PASS/FAIL Tests</td>
<td>3-53</td>
</tr>
<tr>
<td>Example 6C: Setting Up Limit Lines</td>
<td>3-53</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-55</td>
</tr>
<tr>
<td>Example 6D: Performing PASS/FAIL Tests While Tuning</td>
<td>3-56</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-58</td>
</tr>
<tr>
<td>Example 7: Report Generation</td>
<td>3-59</td>
</tr>
<tr>
<td>Example 7A1: Operation Using Talker/Listener Mode</td>
<td>3-59</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-60</td>
</tr>
<tr>
<td>Example 7A2: Controlling Peripherals Using Pass-Control Mode</td>
<td>3-61</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-63</td>
</tr>
<tr>
<td>Example 7A3: Printing with the Serial Port</td>
<td>3-63</td>
</tr>
<tr>
<td>Example 7B1: Plotting to a File and Transferring File Data to a Plotter</td>
<td>3-65</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-67</td>
</tr>
<tr>
<td>Plotting to a File and Utilizing PC-Graphics Applications</td>
<td>3-67</td>
</tr>
<tr>
<td>Example 7B2: Reading Plot Files From a Disk</td>
<td>3-68</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-74</td>
</tr>
<tr>
<td>Example 7C: Reading ASCII Disk Files to the System Controller Disk File</td>
<td>3-75</td>
</tr>
<tr>
<td>Running the Program</td>
<td>3-78</td>
</tr>
</tbody>
</table>

Index
Figures

1-1. Data Processing Chain ........................................ 1-38
1-2. Key Codes ................................................... 1-41
2-1. HP-IB Bus Structure .......................................... 2-3
2-2. Analyzer Single Bus Concept .................................. 2-6
2-3. The Data-Processing Chain .................................... 2-21
2-4. Status Reporting Structure .................................... 2-24
2-5. Status Reporting Structure .................................... 2-62
3-1. The HP 8753D/Option 011 Network Analyzer System with Controller ........................................ 3-2

Tables

1-1. OPC-compatible Commands ...................................... 1-3
1-2. Key Select Codes ............................................... 1-9
1-3. HP-IB Only Commands .......................................... 1-29
1-4. Relationship Between Calibrations and Classes .............. 1-34
1-5. Calibration Arrays ............................................. 1-35
1-6. Units as a Function of Display Format ....................... 1-39
1-7. Disk File Names ................................................. 1-40
2-1. Code Naming Convention ....................................... 2-10
2-2. FORM 4 (ASCII) Data-Transfer Character Definitions .... 2-15
2-3. Units as a Function of Display Format ....................... 2-17
2-4. HP 8753D/Option 011 Network Analyzer Array-Data Formats 2-19
2-5. Status Bit Definitions .......................................... 2-25
3-1. Additional BASIC 6.2 Programming Information ........... 3-1
3-2. Additional HP-IB Information .................................. 3-1
3-3. HP 8753D/Option 011 Network Analyzer Array-Data Formats 3-22
## Contents

1. HP-IB Command Reference
   - HP-IB Capabilities ........................................ 1-2
   - Held Commands ........................................... 1-2
   - Operation Complete ...................................... 1-3
   - Command Interrogate ..................................... 1-3
   - Identification ............................................ 1-4
   - Output Queue ............................................. 1-4
   - Units ...................................................... 1-4
   - Input Data ................................................ 1-5
   - Input Syntax ............................................. 1-5
     - General Structure ...................................... 1-5
   - Valid Characters ......................................... 1-6
   - Programming Data ........................................ 1-7
   - Key Select Codes for the HP 8753D/Option 011 Network Analyzer ... 1-8
   - HP-IB Only Commands .................................... 1-29
   - Calibration ............................................... 1-34
   - Display Graphics ........................................ 1-36
   - User Graphics Units .................................... 1-36
   - HP-GL subset: ............................................ 1-36
     - Accepted but ignored HP-GL commands: .............. 1-37
   - Useful Tables and Figures ................................ 1-38
   - Disk File Names .......................................... 1-40
   - Alphabetical Code Listing ............................... 1-42

Index
### Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Data Processing Chain</td>
<td>1-38</td>
</tr>
<tr>
<td>1-2</td>
<td>Key Codes</td>
<td>1-41</td>
</tr>
</tbody>
</table>

### Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>OPC-compatible Commands</td>
<td>1-3</td>
</tr>
<tr>
<td>1-2</td>
<td>Key Select Codes</td>
<td>1-9</td>
</tr>
<tr>
<td>1-3</td>
<td>HP-IB Only Commands</td>
<td>1-29</td>
</tr>
<tr>
<td>1-4</td>
<td>Relationship Between Calibrations and Classes</td>
<td>1-34</td>
</tr>
<tr>
<td>1-5</td>
<td>Calibration Arrays</td>
<td>1-35</td>
</tr>
<tr>
<td>1-6</td>
<td>Units as a Function of Display Format</td>
<td>1-39</td>
</tr>
<tr>
<td>1-7</td>
<td>Disk File Names</td>
<td>1-40</td>
</tr>
</tbody>
</table>
HP-IB Command Reference

This chapter is a reference for operation of the HP 8753D/Option 011 network analyzer under HP-IB control. You should already be familiar with making measurements with the analyzer. Information about the HP-IB commands is organized as follows:

- overview of HP-IB, including:
  - HP-IB capabilities of the analyzer
  - held commands
  - operation complete (OPC) commands
  - interrogate commands
  - syntax
- key select codes arranged by front-panel hardkey
- HP-IB only commands
- measurement calibration sequence
- display graphics

For information about manual operation of the analyzer, refer to the HP 8753D/Option 011 Network Analyzer User's Guide.
**HP-IB Capabilities**

As defined by the IEEE 488.1 standard, the analyzer has the following capabilities:

- **SH1**: Full source handshake capability.
- **AH1**: Full acceptor handshake capability.
- **T6**: Can be a basic talker, answers serial poll, unaddresses if MLA is issued.
- **L4**: Acts as a basic listener and unaddresses if MTA is issued.
- **SR1**: Can issue service requests.
- **RL1**: Will do remote, local, and local lockout.
- **PP0**: Does not respond to parallel poll.
- **DC1**: Device clear capability.
- **DT1**: Will respond to device trigger in hold mode.
- **C1, C2, C3, C10**: No controller capabilities in talker/listener mode. System controller mode can be selected under the LOCAL menu. Pass control capability in pass control mode.
- **E2**: Tri-state drivers.

These codes are completely explained in the IEEE Std 488-1978 document, published by the Institute of Electrical and Electronic Engineers, Inc., 345 East 47th Street, New York, New York 11017.

**Held Commands**

The analyzer cannot process HP-IB commands while executing certain key commands, called held commands. Once a held command is received, the analyzer will read new commands into the input buffer, but it will not begin the execution of any commands until the completion of the held command. When the 15-character input buffer is full, the analyzer will hold off the bus until it is able to process the commands in the buffer.

**Note**

Commands that call a calibration class are held if there is just one standard in the class, since such commands trigger a measurement.
Operation Complete

You can synchronize programs with the execution of certain hold commands with the operation complete command. This function is enabled by issuing 0PC; or 0PC?; prior to an OPC-compatible command. The operation complete bit will then be set at the completion of the OPC-compatible command's execution. For example, issuing 0PC;SING; causes the OPC bit to be set when the single sweep is finished. Issuing 0PC?; causes the analyzer to output a one when the command execution is complete. Addressing the analyzer to talk after issuing 0PC?; will not cause an "addressed to talk without selecting output" error, but the analyzer will halt the computer by not transmitting the one until the command has completed. For example, issuing 0PC?;PRES; and then immediately interrogating the analyzer causes the bus to halt until the instrument preset is complete and the analyzer outputs a one.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAN1</td>
<td>FWDM^3</td>
</tr>
<tr>
<td>CHAN2</td>
<td>FWDT^3</td>
</tr>
<tr>
<td>CLASS11A^1</td>
<td>HARMOFF</td>
</tr>
<tr>
<td>CLASS11B^1</td>
<td>HARMSEC</td>
</tr>
<tr>
<td>CLASS11C^1</td>
<td>HARMTHIR</td>
</tr>
<tr>
<td>CLASS22A^1</td>
<td>INSMEXSA</td>
</tr>
<tr>
<td>CLASS22B^1</td>
<td>INSMEXSM</td>
</tr>
<tr>
<td>CLASS22C^1</td>
<td>INSMNETA</td>
</tr>
<tr>
<td>CLEARALL</td>
<td>INSMTUNR</td>
</tr>
<tr>
<td>DATI</td>
<td>ISOD</td>
</tr>
<tr>
<td>DONE</td>
<td>MANTRIG</td>
</tr>
<tr>
<td>EDITDONE</td>
<td>NOOP</td>
</tr>
<tr>
<td>EXTTOFF</td>
<td>NUMG</td>
</tr>
<tr>
<td>EXTTON</td>
<td>PRES</td>
</tr>
<tr>
<td>EXTTPOIN</td>
<td>RAID</td>
</tr>
<tr>
<td>FREQOFFS&lt;ON</td>
<td>OFF&gt;</td>
</tr>
<tr>
<td>FWDI^1</td>
<td>RAISOL^1</td>
</tr>
</tbody>
</table>

1 The class commands are OPC-compatible if there is only one standard in the class.

Command Interrogate

To interrogate one of the front-panel-equivalent commands listed in Table 1-2, append a question mark to the command root. This causes the analyzer to output the state of that function as a single number in ASCII format, form 4. If the function is a settable function, such as power or sweep time, the analyzer will output the current value of that function. If the function is either ON/OFF or one selection of several, the analyzer outputs a one for ON or selected, and a zero for OFF. If a command that does not have a defined response is interrogated, the analyzer outputs a zero. Interrogating a function does not put it in the active entry area.
Identification

The analyzer's response to IDN? is "HEWLETT PACKARD,8753D,0,X.XX" where X.XX is the firmware revision of the instrument.

Output Queue

Whenever a command to output data is received, the analyzer puts the data into the output queue to be copied out by the next read operation. The queue, however, is only one event long: the next command to output data will overwrite the data already in the queue. Therefore, it is important to read the output queue immediately after every interrogation or request to output data.

Units

The analyzer outputs data in basic units such as Hz, dB, seconds, ohms, etc.

Input data is assumed to be in basic units unless one of the following units expressions qualifies the data input (upper and lower case are equivalent):

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
<th>Symbol</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Seconds</td>
<td>HZ</td>
<td>Hertz</td>
</tr>
<tr>
<td>MS</td>
<td>Milliseconds</td>
<td>KHz</td>
<td>Kiloertz</td>
</tr>
<tr>
<td>US</td>
<td>Microseconds</td>
<td>MHz</td>
<td>Megahertz</td>
</tr>
<tr>
<td>NS</td>
<td>Nanoseconds</td>
<td>GHZ</td>
<td>Gigahertz</td>
</tr>
<tr>
<td>PS</td>
<td>Picoseconds</td>
<td>DB</td>
<td>dB or dBm</td>
</tr>
<tr>
<td>FS</td>
<td>Femtoseconds</td>
<td>V</td>
<td>Volts</td>
</tr>
</tbody>
</table>
Input Data

Input Syntax
The HP-IB commands accepted by the analyzer can be grouped into four input syntax types. The analyzer does not distinguish between upper and lower case letters.

General Structure
[code][appendage] [data] [unit] [terminator]

[code] The root mnemonic (these codes are described in the "Alphabetical Code Listing" later in this chapter.)

[appendage] A qualifier attached to the root mnemonic. Possible appendages are ON or OFF, which toggle a function ON or OFF, or integers, which specify one option out of several. There can be no spaces or symbols between the code and the appendage.

[data] A single operand used by the root mnemonic, usually to set the value of a function. The data can be a number or a character string. Numbers are accepted as integers or decimals, with power of ten specified by E, as in STAR 0.2E+10, which sets the start frequency to 2 GHz. Character strings must be enclosed by quotation marks at the beginning and ending of the string.

[unit] The units of the operand, if applicable. If no units are specified, the analyzer assumes the basic units as described under "Units" earlier in this chapter. The data is entered into the function when either units or a terminator is received.

[terminator] Indicates the end of the command, enters the data, and turns OFF the active entry area. The terminator should be a semicolon. Terminators are not necessary for the analyzer to interpret commands correctly, but in the case of a syntax error, the analyzer will attempt to recover at the next terminator. The analyzer also interprets line feeds and HP-IB END OR IDENTIFY (EOI) messages as terminators.
The specific syntaxes are as follows:

**SYNTAX TYPE 1:** [code] [terminator]

These are simple action commands that require no complementary information, such as AUTO; (auto scales the active channel.)

**SYNTAX TYPE 2:** [code][appendage] [terminator]

These are simple action commands requiring limited customization, such as CORRON;, CORROFF; (turn error correction ON or OFF;) or RECA1;, RECA2;, RECA3; ... (recall register 1, 2, 3 ... ) There can be no characters or symbols between the code and the appendage. In the following cases: CLEAR[EG][D], RE[EC][A][D][D], SAV[ER][EG][D], and TTL[REG][D], [D] must be 2 characters. For example, CLEAR[EG]01; is correct; CLEAR[EG]1; is not.

**SYNTAX TYPE 3:** [code] [data] [unit] [terminator]

These are data input commands such as STAR 1.0 GHZ; (set the start frequency to 1 GHz).

**SYNTAX TYPE 4:** [code][appendage] [data] [terminator]

These are titling and marker commands that have an appendage, such as TITL “FILTER” (enter FILTER as the CRT title), TITR1 “STATE1”, TITR2 “EMPTY” (title register 1 STATE1, title register 2 EMPTY.)

**INTERROGATE SYNTAX:** [code][?]?

To interrogate a front-panel-equivalent function, append a question mark to the root mnemonic (for example: POWE?, AVERO?, or REAL?). To interrogate commands with integer appendages, place the question mark after the appendage.

**Valid Characters**

The analyzer will accept letters, changing lower case to upper case, numbers, decimal points, +, -, semicolons, carriage returns and line feeds. Leading zeros, spaces, carriage returns, and unnecessary terminators are ignored, except when inserted into or between a mnemonic and/or an appendage. If the analyzer does not recognize a character as appropriate, it generates a syntax error message and recovers at the next terminator.
Programming Data

The front-panel-equivalent command mnemonics are listed in Table 1-2. These commands perform the same function as a front-panel key, and are arranged functionally by front-panel key. Table 1-3 lists functions that have no logical equivalent in manual operation. They concern data transmission, status reporting, and special HP-IB functions. Following these tables is a list of all these mnemonics with their descriptions.

In most cases, the commands were named following these rules:

1. Simple commands are the first four letters of the function they control, as in POWE. If the function label is two words, the first three mnemonic letters are the first three letters of the first word, and the fourth mnemonic letter is the first letter of the second word. For example, ELED is derived from electrical delay.

2. If there are many commands grouped together in a class, as in markers or plotting pen numbers, the command is increased to eight letters. The first four letters are the class label derived using rule one. The last four letters are the function specifier, again derived using rule one. An example of this is the class pen numbers PENN, which is used with several functions such as PENNDATA, PENNMEMO.

These rules were not always followed in order to maintain compatibility with other products, to make commands more meaningful and easier to remember, and when technical considerations prevented their use.
Key Select Codes for the HP 8753D/Option 011
Network Analyzer

The HP-IB mnemonics in this table are functionally arranged by their front-panel key equivalent.

<table>
<thead>
<tr>
<th>Keys</th>
<th>Page Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG</td>
<td>9</td>
</tr>
<tr>
<td>CAL-Error correction, calibration</td>
<td>9</td>
</tr>
<tr>
<td>CAL-Calibration kits</td>
<td>11</td>
</tr>
<tr>
<td>CAL-Power meter calibration</td>
<td>13</td>
</tr>
<tr>
<td>CHANNEL</td>
<td>14</td>
</tr>
<tr>
<td>COPY</td>
<td>14</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>16</td>
</tr>
<tr>
<td>ENTRY</td>
<td>17</td>
</tr>
<tr>
<td>FORMAT</td>
<td>17</td>
</tr>
<tr>
<td>LOCAL</td>
<td>18</td>
</tr>
<tr>
<td>MEAS</td>
<td>19</td>
</tr>
<tr>
<td>MENU (stimulus)</td>
<td>20</td>
</tr>
<tr>
<td>MARKER</td>
<td>21</td>
</tr>
<tr>
<td>MARKER FCTN</td>
<td>22</td>
</tr>
<tr>
<td>SAVE/RECALL, internal registers</td>
<td>23</td>
</tr>
<tr>
<td>SAVE/RECALL, disk files</td>
<td>24</td>
</tr>
<tr>
<td>SCALE REF</td>
<td>24</td>
</tr>
<tr>
<td>SEQ-Sequencing</td>
<td>25</td>
</tr>
<tr>
<td>STIMULUS</td>
<td>26</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>27</td>
</tr>
<tr>
<td>SYSTEM-Limit testing</td>
<td>28</td>
</tr>
<tr>
<td>SYSTEM-Transform</td>
<td></td>
</tr>
</tbody>
</table>

Column headings:

Function: The front-panel function affected by the mnemonic.
Action: The effects of the mnemonic on that function.
Mnemonic: The HP-IB mnemonic.
S: Syntax type. See “Input Syntax.”
?: Interrogate response. If a response is defined, it is listed.
O: OPC-compatible command.
Range: The range of acceptable inputs and corresponding units.

Symbol conventions:

[ ] An optional operand.
D A numerical operand.
I An integer appendage that is part of the command. For example, CLEA<i>, where i = 1 to 5, indicates that the actual commands are CLEA1, CLEA2, CLEA3, CLEA4, and CLEA5.
$ A character-string operand which must be enclosed by quotes.
<> A necessary appendage.
| An either/or choice in appendages.
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Averaging</td>
<td>Restart</td>
<td>AVERREST</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Factor</td>
<td>AVERFACT[D]</td>
<td></td>
<td>3</td>
<td>D</td>
<td></td>
<td>0 to 999</td>
</tr>
<tr>
<td>On/off</td>
<td>AVER&lt;ON</td>
<td>OFF&gt;</td>
<td></td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td>Smoothing</td>
<td>Set aperture</td>
<td>SMOOAPER[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0.05 to 20%</td>
</tr>
<tr>
<td>On/off</td>
<td>SMOO&lt;ON</td>
<td>OFF&gt;</td>
<td></td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td>IF bandwidth</td>
<td>Set bandwidth</td>
<td>IFBW[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>10, 30, 100, 300,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1000, 3000 Hz</td>
</tr>
</tbody>
</table>

**CAL-error correction, calibration**

<p>| Correction                   | On/off          | CORR&lt;ON|OFF&gt;   | 2 | 1,0|    |                     |
| Interpolative correction     | On/off          | COR&lt;ON|OFF&gt;  | 2 | 1,0|    |                     |
| Cal sequence                 | Resume          | RESC   | 1 |    |    |                     |
| Port extensions              | Port 1          | PORT1[D] | 3 | D |    | ±10 s               |
|                              | Port 2          | PORT2[D] | 3 | D |    | ±10 s               |
|                              | Input A         | PORTA[D] | 3 | D |    | ±10 s               |
|                              | Input B         | PORTB[D] | 3 | D |    | ±10 s               |
|                              | Off             | PO&lt;ON|OFF&gt;  | 2 | 1,0|    |                     |
| Velocity factor              | Set value       | VELOFACT[D]    | 3 | D |    | 0 to 10             |
| ZO                           | Set Value       | SETZ[D]       | 3 | D |    | 0.1 to 600Ω        |
| Test set switching           | Continuous/full 2-port cal | CSWION | 2 | 1,0|    |                     |
|                              | Hold/quasi 2-port cal | CSWIOFF | 2 | 1,0|    |                     |
| Begin cal sequence           | Response        | CALIRESP      | 1 |    | 0,1|                     |
|                              | Response and Isol | CALIRAI | 1 |    | 0,1|                     |
|                              | S11 1-port      | CALIS111      | 1 |    | 0,1|                     |
|                              | S22 1-port      | CALIS221      | 1 |    | 0,1|                     |
|                              | Full 2-port     | CALIFUL2      | 1 |    | 0,1|                     |
|                              | One path 2-port | CALIONE2      | 1 |    | 0,1|                     |
|                              | TRL*/LRM* 2-port | CALITRL2     | 1 |    | 0,1|                     |
|                              | None            | CALN          | 1 |    | 0,1|                     |
| Sweep mode                   | Chop            | CHOPAB        | 1 |    | 0,1|                     |
|                              | Alternate       | ALTAB         | 1 |    | 0,1|                     |
| Intermediate cal steps, 1 path/2-port | Isolation | ISOOP | 1 |    |    |                     |
|                              | Reflection      | REFOP         | 1 |    |    |                     |
|                              | Transmission    | TRAOP         | 1 |    |    |                     |</p>
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intermediate cal step, 2-port cal</td>
<td>Transmission</td>
<td>TRAN</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reflection (incl. Line in TRL)</td>
<td>REFL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Isolation</td>
<td>ISOL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select response &amp; isolation class</td>
<td>Response</td>
<td>RAIRESF</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Isolation</td>
<td>RAIISOL</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td>Select reflection class</td>
<td>S11A (open)</td>
<td>CLASS11A</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S11B (short)</td>
<td>CLASS11B</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S11C (load)</td>
<td>CLASS11C</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S22A (open)</td>
<td>CLASS22A</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S22B (short)</td>
<td>CLASS22B</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S22C (load)</td>
<td>CLASS22C</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td>Select transmission class</td>
<td>Fwd transmission</td>
<td>FWDT</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rev transmission</td>
<td>REV T</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fwd match</td>
<td>FWDM</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rev match</td>
<td>REVM</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td>Select isolation class</td>
<td>Forward isolation</td>
<td>FWDI</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reverse isolation</td>
<td>REVI</td>
<td>1</td>
<td></td>
<td>OPC(^1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omit isolation</td>
<td>OMII</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select standard in class</td>
<td>Standard A</td>
<td>STANA</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Standard B</td>
<td>STANB</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Standard C</td>
<td>STANC</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Standard D</td>
<td>STAND</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Standard E</td>
<td>STANE</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Standard F</td>
<td>STANF</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Standard G</td>
<td>STANG</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td>Sliding load</td>
<td>Set</td>
<td>SLIS</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Done</td>
<td>SLID</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Done with:</td>
<td>Class</td>
<td>DONE</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Isolation</td>
<td>ISOD</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reflection</td>
<td>REFD</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transmission</td>
<td>TRAD</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\) The class commands are OPC-compatible if there is only one standard in the class. If there is just one standard, that standard is measured automatically. If there is more than one standard in the class, the class command only calls another menu. Also, the reflection classes "B" and "C" (e.g., S11B, S11C) are actually transmission measurements of the "LINE" when the TRL calibration type is selected. The transmission classes are used for the "THRU" when TRL calibration is selected.
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save cal</td>
<td>Response</td>
<td>RESPDONE</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td>Resp and isol</td>
<td>RAID</td>
<td></td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td>1-port cal</td>
<td>SAV1</td>
<td></td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td>2-port cal</td>
<td>SAV2</td>
<td></td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
</tbody>
</table>

**CAL-calibration kits**

<table>
<thead>
<tr>
<th>Select default kits</th>
<th>7 mm</th>
<th>CALK7MM</th>
<th>1</th>
<th>1,0</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.5 mmC</td>
<td>CALK35MM</td>
<td>1</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.5 mmD</td>
<td>CALK35MD</td>
<td>1</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type N, 50 ohm</td>
<td>CALKN50</td>
<td>1</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type N, 75 ohm</td>
<td>CALKN75</td>
<td>1</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User-defined</td>
<td>CALKUSED</td>
<td>1</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Modify kit**

<table>
<thead>
<tr>
<th>Modify kit</th>
<th>Modify current</th>
<th>MODII</th>
<th>1</th>
</tr>
</thead>
</table>

**Define standard number (begin standard definition)**

<table>
<thead>
<tr>
<th>Define standard type</th>
<th>Open</th>
<th>STDTOPEN</th>
<th>1</th>
<th>1,0</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Short</td>
<td>STDTSHDR</td>
<td>1</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Load</td>
<td>STDTLDR</td>
<td>1</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delay/thru</td>
<td>STDTDLA</td>
<td>1</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arbitrary imped.</td>
<td>STDTARBI</td>
<td>1</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Define standard parameters**

<table>
<thead>
<tr>
<th>Define standard offsets</th>
<th>Delay</th>
<th>OFSD[D]</th>
<th>3</th>
<th></th>
<th>±1 s</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Loss</td>
<td>OFSL[D]</td>
<td>3</td>
<td></td>
<td>0 to 1000 T/s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Z0</td>
<td>OFSZ[D]</td>
<td>3</td>
<td></td>
<td>0.1 to 5002</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Min. frequency</td>
<td>MINF[D]</td>
<td>3</td>
<td></td>
<td>0 to 1000 GHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Max. frequency</td>
<td>MAXF[D]</td>
<td>3</td>
<td></td>
<td>0 to 1000 GHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Coaxial</td>
<td>COAX</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Waveguide</td>
<td>WAVE</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Standard done**

<table>
<thead>
<tr>
<th>Standard done</th>
<th>Standard defined</th>
<th>STDD</th>
<th>1</th>
</tr>
</thead>
</table>

**Label standard**

<p>| Label standard | LABS[$] | 4 | 10 character |</p>
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify class</td>
<td>Response</td>
<td>SPECRESP[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>Resp &amp; Isol</td>
<td>SPECRESP[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>SIIA (open)</td>
<td>SPECSI11A[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>SIIB (short)</td>
<td>SPECSI11B[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>SIIIC (load)</td>
<td>SPECSI11C[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>S22A (open)</td>
<td>SPECSI22A[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>S22B (short)</td>
<td>SPECSI22B[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>S22C (load)</td>
<td>SPECSI22C[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>Forward Trans</td>
<td>SPECFWDT[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>Forward Match</td>
<td>SPECTFDWI[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>Reverse Trans</td>
<td>SPECREVT[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>Reverse Match</td>
<td>SPECREVM[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>TRL, Reflect, Forward, Match</td>
<td>SPECTRFM[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>TRL, Reflect, Reverse, Match</td>
<td>SPECTRFM[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>TRL, Line, Forward, Match</td>
<td>SPECTLFM[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>TRL, Line, Forward, Trans</td>
<td>SPECTLFT[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>TRL, Line, Reverse, Match</td>
<td>SPECTLRFM[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>TRL, Line, Reverse, Trans</td>
<td>SPECTLRT[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>TRL, Thru, Forward, Match</td>
<td>SPECTTFM[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>TRL, Thru, Forward, Trans</td>
<td>SPECTTFT[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>TRL, Thru, Reverse, Match</td>
<td>SPECTTRM[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td></td>
<td>TRL, Thru, Reverse, Trans</td>
<td>SPECTTRT[I, I... ]</td>
<td>3</td>
<td></td>
<td>Std numbers</td>
</tr>
<tr>
<td>Class done</td>
<td></td>
<td>CLAD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Label class</td>
<td>Response</td>
<td>LABERESP[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>Resp. &amp; isolation</td>
<td>LABERESP[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>SIIA</td>
<td>LABESI1A[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>SIIB</td>
<td>LABESI1B[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>SIIIC</td>
<td>LABESI1C[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>S22A</td>
<td>LABESI22A[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>S22B</td>
<td>LABESI22B[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>S22C</td>
<td>LABESI22C[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>Forward Trans</td>
<td>LABEFWDT[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>Forward Match</td>
<td>LABEFWDM[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>Reverse Trans</td>
<td>LABERREV[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>Reverse Match</td>
<td>LABEREV[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>TRL, Reflect, Forward, Match</td>
<td>LABETRFM[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>TRL, Reflect, Reverse, Match</td>
<td>LABETRM[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>TRL, Line, Forward, Match</td>
<td>LABETFEM[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>TRL, Line, Forward, Trans</td>
<td>LABETLPT[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>TRL, Line, Reverse, Match</td>
<td>LABETLRM[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>TRL, Line, Reverse, Trans</td>
<td>LABETLRT[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>TRL, Thru, Forward, Match</td>
<td>LABETTFM[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>TRL, Thru, Forward, Trans</td>
<td>LABETTFT[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>TRL, Thru, Reverse, Match</td>
<td>LABETTRM[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td></td>
<td>TRL, Thru, Reverse, Trans</td>
<td>LABETTRT[$]</td>
<td>4</td>
<td></td>
<td>10 character</td>
</tr>
</tbody>
</table>

1-12 HP-IB Command Reference
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label kit</td>
<td></td>
<td>LABK[S]</td>
<td>4</td>
<td></td>
<td></td>
<td>10 character</td>
</tr>
<tr>
<td>Kit done</td>
<td></td>
<td>KITD</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save kit</td>
<td>Into user kit</td>
<td>SAVEUSEK</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CAL-power meter calibration**

<table>
<thead>
<tr>
<th>Power meter cal</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>PWMCOFF[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td></td>
<td>Cal power:</td>
</tr>
<tr>
<td>Each sweep</td>
<td>PWMCEACS[D]</td>
<td>3</td>
<td>0,1</td>
<td></td>
<td></td>
<td>-100 to 100 dB</td>
</tr>
<tr>
<td>One sweep</td>
<td>PWMCONES[D]</td>
<td>3</td>
<td>0,1</td>
<td></td>
<td></td>
<td>Cal power:</td>
</tr>
<tr>
<td>Take cal sweep 1</td>
<td>TAKC</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>-100 to 100 dB</td>
</tr>
<tr>
<td>Number of readings</td>
<td>NUMR[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td></td>
<td>1 to 100</td>
</tr>
<tr>
<td>Set port cal pwr</td>
<td>PWRMCA[D]</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>Cal power:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-100 to 100 dB</td>
</tr>
</tbody>
</table>

**Edit power loss table**

<table>
<thead>
<tr>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>On/off</td>
<td>PWRLOSS&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit list</td>
<td>POWLST</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use sensor A or B</td>
<td>USES&lt;ENSA/ENSB&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add segment</td>
<td>SADD</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit segment N</td>
<td>SEDI[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>1 to 12</td>
</tr>
<tr>
<td>Done with segment</td>
<td>SDON</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete segment</td>
<td>SDEL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Done</td>
<td>EDITDONE</td>
<td>1</td>
<td></td>
<td></td>
<td>OPC</td>
</tr>
<tr>
<td>Clear list</td>
<td>CLEL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Edit power loss segment**

<table>
<thead>
<tr>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>POWLFREQ[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range 2</td>
</tr>
<tr>
<td>Value</td>
<td>POWLLOSS[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>-9900 to 9900 dB</td>
</tr>
</tbody>
</table>

**Edit cal sensor table**

<table>
<thead>
<tr>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit sensor menu A</td>
<td>CALFSENA</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit sensor menu B</td>
<td>CALFSENB</td>
<td>1</td>
<td></td>
<td></td>
<td>HP 438A only</td>
</tr>
<tr>
<td>Add segment</td>
<td>SADD</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit segment N</td>
<td>SEDI[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>1 to 12</td>
</tr>
<tr>
<td>Done with segment</td>
<td>SDON</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete segment</td>
<td>SDEL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Done</td>
<td>EDITDONE</td>
<td>1</td>
<td></td>
<td></td>
<td>OPC</td>
</tr>
<tr>
<td>Clear list</td>
<td>CLEL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 Requires pass-control mode when using the HP-IB port.
2 For frequency sweeps: 30 kHz to 3 GHz. (To 6 GHz for Option 006). For power sweeps: -15 to 20 dBm in range 0, 25 dB maximum in other ranges (100 to 100 with power meter cal on). For CW time: 0 to 24 hours. For frequency sweep, transform on: ± 1/frequency step. For CW time sweep, transform on: ±1/time step.
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>？</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit cal sensor</td>
<td>Frequency</td>
<td>CALFFREQ[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td>segment</td>
<td>Cal factor</td>
<td>CALPCALF[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0 to 200%</td>
</tr>
<tr>
<td>Channel</td>
<td>CH 1 active</td>
<td>CHAN1</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CH 2 active</td>
<td>CHAN2</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td>COPY</td>
<td>To printer&lt;sup&gt;2&lt;/sup&gt;</td>
<td>PRINALL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>To plotter&lt;sup&gt;2&lt;/sup&gt;</td>
<td>PLOT</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>To plotter</td>
<td>OUTPPLLOT</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>To printer</td>
<td>OUTPRPRIN</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Printer</td>
<td>Auto feed</td>
<td>PRTTRAUTF&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td>Printer</td>
<td>Form feed</td>
<td>PRTTRFORF</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Printer setup</td>
<td>Default</td>
<td>DEFLPRINT</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plotter</td>
<td>Auto feed</td>
<td>PLTTRAUTF&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td>Plotter</td>
<td>Form feed</td>
<td>PLTRFORF</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plotter setup</td>
<td>Default</td>
<td>DFLT</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>List values</td>
<td></td>
<td>LISV</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating parameters</td>
<td></td>
<td>OPEP</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print list values or</td>
<td>Raster display dump to HP-IB&lt;sup&gt;2&lt;/sup&gt;</td>
<td>PRINTALL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>operating and marker</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>parameters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Next page</td>
<td></td>
<td>NEXP</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Restore display</td>
<td></td>
<td>RESD</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select print color</td>
<td>Monochrome</td>
<td>PRIS</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Color</td>
<td>PRIC</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<sup>1</sup> For frequency sweeps: 30 kHz to 3 GHz. (To 6 GHz for Option 006). For power sweeps: -15 to 20 dBm in range 0, 25 dB maximum in other ranges (-100 to +100 with power meter cal on). For CW time: 0 to 24 hours. For frequency sweep, transform on: ± 1/frequency step. For CW time sweep, transform on: ± 1/time step.

<sup>2</sup> Requires pass-control mode when using the HP-IB port.
Table 1-2. Key Select Codes (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print feature color</td>
<td>Data channel 1</td>
<td>PCOLDATA1&lt;color&gt;</td>
<td>2</td>
<td>?</td>
<td>O</td>
<td>Colors&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td>Data channel 2</td>
<td>PCOLDATA2&lt;color&gt;</td>
<td>2</td>
<td>?</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory channel 1</td>
<td>PCOLMEMO1&lt;color&gt;</td>
<td>2</td>
<td>?</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory channel 2</td>
<td>PCOLMEMO2&lt;color&gt;</td>
<td>2</td>
<td>?</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graticule</td>
<td>PCOLGRAT&lt;color&gt;</td>
<td>2</td>
<td>?</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td>PCOLTEXT&lt;color&gt;</td>
<td>2</td>
<td>?</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Warning</td>
<td>PCOLWARN&lt;color&gt;</td>
<td>2</td>
<td>?</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quadrant</td>
<td>Left lower</td>
<td>LEFT</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left upper</td>
<td>LEFU</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right lower</td>
<td>RIGL</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right upper</td>
<td>RIGU</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full page</td>
<td>FULP</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Features to be plotted</td>
<td>Data</td>
<td>PDATA&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>PMEM&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graticule</td>
<td>PGRAT&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td>PTEXT&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marker</td>
<td>PMKR&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pen number</td>
<td>Data</td>
<td>PENNDATA[D]</td>
<td>3</td>
<td></td>
<td>0, 1, 2 ... 10</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>PENNMEMO[D]</td>
<td>3</td>
<td></td>
<td>0, 1, 2 ... 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graticule</td>
<td>PENNGRAT[D]</td>
<td>3</td>
<td></td>
<td>0, 1, 2 ... 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td>PENNTXT[D]</td>
<td>3</td>
<td></td>
<td>0, 1, 2 ... 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marker</td>
<td>PENNMARK[D]</td>
<td>3</td>
<td></td>
<td>0, 1, 2 ... 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line type</td>
<td>Data</td>
<td>LINTDATA[D]</td>
<td>3</td>
<td></td>
<td>0, 1, 2 ... 7</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>LINTMEMO[D]</td>
<td>3</td>
<td></td>
<td>0, 1, 2 ... 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plot scale</td>
<td>Full page</td>
<td>SCAPFULL</td>
<td>1</td>
<td></td>
<td>0, 1, 2 ... 7</td>
<td></td>
</tr>
<tr>
<td>Graticule to p1,p2</td>
<td>SCAPGRAT</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plot speed</td>
<td>Slow</td>
<td>PLOSSLOW</td>
<td>1</td>
<td></td>
<td>0, 1, 2 ... 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fast</td>
<td>PLOSSFAST</td>
<td>1</td>
<td></td>
<td>0, 1, 2 ... 7</td>
<td></td>
</tr>
</tbody>
</table>

1 Colors = white|cyan|magenta|blue|yellow|green|red|black
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DISPLAY</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channels</td>
<td>Dual on/off</td>
<td>DUAC&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Split on/off</td>
<td>SPLD&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D2/D1 to D2</td>
<td>D1DIVD2&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td>Display</td>
<td>Data</td>
<td>DISPDATA</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Memory only</td>
<td>DISPMEMO</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data and mem</td>
<td>DISPDATM</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data/mem</td>
<td>DISPDDM</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data — mem</td>
<td>DISPDMM</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data to mem</td>
<td>DATI</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>OPC</td>
</tr>
<tr>
<td>Beeper</td>
<td>On done</td>
<td>BEEPDONE&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>On warning message</td>
<td>BEEPWARN&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td>CRT</td>
<td>Intensity</td>
<td>INTE[D]</td>
<td>3</td>
<td>D</td>
<td>0</td>
<td>to 100</td>
</tr>
<tr>
<td></td>
<td>Focus</td>
<td>FOCU[D]</td>
<td>3</td>
<td>D</td>
<td>0</td>
<td>to 100</td>
</tr>
<tr>
<td></td>
<td>Title</td>
<td>TITL[S]</td>
<td>4</td>
<td>$</td>
<td>48</td>
<td>character</td>
</tr>
<tr>
<td>Frequency</td>
<td>notation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>notation</td>
<td>Blank</td>
<td>PRECO</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adjust display</td>
<td>Background intensity</td>
<td>BACI[D]</td>
<td>3</td>
<td>D</td>
<td>0</td>
<td>to 100</td>
</tr>
<tr>
<td></td>
<td>Save colors</td>
<td>SVCO</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Recall colors</td>
<td>RECO</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default colors</td>
<td>DEFC</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify colors</td>
<td>Ch 1 data/lim in</td>
<td>COLOCH1D</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ch 1 memory</td>
<td>COLOCH1M</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ch 2 data/lim in</td>
<td>COLOCH2D</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ch 2 memory</td>
<td>COLOCH2M</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Graticule</td>
<td>COLOGRAT</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>COLOTEXT</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Warning</td>
<td>COLOWARN</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adjust color</td>
<td>Brightness</td>
<td>CBRI[D]</td>
<td>3</td>
<td>D</td>
<td>0</td>
<td>to 100</td>
</tr>
<tr>
<td></td>
<td>Color</td>
<td>COLOR[D]</td>
<td>3</td>
<td>D</td>
<td>0</td>
<td>to 100</td>
</tr>
<tr>
<td></td>
<td>Tint</td>
<td>TINT[D]</td>
<td>3</td>
<td>D</td>
<td>0</td>
<td>to 100</td>
</tr>
<tr>
<td></td>
<td>Reset</td>
<td>RSCO</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Action</td>
<td>Mnemonic</td>
<td>S</td>
<td>?</td>
<td>O</td>
<td>Range</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>----------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>-------------</td>
</tr>
<tr>
<td>ENTRY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step keys</td>
<td>Up</td>
<td>UP</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Down</td>
<td>DOWN</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entry off</td>
<td></td>
<td>ENTO</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORMAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Format</td>
<td>Log mag</td>
<td>LOGM</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phase</td>
<td>PHAS</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delay</td>
<td>DELA</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Smith chart</td>
<td>SMIC</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Polar</td>
<td>POLA</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lin mag</td>
<td>LINM</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Real</td>
<td>REAL</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Imaginary</td>
<td>IMAG</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SWR</td>
<td>SWR</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HP-IB modes</td>
<td>Talker/listener</td>
<td>TALKLIST</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Use pass control</td>
<td>USEPASC</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Debug</td>
<td>Display commands</td>
<td>DEBU&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td>Disk drive</td>
<td>Unit</td>
<td>DISCUNIT[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0 to 30</td>
</tr>
<tr>
<td></td>
<td>Volume</td>
<td>DISCVOL[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0 to 30</td>
</tr>
<tr>
<td>HP-IB addresses</td>
<td>Plotter</td>
<td>ADDR&lt;PLOT[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0 to 30</td>
</tr>
<tr>
<td></td>
<td>Printer</td>
<td>ADDRPRIN[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0 to 30</td>
</tr>
<tr>
<td></td>
<td>Disk drive</td>
<td>ADDRDISC[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0 to 30</td>
</tr>
<tr>
<td></td>
<td>Controller</td>
<td>ADDRCNT[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0 to 30</td>
</tr>
<tr>
<td>Power meter</td>
<td>Address</td>
<td>ADDRPOWM[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0 to 30</td>
</tr>
<tr>
<td></td>
<td>Type</td>
<td>POWM&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td>Plotter type</td>
<td>Plotter</td>
<td>PLTTYPFLTR</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HPGL printer</td>
<td>PLTTYHPGL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Printer type</td>
<td>ThinkJet</td>
<td>PRNTYPEJ</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DeskJet</td>
<td>PRNTYPDJ</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LaserJet</td>
<td>PRNTYPLJ</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PaintJet</td>
<td>PRNTYPPJ</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Epson-P2</td>
<td>PRNTYPEP</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Printer port</td>
<td>HP-IB</td>
<td>PRNPRTHPIB</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Parallel</td>
<td>PRNPRT PARA</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Serial</td>
<td>PRNPRTSERI</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 1-2. Key Select Codes (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plotter port</td>
<td>HP-IB</td>
<td>PLTPRTHPIB</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Parallel</td>
<td>PLTPRTPARA</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Serial</td>
<td>PLTPRTSERI</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Disk</td>
<td>PLTPRTDISK</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Printer serial port</td>
<td>Baud rate</td>
<td>PRNTRBAUD[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>1200, 2400, 4800, 9600, 19200</td>
</tr>
<tr>
<td>Printer serial port</td>
<td>Handshake</td>
<td>PRNHNDSHK&lt;DIR;XON&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td>1 = DTR-DSR 0 = XON-XOFF</td>
</tr>
<tr>
<td>Plotter serial port</td>
<td>Baud rate</td>
<td>PLTRBAUD[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>1200, 2400, 4800, 9600, 19200</td>
</tr>
<tr>
<td>Plotter serial port</td>
<td>Handshake</td>
<td>PLTHNDSHK&lt;DIR;XON&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td>1 = DTR-DSR 0 = XON-XOFF</td>
</tr>
<tr>
<td>Parallel port</td>
<td>Configure</td>
<td>PARAL&lt;GPIO;CPY&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td>1 = GPIO (General Purpose I/O), 0 = CPY (COPY use)</td>
</tr>
</tbody>
</table>

### MEAS

<table>
<thead>
<tr>
<th>Input ports</th>
<th>Action</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A/B</td>
<td>AR</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B/R</td>
<td>BR</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A/B</td>
<td>AB</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>MEASA</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>MEASB</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>MEASR</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selects test port 1 or 2</td>
<td>TSTP&lt;P1,P2&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td></td>
<td></td>
<td>1 = P1, 0 = P2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S-parameters</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S11</td>
<td>S11</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S12</td>
<td>S12</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S21</td>
<td>S21</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S22</td>
<td>S22</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Conversion to alternate parameters

<table>
<thead>
<tr>
<th>Conversion to alternate parameters</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>CONVOFF</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z:reflection</td>
<td>CONVZREF</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z:transmission</td>
<td>CONVZTRA</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y:reflection</td>
<td>CONVYREF</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y:transmission</td>
<td>CONVYTRA</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/S</td>
<td>CONVIDS</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analog input</td>
<td>ANAI[D]</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 Syntax type 1 when ANABOFF, Syntax type 3, and range = 1 to 31, when ANABON.
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Power</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>Level</td>
<td>POWER[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>-85 to -20 dBm</td>
</tr>
<tr>
<td></td>
<td>Trip</td>
<td>POWER&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Always couple power</td>
<td>COUP&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port power coupling</td>
<td>PORTP&lt;CP</td>
<td>PLD</td>
<td>UNC</td>
<td>PLD&gt;</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Range 0</td>
<td>PRAN0</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td>1=CP|LD, 0=UNC|LD</td>
</tr>
<tr>
<td></td>
<td>Range 1</td>
<td>PRAN1</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Range 2</td>
<td>PRAN2</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Range 3</td>
<td>PRAN3</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Range 4</td>
<td>PRAN4</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Range 5</td>
<td>PRAN5</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Range 6</td>
<td>PRAN6</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Range 7</td>
<td>PRAN7</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Power range auto/manual</td>
<td>PWRR&lt;PAUTO</td>
<td>PMAN&gt;</td>
<td>2</td>
<td>0,1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Source power on/off</td>
<td>SOUP&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td><strong>Test set</strong></td>
<td>attenuation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port 1</td>
<td>ATTP1[D](^1)</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0, 10, 20 \ldots 70 dB</td>
</tr>
<tr>
<td></td>
<td>Port 2</td>
<td>ATTP2[D](^1)</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0, 10, 20 \ldots 70 dB</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td>SWET[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0 to 86,400 s</td>
</tr>
<tr>
<td><strong>Measurement</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Measurement</td>
<td>Restart</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hold</td>
<td>HOLD</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Single</td>
<td>SING</td>
<td>1</td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Number of groups</td>
<td>NUMG[D]</td>
<td>3</td>
<td>D</td>
<td>OPC</td>
<td>1 to 999</td>
</tr>
<tr>
<td></td>
<td>Continuous</td>
<td>CONT</td>
<td>1</td>
<td>0,1</td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>External trigger off</td>
<td>EXTTOFF</td>
<td>1</td>
<td>0,1</td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>External trigger on sweep</td>
<td>EXTTON</td>
<td>1</td>
<td>0,1</td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>External trigger on point</td>
<td>EXTTPOIN</td>
<td>1</td>
<td>0,1</td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Manual trigger on point</td>
<td>MANTRIG</td>
<td>1</td>
<td>0,1</td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td><strong>Points</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Points</td>
<td>POIN[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>3, 11, 25, 51, 101</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>201, 401, 801, 1501</td>
</tr>
<tr>
<td><strong>Coupled channels</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Coupled channels</td>
<td>COUC&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td><strong>CW freq</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CW freq</td>
<td>CWFREQ[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range(^2)</td>
</tr>
<tr>
<td><strong>Power slope</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Power slope</td>
<td>SLOPE[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>-2 to 2 dB/GHz</td>
</tr>
<tr>
<td></td>
<td>On/off</td>
<td>SLOPO&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
</tbody>
</table>

1 Option 011 only.

2 For frequency sweeps: 30 kHz to 3 GHz. (To 6 GHz for Option 006). For power sweeps: -15 to 20 dBm in range 0, 25 dB maximum in other ranges (-100 to +100 with power meter cal on). For CW time: 0 to 24 hours. For frequency sweep, transform on: ± 1/frequency step. For CW time sweep, transform on: ± 1/time step.
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sweep type</td>
<td>Linear</td>
<td>LINFREQ</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Log</td>
<td>LOGFREQ</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>List</td>
<td>LSTFREQ</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Select a segment</td>
<td>SSEG[D]</td>
<td>3</td>
<td>0,1</td>
<td></td>
<td>1 to 30</td>
</tr>
<tr>
<td></td>
<td>Select all segments</td>
<td>ASEG</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Power</td>
<td>POWS</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CW time</td>
<td>CWTIME</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit list</td>
<td>Begin</td>
<td>EDITLIST</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add segment</td>
<td>SADD</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Edit segment N</td>
<td>SED[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>1 to 30</td>
</tr>
<tr>
<td></td>
<td>Done with segment</td>
<td>SDON</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delete segment</td>
<td>SDEL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Done</td>
<td>EDITDONE</td>
<td>1</td>
<td></td>
<td></td>
<td>OPC</td>
</tr>
<tr>
<td></td>
<td>Clear list</td>
<td>CLEL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit segment</td>
<td>Start</td>
<td>STAR[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Stop</td>
<td>STOP[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Center</td>
<td>CENT[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Span</td>
<td>SPAN[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Points</td>
<td>POIN[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>1 to 1632</td>
</tr>
<tr>
<td></td>
<td>Stepsize</td>
<td>STPSIZE[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>CW</td>
<td>CWFREQ[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td>MARKER</td>
<td>Select active</td>
<td>1 to 4</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>All off</td>
<td>MARKOFF</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marker zero</td>
<td>Zero offsets</td>
<td>MARKZERO</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delta reference</td>
<td>1 to 4</td>
<td>DELR&lt;1&gt;</td>
<td>2</td>
<td>0,1</td>
<td></td>
<td>1 to 4</td>
</tr>
<tr>
<td></td>
<td>Fixed marker</td>
<td>DELRFIXM</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mode off</td>
<td>DELO</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fixed mtr position</td>
<td>Stimulus</td>
<td>MAREFSTI[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>MAREFVAL[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Amplitude range&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Aux value</td>
<td>MAREPAUV[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Amplitude range&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

<sup>1</sup> For frequency sweeps: 30 kHz to 3 GHz. (To 6 GHz for Option 006). For power sweeps: -15 to 20 dBm in range 0, 25 dB maximum in other ranges (−100 to +100 with power meter cal on). For CW time: 0 to 24 hours. For frequency sweep, transform on: ±1/frequency step. For CW time sweep, transform on: ±1/time step.

<sup>2</sup> For log magnitude: ±500 dB. For phase: ±500 degrees. For Smith chart and Polar: ±500 units. For linear magnitude: ±500 units. For SWR: ±500 units. The scale is always positive, and has minimum values of .001 dB, 10e-12 degrees, 10e-15 seconds, and 10 picoamperes.
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marker placement</td>
<td>Continuous</td>
<td>MARKCONT</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Discrete</td>
<td>MARKDISC</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coupled</td>
<td>Couple channels</td>
<td>MARKCOUP</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Uncouple</td>
<td>MARKUNCO</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Displayed</td>
<td>On/off</td>
<td>DISM&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polar markers</td>
<td>Log</td>
<td>POLMLOG</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Linear</td>
<td>POLMLIN</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re/Im</td>
<td>POLMRI</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smith markers</td>
<td>Linear</td>
<td>SMIMLIN</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Log</td>
<td>SMIMLOG</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re/Im</td>
<td>SMIMRI</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>R-JX</td>
<td>SMIMRX</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G-JB</td>
<td>SMIMGB</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MARKER FCTN**

<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set function to marker value</td>
<td>Start</td>
<td>MARKSTAR</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stop</td>
<td>MARKSTOP</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Center</td>
<td>MARKCENT</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Span</td>
<td>MARKSPAN</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reference</td>
<td>MARKREF</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delay</td>
<td>MARKDELA</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search</td>
<td>Off</td>
<td>SEAOFF</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Maximum</td>
<td>SEAMAX</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minimum</td>
<td>SEAMIN</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Target</td>
<td>SEATARG[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Amplitude range(^1)</td>
</tr>
<tr>
<td></td>
<td>Search left</td>
<td>SEAL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Search right</td>
<td>SEAR</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td>Value</td>
<td>WIDV[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Amplitude range(^1)</td>
</tr>
<tr>
<td>Search</td>
<td>Search on/off</td>
<td>WIDT&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tracking search</td>
<td>On/off</td>
<td>TRACK&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Statistics</td>
<td>On/off</td>
<td>MEASTAT&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^1\) For log magnitude: ± 500 dB. For phase: ± 500 degrees. For Smith chart and Polar: ± 500 units. For linear magnitude: ± 500 units. For SWR: ± 500 units. The scale is always positive, and has minimum values of .001 dB, 10e-12 degrees, 10e-15 seconds, and 10 picoamps.
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SAVE/RECALL-internal registers</td>
<td></td>
<td></td>
<td></td>
<td>OPC</td>
<td></td>
</tr>
<tr>
<td>Save</td>
<td>Selected reg</td>
<td>SAVE&lt;i&gt;</td>
<td>2</td>
<td></td>
<td>OPC</td>
<td>1 to 5</td>
</tr>
<tr>
<td></td>
<td>Selected reg</td>
<td>SAVEREG&lt;i&gt;</td>
<td>2</td>
<td></td>
<td>OPC</td>
<td>01 to 31</td>
</tr>
<tr>
<td>Clear</td>
<td>Selected reg</td>
<td>CLEA&lt;i&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>1 to 5</td>
</tr>
<tr>
<td></td>
<td>Selected reg</td>
<td>CLEARREG&lt;i&gt;</td>
<td>2</td>
<td></td>
<td>OPC</td>
<td>01 to 31</td>
</tr>
<tr>
<td></td>
<td>All regs</td>
<td>CLEARALL</td>
<td>1</td>
<td></td>
<td></td>
<td>OPC</td>
</tr>
<tr>
<td>Recall</td>
<td>Selected reg</td>
<td>RECA&lt;i&gt;</td>
<td>2</td>
<td></td>
<td>OPC</td>
<td>1 to 5</td>
</tr>
<tr>
<td></td>
<td>Selected reg</td>
<td>RECARREG&lt;i&gt;</td>
<td>2</td>
<td></td>
<td>OPC</td>
<td>01 to 31</td>
</tr>
<tr>
<td>Title</td>
<td>Internal reg</td>
<td>TITR&lt;i&gt;[$]</td>
<td>4</td>
<td></td>
<td></td>
<td>1 to 5, 10 character</td>
</tr>
<tr>
<td></td>
<td>Internal reg</td>
<td>TITREG&lt;i&gt;[$]</td>
<td>4</td>
<td></td>
<td></td>
<td>01 to 31, 10 character</td>
</tr>
<tr>
<td></td>
<td>Titles save state filename (sequencing only)</td>
<td>TITF0</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Title plot file</td>
<td>TITP</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 1-2. Key Select Codes (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SAVE/RECALL-disk files</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purge</td>
<td>Selected file(^1)</td>
<td>PURG&lt;1&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>1 to 5</td>
</tr>
<tr>
<td>Store</td>
<td>To disk(^2)</td>
<td>STOR&lt;1&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>1 to 5</td>
</tr>
<tr>
<td>Title</td>
<td>Disk file</td>
<td>TITF&lt;1&gt;[8]</td>
<td>4</td>
<td></td>
<td></td>
<td>1 to 5, 10 character</td>
</tr>
<tr>
<td></td>
<td>Titles save filename (sequencing only)</td>
<td>TITF0</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Title plot file</td>
<td>TITP</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Include with disk files</td>
<td>Data</td>
<td>EXTMDATA&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Raw data</td>
<td>EXTMRAW&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Formatted data</td>
<td>EXTMFORM&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User graphics</td>
<td>EXTMGRAP&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data only</td>
<td>EXTMDATO&lt;ON/OFF&gt;</td>
<td>2</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Save format</strong></td>
<td>ASCII/citifile</td>
<td>SAVUASC</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Binary</td>
<td>SAVUBINA</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Load</strong></td>
<td>From disk(^1)</td>
<td>LOAD&lt;1&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>1 to 5</td>
</tr>
<tr>
<td></td>
<td>File titles(^1)</td>
<td>REFT</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Initialize</strong></td>
<td>Internal disk</td>
<td>INID</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>External disk</td>
<td>INIE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LIF Directory size</td>
<td>DIRS[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>8 to 8192</td>
</tr>
<tr>
<td><strong>Select storage</strong></td>
<td>Internal disk</td>
<td>INTD</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Internal memory</td>
<td>INTM</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>External disk(^1)</td>
<td>EXTD</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Disk format</strong></td>
<td>DOS</td>
<td>FORMATDOS</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LIF</td>
<td>FORMATLIF</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^1\) Requires pass-control mode when using the HP-IB port.
Table 1-2. Key Select Codes (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale</td>
<td>Auto</td>
<td>AUTO</td>
<td>1</td>
<td></td>
<td></td>
<td>Amplitude range^1</td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>SCAL[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Amplitude range^1</td>
</tr>
<tr>
<td>Reference</td>
<td>Position</td>
<td>REFP[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0&lt;D&lt;10</td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>REFV[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Amplitude range^1</td>
</tr>
<tr>
<td></td>
<td>Set to mkr</td>
<td>MARKREF</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Delay</td>
<td>Set delay</td>
<td>ELED[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>±1.0 s</td>
</tr>
<tr>
<td></td>
<td>Set to mkr</td>
<td>MARKDELA</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Phase</td>
<td>Offset</td>
<td>PHAO[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>±360 deg</td>
</tr>
</tbody>
</table>

SEQ-sequencing

| TTL I/O        | TTL out high continuously | TTLOH | 1 |     | 1 to 6 |
|                | TTL out low continuously | TTLOL | 1 |     |        |
|                | TTL low - end sweep high | TTLHPULS | 1 |     |        |
|                | TTL high - end sweep low | TLLPULS | 1 |     |        |
| Programs all GPIO output bits | PARAOUT[D] | 3 |     | 0 to 255 |
| Set specified bit on GPIO | SETBIT[D] | 2 |     | 0 to 7 |
| Clear specified bit on GPIO | CLEABIT[D] | 2 |     | 0 to 7 |
| Specify input GPIO bit for IPBI | PARAIN[D] | 2 |     | 0 to 4 |
| Input GPIO bit high - do SEQ<1> | IFBIHIGH | 1 |     |     |
| Input GPIO bit low - do SEQ<1> | IFBILOW | 1 |     |     |
| Set test set I/O lines D0 - D2 for forward measurement | TSTIOFWD[D] | 3 | D | 0 to 7 |
| Set test set I/O lines D0 - D2 for reverse measurement | TSTSTOREV[D] | 3 | D | 0 to 7 |

^1 For log magnitude: ± 500 dB. For phase: ± 500 degrees. For Smith chart and Polar: ± 500 units. For linear magnitude: ± 500 units. For SWR: ± 500 units. The scale is always positive, and has minimum values of .001 dB, 10e-12 degrees, 10e-15 seconds, and 10 picounits.
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save/recall sequences</td>
<td>Store to disk$^1$</td>
<td>STORESEQ&lt;i&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>1 to 6</td>
</tr>
<tr>
<td></td>
<td>Recall from disk$^1$</td>
<td>LOADSEQ&lt;i&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>1 to 6</td>
</tr>
<tr>
<td>Special functions</td>
<td>Peripheral address</td>
<td>ADDRPERI[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0 to 30</td>
</tr>
<tr>
<td></td>
<td>Title to peripheral</td>
<td>TITTPERI</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Wait D seconds</td>
<td>SEQWAIT[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>0.1 to 3000 s</td>
</tr>
<tr>
<td></td>
<td>Pause</td>
<td>PAUS</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Marker to CW freq.</td>
<td>MARKCW</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Emit beep</td>
<td>EMIB</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Title to printer</td>
<td>TITTPRIN</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Title to power</td>
<td>TITTPMTR</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Show menus</td>
<td>SHOM</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Assert seq. status bit</td>
<td>ASSS</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Read pwr mtr/HP-IB into title string</td>
<td>PMTRTTIT</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Send number into trace memory</td>
<td>TITTMEM</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Duplicate seq. X to seq. Y</td>
<td>DUPLSEQ&lt;X&gt;SEQ&lt;Y&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>X, Y = 1 to 6</td>
</tr>
<tr>
<td></td>
<td>Print sequence I</td>
<td>PRINSEQ&lt;i&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>1 to 6</td>
</tr>
<tr>
<td></td>
<td>Title sequence I</td>
<td>TITSEQ&lt;i&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>1 to 6</td>
</tr>
<tr>
<td></td>
<td>Clear sequence I</td>
<td>CLEARSEQ&lt;i&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>1 to 6</td>
</tr>
<tr>
<td>Decision making</td>
<td>If limit test pass then do sequence</td>
<td>IFLTFASSSEQ&lt;i&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>1 to 6</td>
</tr>
<tr>
<td></td>
<td>If limit test fail then do sequence</td>
<td>IFLTFAILSEQ&lt;i&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>1 to 6</td>
</tr>
<tr>
<td>Loop counter</td>
<td>Set value</td>
<td>LOOC[D]</td>
<td>3</td>
<td></td>
<td></td>
<td>0 to 32,760</td>
</tr>
<tr>
<td></td>
<td>Increment by 1</td>
<td>INCRLOOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Decrement by 1</td>
<td>DECRLOOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If counter equals 0 then do sequence</td>
<td>IFLEQZESEQ&lt;i&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>1 to 6</td>
</tr>
<tr>
<td></td>
<td>If counter not eq., O then do sequence</td>
<td>IFLCNEZESEQ&lt;i&gt;</td>
<td>2</td>
<td></td>
<td></td>
<td>1 to 6</td>
</tr>
</tbody>
</table>

### STIMULUS

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>CENT[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td></td>
<td>Stimulus range$^2$</td>
</tr>
<tr>
<td>Span</td>
<td>SPAN[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td></td>
<td>Stimulus range$^2$</td>
</tr>
<tr>
<td>Start</td>
<td>STAR[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td></td>
<td>Stimulus range$^2$</td>
</tr>
<tr>
<td>Stop</td>
<td>STOP[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td></td>
<td>Stimulus range$^2$</td>
</tr>
</tbody>
</table>

1 Requires pass control when using the HP-IB port.
2 For frequency sweeps: 30 kHz to 3 GHz. (To 6 GHz for Option 006). For power sweeps: -15 to 20 dBm in range 0, 25 dB maximum in other ranges (-100 to +10 with power meter cal on). For CW time: 0 to 24 hours. For frequency sweep, transform on: ± 1/frequency step. For CW time sweep, transform on: ± 1/time step.
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set clock</td>
<td>Time stamp</td>
<td>TIMESAM&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
</tr>
<tr>
<td></td>
<td>Set date</td>
<td>SETDATE[S]</td>
<td>4</td>
<td></td>
<td>HH:MM:SS</td>
</tr>
<tr>
<td></td>
<td>Set time</td>
<td>SETTIME[S]</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Harmonic mode</td>
<td>Off</td>
<td>HARMOFF</td>
<td>1</td>
<td>0,1</td>
<td>OPC</td>
</tr>
<tr>
<td></td>
<td>Second</td>
<td>HARMSEC</td>
<td>1</td>
<td>0,1</td>
<td>OPC</td>
</tr>
<tr>
<td></td>
<td>Third</td>
<td>HARMTHIR</td>
<td>1</td>
<td>0,1</td>
<td>OPC</td>
</tr>
<tr>
<td>Instrument mode</td>
<td>Network analyzer</td>
<td>INSMNETA</td>
<td>1</td>
<td>0,1</td>
<td>OPC</td>
</tr>
<tr>
<td></td>
<td>Ext. source auto</td>
<td>INSMEXSA</td>
<td>1</td>
<td>0,1</td>
<td>OPC</td>
</tr>
<tr>
<td></td>
<td>Ext. source manual</td>
<td>INSMEXSM</td>
<td>1</td>
<td>0,1</td>
<td>OPC</td>
</tr>
<tr>
<td></td>
<td>Tuned receiver</td>
<td>INSMTUNR</td>
<td>1</td>
<td>0,1</td>
<td>OPC</td>
</tr>
<tr>
<td>Service</td>
<td>Analog bus</td>
<td>ANAB&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
</tr>
<tr>
<td>Frequency offset</td>
<td>On/off</td>
<td>FREQOFFS&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>YOFF[D]</td>
<td>3</td>
<td>D</td>
<td>0 to 3 GHz (0 to 6 GHz Option 006)</td>
</tr>
<tr>
<td></td>
<td>Set RF &gt; LO</td>
<td>RFSTLO</td>
<td>1</td>
<td>0,1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Set RF &lt; LO</td>
<td>RFCITLO</td>
<td>1</td>
<td>0,1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Select up converter</td>
<td>UCONV</td>
<td>1</td>
<td>0,1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Select down converter</td>
<td>DCONV</td>
<td>1</td>
<td>0,1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>View measurement/mixer setup</td>
<td>VIEM&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
</tr>
</tbody>
</table>
## Table 1-2. Key Select Codes (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SYSTEM-limit testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Limit line</td>
<td>On/off</td>
<td>LIMILINE&lt;ON/OFF&gt;</td>
<td>2</td>
<td></td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Limit test</td>
<td>On/off</td>
<td>LIMITEST&lt;ON/OFF&gt;</td>
<td>2</td>
<td></td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Beeper</td>
<td>BEEPFAIL&lt;ON/OFF&gt;</td>
<td>2</td>
<td></td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Limit offset</td>
<td>Stimulus</td>
<td>LIMISTIO[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td>Limit offset</td>
<td>Amplitude</td>
<td>LIMIAMPO[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Amplitude range&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td>Limit offset</td>
<td>Marker to offset</td>
<td>LIMIMAOF</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit table</td>
<td>Begin edit</td>
<td>EDITLIML</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add segment</td>
<td>SADD</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Edit segment D</td>
<td>SEDI[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>1 to 18</td>
</tr>
<tr>
<td></td>
<td>Segment done</td>
<td>SDON</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delete segment</td>
<td>SDEL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Done with edit</td>
<td>EDITDONE</td>
<td>1</td>
<td></td>
<td></td>
<td>OPC</td>
</tr>
<tr>
<td></td>
<td>Clear list</td>
<td>CLEAL</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit segment</td>
<td>Stimulus value</td>
<td>LIMIS[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Marker to stimulus</td>
<td>MARKSTIM</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Upper limit</td>
<td>LIMU[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Amplitude range&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Lower limit</td>
<td>LIML[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Amplitude range&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Delta limits</td>
<td>LIMD[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Amplitude range&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Middle value</td>
<td>LIMM[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Amplitude range&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>Marker to middle</td>
<td>MARKMIDD</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Flat line type</td>
<td>LIMTFL</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sloping line type</td>
<td>LIMTSL</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Single point type</td>
<td>LIMTSP</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<sup>1</sup> For frequency sweeps: 30 kHz to 3 GHz. (To 6 GHz for option 006). For power sweeps: -15 to 20 dBm in range 0, 25 dB maximum in other ranges (-100 to +100 with power meter cal on). For CW time: 0 to 24 hours. For frequency sweep, transform on: ± 1/frequency step. For CW time sweep, transform on: ±1/time step.

<sup>2</sup> For log magnitude: ± 500 dB. For phase: ± 500 degrees. For Smith chart and Polar: ± 500 units. For linear magnitude: ± 500 units. For SWR: ± 500 units. The scale is always positive, and has minimum values of .001 dB, 10e-12 degrees, 10e-15 seconds, and 10 picounits.
<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Mnemonic</th>
<th>S</th>
<th>?</th>
<th>O</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transform</td>
<td>On/off</td>
<td>TIMDTRAN&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td>Set freq</td>
<td>Low pass</td>
<td>SETF</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>Low pass impulse</td>
<td>LOWPIMPU</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Low pass step</td>
<td>LOWPSTEP</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bandpass</td>
<td>BANDPASS</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Specify gate menu</td>
<td>SPEG</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Window</td>
<td>Maximum</td>
<td>WINDMAXI</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>WINDNORM</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minimum</td>
<td>WINDMINI</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Any value</td>
<td>WINDOW[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>State dependent</td>
</tr>
<tr>
<td>Window shape</td>
<td>Use trace memory</td>
<td>WINDUSEM&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td>Demodulation</td>
<td>Off</td>
<td>DEMOOFF</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Amplitude</td>
<td>DEMOAMPL</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phase</td>
<td>DEMOPHAS</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gate</td>
<td>On/off</td>
<td>GATEO&lt;ON</td>
<td>OFF&gt;</td>
<td>2</td>
<td>1,0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Start</td>
<td>GATESTAR[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range¹</td>
</tr>
<tr>
<td></td>
<td>Stop</td>
<td>GATESTOP[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range¹</td>
</tr>
<tr>
<td></td>
<td>Center</td>
<td>GATCECNET[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range¹</td>
</tr>
<tr>
<td></td>
<td>Span</td>
<td>GATESPAN[D]</td>
<td>3</td>
<td>D</td>
<td></td>
<td>Stimulus range¹</td>
</tr>
<tr>
<td>Gate shape</td>
<td>Maximum</td>
<td>GATSMAXI</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Wide</td>
<td>GATSWIDE</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>GATSNORM</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minimum</td>
<td>GATSMINI</td>
<td>1</td>
<td>0,1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

¹ For frequency sweeps: 30 kHz to 3 GHz. (76 GHz for Option 006). For power sweeps: -15 to 20 dBm in range 0, 25 dB maximum in other ranges (-100 to -100 with power meter cal on). For CW time: 0 to 24 hours. For frequency sweep, transform on: ± 1/frequency step. For CW time sweep, transform on: ± 1/time step.
### Table 1-3. HP-IB Only Commands

<table>
<thead>
<tr>
<th>Action</th>
<th>Mnemonic</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity</td>
<td>IDN?</td>
<td>1</td>
<td>Outputs the identification string: &quot;HEWLETT PACKARD, 8753D,0.X.XX&quot;, where X.XX is the firmware revision of the instrument.</td>
</tr>
<tr>
<td>Key</td>
<td>KEY[D]</td>
<td>3</td>
<td>Imitates pressing a key. The data transmitted is the key code, as defined in Figure 1-2.</td>
</tr>
<tr>
<td>Key code</td>
<td>KOR?</td>
<td>1</td>
<td>Outputs last key code or knob count. If the reply is positive, it is a key code. If it is negative, then set bit 15 equal to bit 14, and the resulting two byte integer is the RPG knob count. It can be either positive or negative. There are about 120 counts per turn.</td>
</tr>
<tr>
<td>Move marker</td>
<td>MARKBUCK[D]</td>
<td>3</td>
<td>Moves the marker to the selected point on the trace. On a 201 point sweep, D can range from 0 to 200.</td>
</tr>
<tr>
<td>On completion</td>
<td>OPC</td>
<td>1</td>
<td>Reports completion of the last OPC-compatible command received since OPC; or OPC?; was received.</td>
</tr>
<tr>
<td>Plot/print</td>
<td>PSOFT&lt;ON</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>softkeys</td>
<td>OFF&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy default</td>
<td>DEFLTCP</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Revision</td>
<td>SOFR</td>
<td>1</td>
<td>Displays the software revision on the analyzer.</td>
</tr>
<tr>
<td>Learn string</td>
<td>SELL[D]</td>
<td>2</td>
<td>Selects the learn string revision to input to and output from the analyzer. The valid parameters are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0: Defaults to current revision.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>201: Revision 8753B 2.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>300: Revision 8753B 3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>401: Revision 8753C 4.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>402: Revision 8753C 4.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>412: Revision 8753C 4.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>413: Revision 8753C 4.13</td>
</tr>
<tr>
<td>Sampler</td>
<td>SAMC&lt;ON</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>OFF&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test set</td>
<td>TESS?</td>
<td>1</td>
<td>Returns a one on the standard analyzer. (The standard analyzer does not recognize external test sets.) For option 011: returns a one or a two if an HP 85046A/B S-parameter test set is present; returns a four or a five if an HP 85047A S-parameter test set is present; returns a three if an HP 85046 E20 mixer test set is present.</td>
</tr>
<tr>
<td>Delay</td>
<td>WAIT</td>
<td>3</td>
<td>Used with OPC to ensure completion of non-OPCable commands. The command que must be empty to continue.</td>
</tr>
<tr>
<td>Action</td>
<td>Mnemonic</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------</td>
<td>--------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Copy titles</td>
<td>COPYFRFT</td>
<td>1</td>
<td>Copies file titles defined by TITP&lt;i&gt; into register titles.</td>
</tr>
<tr>
<td>Copy titles</td>
<td>COPYFRFT</td>
<td>1</td>
<td>Copies register titles defined by TITR&lt;i&gt; into file titles.</td>
</tr>
<tr>
<td>Preset</td>
<td>PRES</td>
<td>1</td>
<td>Presets instrument.</td>
</tr>
<tr>
<td>External trigger</td>
<td>EXTHIGH</td>
<td>1</td>
<td>Sets the trigger polarity high.</td>
</tr>
<tr>
<td>External trigger</td>
<td>EXTTLLOW</td>
<td>1</td>
<td>Sets the trigger polarity low.</td>
</tr>
<tr>
<td><strong>INPUT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>INFUDATA[D]</td>
<td>3</td>
<td>Accepts error-corrected data.</td>
</tr>
<tr>
<td>Formatted</td>
<td>INPUFORM[D]</td>
<td>3</td>
<td>Accepts formatted data.</td>
</tr>
<tr>
<td>Uncorrected</td>
<td>INPURAW1[D]</td>
<td>3</td>
<td>Accepts raw data.</td>
</tr>
<tr>
<td></td>
<td>INPURAW2[D]</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INPURAW3[D]</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INPURAW4[D]</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Error coefficient</td>
<td>INPUCALC&lt;01,02 ... 12&gt;[D]</td>
<td>3</td>
<td>Accepts the individual error coefficient arrays. Issue the command that begins the calibration the coefficients are from (e.g. CALIS11); then input the data. Finally, issue SAVC; and trigger a sweep.</td>
</tr>
<tr>
<td>Power meter cal.</td>
<td>INPUMCAL&lt;1,2&gt;[D]</td>
<td>3</td>
<td>Accepts power meter cal array. Values should be entered as 100 times the desired source power in dB.</td>
</tr>
<tr>
<td>Cal kit</td>
<td>INPUCALK[D]</td>
<td>3</td>
<td>Accepts a cal kit.</td>
</tr>
<tr>
<td>Learn string</td>
<td>INPULEAS[D]</td>
<td>3</td>
<td>Accepts the learn string. Preceded by SELL if learn string is not current revision.</td>
</tr>
<tr>
<td><strong>MENUS</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Averaging</td>
<td>MENUAVG</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Calibration</td>
<td>MENUCAL</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Copy</td>
<td>MENUCOPY</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Display</td>
<td>MNU_DISP</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Format</td>
<td>MNU_FORM</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Marker</td>
<td>MNU_MARK</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Meas</td>
<td>MNU_MEAS</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Marker function</td>
<td>MNU_MRKF</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Off</td>
<td>MENU&lt;ON/OFF&gt;</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Save Recall</td>
<td>MNU_RECA</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Scale</td>
<td>MNU_SCL</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Sequence</td>
<td>MNUSEQU</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Stimulus</td>
<td>MNU_STIM</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>System</td>
<td>MNU_SYS</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Mnemonic</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Active function</td>
<td>OUTFACT1</td>
<td>1</td>
<td>Outputs value of function in active entry area in ASCII format.</td>
</tr>
<tr>
<td>Error coefficient</td>
<td>OUTPCALC&lt;01,02 ... 12&gt;</td>
<td>2</td>
<td>Outputs the selected error coefficient array from the active channel. Each array is the same as a data array. See Table 1-5, for the contents of the arrays.</td>
</tr>
<tr>
<td>Interp. cal.</td>
<td>OUTPICAL&lt;01,02 ... 12&gt;</td>
<td>2</td>
<td>Outputs the selected interpolated cal coefficient array.</td>
</tr>
<tr>
<td>Cal kit</td>
<td>OUTPCALK</td>
<td>1</td>
<td>Outputs the active cal kit, a less than 1000 byte string in form 1.</td>
</tr>
<tr>
<td>Data</td>
<td>OUTPDATA</td>
<td>1</td>
<td>Outputs the error corrected data from the active channel in real/imaginary pairs. See Figure 1-1.</td>
</tr>
<tr>
<td>Error</td>
<td>OUTPERRO</td>
<td>1</td>
<td>Outputs the oldest error in the error queue. The error number is transmitted, then the error message, in ASCII format.</td>
</tr>
<tr>
<td>Formatted</td>
<td>OUTPFORM</td>
<td>1</td>
<td>Outputs the formatted trace data from the active channel in current display units. See Table 1-6 for data transmitted.</td>
</tr>
<tr>
<td>Power meter cal. 1</td>
<td>OUTPIMCAL&lt;1,2&gt;</td>
<td>2</td>
<td>Outputs the interpolated power meter cal array for channel 1 or channel 2.</td>
</tr>
<tr>
<td>Power meter cal. 2</td>
<td>OUTPPIMCAL&lt;1,2&gt;</td>
<td>2</td>
<td>Outputs power meter cal array for channel 1 or channel 2. Values are sent as 100 times the source power in dB.</td>
</tr>
<tr>
<td>Identity</td>
<td>OUTPIDEN</td>
<td>1</td>
<td>Outputs identification string, same as IDN?.</td>
</tr>
<tr>
<td>Key code</td>
<td>OUTPK</td>
<td>1</td>
<td>Outputs the code of the last key pressed, in ASCII format. See Figure 1-2 for key codes. A -1 is transmitted for a knob turn.</td>
</tr>
<tr>
<td>Learn string</td>
<td>OUTPLEAS</td>
<td>1</td>
<td>Outputs the learn string, a less than 3,000 byte string in form 1.</td>
</tr>
<tr>
<td>External source</td>
<td>OUTPRFR</td>
<td>1</td>
<td>Outputs external source RF frequency when in external source instrument mode.</td>
</tr>
<tr>
<td>Action</td>
<td>Mnemonic</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Sequencing</td>
<td>OUTFSEQ&lt;1&gt;</td>
<td>2</td>
<td>Outputs sequence I (I = 1 to 6) listing over HP-IB.</td>
</tr>
<tr>
<td>Limit failures</td>
<td>OUTPLIMF</td>
<td>1</td>
<td>Outputs the limit results as described under OUTPLIML for only those stimulus points that failed.</td>
</tr>
<tr>
<td>Limit list</td>
<td>OUTPLIML</td>
<td>1</td>
<td>Outputs the limit test results for each stimulus point. The results consist of four numbers. The first is the stimulus value tested, the second is the test result: -1 for no test, 0 for fail, 1 for pass. The third number is the upper limit value, and the fourth is the lower limit value. This is a form 4 transfer.</td>
</tr>
<tr>
<td>Limit marker</td>
<td>OUTPLIMM</td>
<td>1</td>
<td>Outputs the limit test results as described for OUTPLIML at the marker.</td>
</tr>
<tr>
<td>Marker</td>
<td>OUTPMARK</td>
<td>1</td>
<td>Outputs the active marker values in 3 numbers. The first two numbers are the marker values, and the last is the stimulus value. See Table 1-6 for the marker values.</td>
</tr>
<tr>
<td>Memory</td>
<td>OUTPMEMO</td>
<td>1</td>
<td>Outputs the memory trace from the active channel. It is error corrected data in real/imaginary pairs, and can be treated the same as data from OUTPDATA.</td>
</tr>
<tr>
<td>Marker statistics</td>
<td>OUTPMSIA</td>
<td>1</td>
<td>Outputs marker statistics: mean, standard deviation, and peak to peak deviation. ASCII format.</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>OUTPMWID</td>
<td>1</td>
<td>Outputs results of bandwidth search: bandwidth, center, and Q. ASCII format.</td>
</tr>
<tr>
<td>Bandwidth and Loss</td>
<td>OUTPMWIL</td>
<td></td>
<td>Outputs results of bandwidth search: bandwidth, center, loss, and Q.</td>
</tr>
<tr>
<td>Clock</td>
<td>READDATE</td>
<td>1</td>
<td>Outputs the date of the clock in the following format: DD MMM YYYY</td>
</tr>
<tr>
<td>Clock</td>
<td>READTIME</td>
<td>1</td>
<td>Outputs the time of the clock in the following format: HH:MM:SS</td>
</tr>
<tr>
<td>Action</td>
<td>Mnemonic</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OUTPUT (Continued)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print list values</td>
<td>OUTPPRN</td>
<td>1</td>
<td>Prints screen to computer.</td>
</tr>
<tr>
<td></td>
<td>OUTPPRNALL</td>
<td></td>
<td>Prints all list values or operating and marker parameters in text mode to HP-IB. Requires you to have previously selected the function to output. Use LSV for values, OPEP for operating parameters.</td>
</tr>
<tr>
<td>Raw data</td>
<td>OUTPRAW1</td>
<td>1</td>
<td>Outputs uncorrected data arrays for the active channel. Raw 1 holds the data unless a 2-port calibration is on, in which case the arrays hold S11, S21, S12, and S22, respectively. The data is in real/imaginary pairs.</td>
</tr>
<tr>
<td></td>
<td>OUTPRAW2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OUTPRAW3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OUTPRAW4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status byte</td>
<td>OUTPSTAT</td>
<td>1</td>
<td>Outputs the status byte. ASCII format.</td>
</tr>
<tr>
<td>Display title</td>
<td>OUTPTITL</td>
<td>1</td>
<td>Outputs the display title. ASCII format.</td>
</tr>
</tbody>
</table>

**OUTPUT FORMATS**

| FORM1 | 1 HP 8753 internal format, with header. |
| FORM2 | 1 32 bit floating point, with header.  |
| FORM3 | 1 64 bit floating point, with header.  |
| FORM4 | 1 ASCII format. No header.             |
| FORM5 | 1 32 bit PC format (bytes reversed).   |

**SOFTWARE**

| Press | SOFT{1} | 2 Activates softkey 1, 1=1 to 8. |
| Label | WRSK<1 TO 8>[8] | 4 Writes label (10 char) to indicated softkey. |

**STATUS REPORTING**

| Clear  | CLES    | 1 Clears the status byte. |
| Interrogate  | ESB? | 1 Returns event-status register B. |
|           | ESR?  | 1 Returns the event-status register. |
|           | OUTPSTAT | 1 Returns the status byte. |
| Enable   | ESE[D] | 3 Enables event-status register: (0<D<255) |
|          | ESNB[D] | 3 Enables event-status register B: (0<D<255) |
|          | SRE[D] | 3 Enables SRQ: (0<D<255) |
Calibration

Measurement calibration over HP-IB follows the same command sequence as a calibration from the front-panel:

1. Start by selecting a calibration kit, such as 50 ohm type N (CALKN50; over HP-IB.)
2. Select a calibration type, such as S11 1-port (CALIS111; over HP-IB.)
3. Call each class used by the calibration type, such as FORWARD; OPEN (CLASS11A; over HP-IB.) During a 2-port calibration, the reflection, transmission, and isolation subsequences must be opened before the classes in the subsequence are called, and then closed at the end of each subsequence.
4. If a class has more than one standard in it, select a standard from the menu presented (STANA to STANG over HP-IB.)
5. If, during a calibration, two standards are measured to satisfy one class, the class must be closed with DONE;
6. Declare the calibration done, such as with DONE 1-PORT CAL (SAV1; over HP-IB.)

The STANA to STANG commands are all held commands because they trigger a sweep. If a class has only one standard in it, which means that it will trigger a sweep when called, the class command will be held also.

Note

Since different cal kits can have a different number of standards in a given class, any automated calibration sequence is valid only for a specific cal kit.

<table>
<thead>
<tr>
<th>Table 1-4. Relationship Between Calibrations and Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
</tr>
<tr>
<td>Reflection; 1</td>
</tr>
<tr>
<td>S11A, RE FW MTC</td>
</tr>
<tr>
<td>S11B, LN FW MTC</td>
</tr>
<tr>
<td>S11C, LN FW TRAN</td>
</tr>
<tr>
<td>S22A, LN RV MTCH</td>
</tr>
<tr>
<td>S22B, LN RV TRAN</td>
</tr>
<tr>
<td>S22C, LN RV TRAN</td>
</tr>
<tr>
<td>Transmission; 1</td>
</tr>
<tr>
<td>Forward match</td>
</tr>
<tr>
<td>Forward trans</td>
</tr>
<tr>
<td>Reverse match</td>
</tr>
<tr>
<td>Reverse trans</td>
</tr>
<tr>
<td>Isolation; 1</td>
</tr>
<tr>
<td>Forward</td>
</tr>
<tr>
<td>Reverse</td>
</tr>
<tr>
<td>Response</td>
</tr>
<tr>
<td>Response and isolation</td>
</tr>
</tbody>
</table>

1 These subheadings must be called when doing 2-port calibrations.
<table>
<thead>
<tr>
<th>Array</th>
<th>Response</th>
<th>Response and Isolation</th>
<th>1-port</th>
<th>2-port</th>
<th>TRL*/LRM*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ER or ET</td>
<td>EX (ED)²</td>
<td>ED</td>
<td>EDF</td>
<td>ED F</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>ES</td>
<td>EDF</td>
<td>ES F</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>E (ER)</td>
<td>ER</td>
<td>ERF</td>
<td>ER F</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>EXF</td>
<td>EXF</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>ELF</td>
<td>LY</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>ETR</td>
<td>TPR</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>ES R</td>
<td>ESR</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>ER R</td>
<td>ESR</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td>EX R</td>
<td>EX R</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>EL R</td>
<td>LR</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td>E TR</td>
<td>E TR</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 One path, 2-port cal duplicates arrays 1 to 6 in arrays 7 to 12.
2 Response and isolation corrects for crosstalk and transmission tracking in transmission measurements, and for directivity and reflection tracking in reflection measurements.

**Meaning of first subscript:**

D = directivity  
S = source match  
R = reflection tracking  
X = crosstalk  
L = load match  
T = transmission tracking

**Meaning of second subscript:**

F = forward  
R = reverse
Display Graphics

User Graphics Units

Size of Graticule only
- length = 350 to 4915
- height = 150 to 3950

Size of Complete Display (graticule plus annotation and softkey labels)
- length = 0 to 5850
- height = 0 to 4095

HP-GL subset:

Command | Description
---|---
AF; | Erases the user graphics display.
CS; | Turns off the measurement display.
DIRun,rise; | Specifies the direction in which characters are lettered.
run,rise:
1,0 = 0 degrees
0,1 = 90 degrees
-1,0 = 180 degrees
0,-1 = 270 degrees

DF; | Sets the default values.
LB[text][etx]; | Labels the display, placing the symbols starting at the current pen position. All incoming characters are printed until the etx symbol is received. The default etx symbol is the ASCII value 3 (not the character 3).
L?la; | Specifies line type:

<table>
<thead>
<tr>
<th>a</th>
<th>line</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>solid</td>
</tr>
<tr>
<td>1</td>
<td>solid</td>
</tr>
<tr>
<td>2</td>
<td>short dashes</td>
</tr>
<tr>
<td>3</td>
<td>long dashes</td>
</tr>
</tbody>
</table>

OP; | Outputs P1 and P2, the scaling limits: 0,0,5850,4095.
PAX,y; | Draws from the current pen position to x,y. There can be many pairs of x,y coordinates within one command. They are separated by commas, and the entire sequence is terminated with a semicolon.
PD; | Pen down. A line is drawn only if the pen is down.
PG;
   Erases the user graphics display.
PRx,y;
   Plot relative: draws a line from the current pen position to a position y
   up and x over.
PU;
   Pen up. Stops anything from being drawn.
RS;
   Turns ON the measurement display.
SIh,w;
   Sets the character size, for height h and width w in centimeters:

<table>
<thead>
<tr>
<th>h</th>
<th>w</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.16</td>
<td>0.20</td>
<td>smallest</td>
</tr>
<tr>
<td>0.25</td>
<td>0.30</td>
<td></td>
</tr>
<tr>
<td>0.33</td>
<td>0.39</td>
<td></td>
</tr>
<tr>
<td>0.41</td>
<td>0.49</td>
<td>largest</td>
</tr>
</tbody>
</table>

SPn;
   Selects pen n:

<table>
<thead>
<tr>
<th>n</th>
<th>brightness</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>blank</td>
</tr>
<tr>
<td>1</td>
<td>brightest</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>dimmest</td>
</tr>
</tbody>
</table>

Accepted but ignored HP-GL commands:

IM  Input service request mask
IP  Input Pl,P2 scaling points
IW  Input window
OC  Output current pen position
OE  Output error
OI  Output identity
OS  Output status
SL  Character slant
SR  Relative character size
Figure 1-1. Data Processing Chain
<table>
<thead>
<tr>
<th>Display Format</th>
<th>Marker Mode</th>
<th>OUTPMARK</th>
<th>OUTPFORM</th>
<th>MARKER READOUT*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>value 1</td>
<td>value 2</td>
<td>value 1</td>
</tr>
<tr>
<td>LOG MAG</td>
<td>dB</td>
<td></td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>PHASE</td>
<td>degrees</td>
<td></td>
<td>degrees</td>
<td></td>
</tr>
<tr>
<td>DELAY</td>
<td>seconds</td>
<td></td>
<td>seconds</td>
<td></td>
</tr>
<tr>
<td>SMITH CHART</td>
<td>LIN MKR</td>
<td>lin mag</td>
<td>degrees</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>LOG MKR</td>
<td>dB</td>
<td>degrees</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>Re/Im</td>
<td>real</td>
<td>imag</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>R + jX†</td>
<td>real ohms</td>
<td>imag ohms</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>G + jB</td>
<td>real</td>
<td>imag</td>
<td>real</td>
</tr>
<tr>
<td>POLAR</td>
<td>LIN MKR</td>
<td>lin mag</td>
<td>degrees</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>LOG MKR</td>
<td>dB</td>
<td>degrees</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>Re/Im</td>
<td>real</td>
<td>imag</td>
<td>real</td>
</tr>
<tr>
<td>LIN MAG</td>
<td>lin mag</td>
<td></td>
<td>lin mag</td>
<td></td>
</tr>
<tr>
<td>REAL</td>
<td>real</td>
<td></td>
<td>real</td>
<td></td>
</tr>
<tr>
<td>SWR</td>
<td>SWR</td>
<td></td>
<td>SWR</td>
<td></td>
</tr>
</tbody>
</table>

* The marker readout values are the marker values displayed in the upper right-hand corner of the display. They also correspond to the value and auxiliary value associated with the fixed marker.

† Value 2 is not significant in this format, though it is included in data transfers.

‡ To read the OUTPFORM for R + jX data, use the following key presses:
[FORMAT][POLAR][MEAS][MORE][CONVERSION][2 REF], or the following commands over HP-IB:
POLAJ CONVZREF).
Disk File Names

Disk files created by the analyzer consist of a state name of up to eight characters, such as FILTER, appended with up to two characters. In LIF format, the file name is FILTERXX. In DOS format, the filename is FILTER.XX. The first appended character is the file type, telling the kind of information in the file. The second appended character is a data index, used to distinguish files of the same type. Data and calibration files are form 3 data (without a header) which can be read off the disk. The other files are not meant to be decoded. Table 1-7 lists the appended characters and their meanings.

<table>
<thead>
<tr>
<th>Char 1</th>
<th>Meaning</th>
<th>Char 2</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Instrument state</td>
<td>1</td>
<td>Display graphics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>Graphics index</td>
</tr>
<tr>
<td>G</td>
<td>Graphics</td>
<td></td>
<td>Channel 1</td>
</tr>
<tr>
<td>D</td>
<td>Error corrected data</td>
<td>2</td>
<td>Channel 2</td>
</tr>
<tr>
<td>R</td>
<td>Raw data</td>
<td>1 to 4</td>
<td>Channel 1, raw arrays 1 to 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 to 8</td>
<td>Channel 2, raw arrays 1 to 4</td>
</tr>
<tr>
<td>F</td>
<td>Formatted data</td>
<td>1</td>
<td>Channel 1</td>
</tr>
<tr>
<td>M</td>
<td>Memory trace</td>
<td>2</td>
<td>Channel 2</td>
</tr>
<tr>
<td>P</td>
<td>Instrument state appendix</td>
<td>K</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Cal kit</td>
<td></td>
<td>Stimulus state</td>
</tr>
<tr>
<td>1</td>
<td>Cal data, channel 1</td>
<td>0</td>
<td>Coefficients 1 to 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 to 9</td>
<td>Coefficient 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A</td>
<td>Coefficient 11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>Coefficient 12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Cal data, channel 2</td>
<td>0 to C</td>
<td>Same as channel 1</td>
</tr>
</tbody>
</table>
Figure 1-2. Key Codes

Note 1: Key code 63 is invalid key.

Note 2: OUTPKEY; reports a knob turn as a $-1$.

Note 3: If the two byte integer sent back from KOR? is negative, it is a knob count. Decode the knob count and direction by setting bit 15 equal to bit 14. A positive result indicates counter-clockwise rotation.
### Alphabetical Code Listing

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>Measure and display A/B on the active channel.</td>
</tr>
<tr>
<td>ADDRCONT[D]</td>
<td>Controller HP-IB address: the address where control is returned after a pass control.</td>
</tr>
<tr>
<td>ADDRDISC[D]</td>
<td>Disk HP-IB address.</td>
</tr>
<tr>
<td>ADDRPERI[D]</td>
<td>Peripheral HP-IB address.</td>
</tr>
<tr>
<td>ADDRPLOT[D]</td>
<td>Plotter HP-IB address.</td>
</tr>
<tr>
<td>ADDRPWOM[D]</td>
<td>Power meter HP-IB address.</td>
</tr>
<tr>
<td>ADDRPRIN[D]</td>
<td>Printer HP-IB address.</td>
</tr>
<tr>
<td>ALTAB</td>
<td>Places the analyzer in the alternate inputs measurement mode, where inputs A and B are measured on alternate sweeps. As opposed to CHOPAB;:</td>
</tr>
<tr>
<td>ANAB&lt;ON</td>
<td>OFF&gt;</td>
</tr>
<tr>
<td>ANAI</td>
<td>Measure and display the data at the auxiliary input (ANALOG IN).</td>
</tr>
<tr>
<td>AR</td>
<td>Measure and display A/R on the active channel.</td>
</tr>
<tr>
<td>ASEG</td>
<td>Use all segments for list frequency sweep.</td>
</tr>
<tr>
<td>ASSS</td>
<td>Assert the sequence status bit.</td>
</tr>
<tr>
<td>ATTP1</td>
<td>Selects the amount of attenuation at PORT 1 (option 011 with test set only).</td>
</tr>
<tr>
<td>ATTP2</td>
<td>Selects the amount of attenuation at PORT 2 (option 011 with test set only).</td>
</tr>
<tr>
<td>AUTO</td>
<td>Auto scale the active channel.</td>
</tr>
<tr>
<td>AVERFACT[D]</td>
<td>Set the averaging factor on the active channel.</td>
</tr>
<tr>
<td>AVERO&lt;ON</td>
<td>OFF&gt;</td>
</tr>
<tr>
<td>AVERREST</td>
<td>Restart the averaging on the active channel.</td>
</tr>
<tr>
<td>BACI[D]</td>
<td>Sets the background intensity of the display.</td>
</tr>
<tr>
<td>BANDPASS</td>
<td>Select the time domain bandpass mode.</td>
</tr>
</tbody>
</table>

These commands control the warning beeper, causing it to sound if the indicated condition occurs:

| BEEPDONE<ON|OFF>   | The completion of functions such as save, done with calibration standard, and data trace saved. |
| BEEPFAIL<ON|OFF>   | A limit test failure.                                                       |
| BEEPWARN<ON|OFF>  | The generation of a warning message.                                       |
| BR         | Measure and display B/R on the active channel.                             |
These commands set the open capacitance values of an open circuit while it is being defined as a calibration standard.

C0[D]  
C1[D]  
C2[D]  
C3[D]  

CAL1  Accepted for compatibility with the HP 8510A, where its function is to begin a calibration sequence.

These commands set the power meter calibration factor corrections for the particular sensor used. Sensor B is only valid for the HP 438A which has two input channels:

CALFCALF[D]  Set the calibration factor.
CALFFREQ[D]  Select the frequency for the calibration factor correction.
CALFSEN A  Edit the sensor A calibration factor table.
CALFSEN B  Edit the sensor B calibration factor table.

These commands begin a calibration sequence:

CALIFUL2  Short, load, open, thru (SLOT) 2-port.
CALIONE2  One-path 2-port.
CALIRAI  Response and isolation.
CALIRESP  Response.
CALIS111  S11 1-port.
CALIS221  S22 1-port.
CALITRL2  Thru, reflect, line or Line, reflect, match (TRL*/LRM*) 2-port.

These commands select a default calibration kit:

CALK35MD  3.5 mm (HP 85033D).
CALK35MM  3.5 mm (HP 85033C).
CALK7MM  7 mm.
CALKN50  Type-N 50 ohm.
CALKN75  Type-N 75 ohm.
CALKUSED  User-defined calibration kit.

CALN  Calibration: none. Turns calibration type to off.
CBRI[D]  Adjusts the color brightness of the selected display feature.
CENT[D]  Sets the center stimulus value. If a list frequency segment is being edited, sets the center of the list segment.
CHAN1  Make channel 1 the active channel. OPC-compatible.
CHAN2  Make channel 2 the active channel. OPC-compatible.
CHOPAB

Places the analyzer in the chop measurement mode. As opposed to ALTAB;

CLAD

Class done, modify cal kit, specify class.

These commands call reflection standard classes during a calibration sequence. If only one standard is in the class, it is measured. If there is more than one, the standard being used must be selected with STAN<A|B|C|D|E|FIG>. If there is only one standard in the class, these commands are OPC-compatible.

CLASS11A  S11A: S11 1-port, opens.
CLASS11B  S11B: S11 1-port, shorts.
CLASS11C  S11C: S11 1-port, loads.
CLASS22A  S22A: S22 1-port, opens.
CLASS22B  S22B: S22 1-port, shorts.
CLASS22C  S22C: S22 1-port, loads.

These commands clear the indicated save/recall registers:

CLEA1  Clears save/recall register 1.
CLEA2  Clears save/recall register 2.
CLEA3  Clears save/recall register 3.
CLEA4  Clears save/recall register 4.
CLEA5  Clears save/recall register 5.
CLEARALL  Clears all the save/recall registers. OPC-compatible.
CLEAREG[I]  Clears save/recall registers 01 through 31. CLEAREG01 through CLEAREG05 are the same as CLEA1 through CLEA5.

CLEABIT [D]  Clears the specified bit on the GPIO.
CLEAL  Clears the limit line list. Should be preceded by EDITLIML.
CLEL  Clear the desired list. This could be a frequency list, power loss list, cal sensor list, or limit test list. Should be preceded by EDITLIML, EDITLIST, POWLLIST, CALFSENA, or CALFSENB.

These commands clear the sequence from the internal registers:

CLESEQ1  Sequence 1.
CLESEQ2  Sequence 2.
CLESEQ3  Sequence 3.
CLESEQ4  Sequence 4.
CLESEQ5  Sequence 5.
CLESEQ6  Sequence 6.

CLES  Clears the status register, the event-status registers, and the enable registers.

CLS  Same as CLES.
COAX

Selects coaxial offsets instead of waveguide while defining a standard during a cal kit modification.

These commands select the indicated display feature for color modification:

COLOCH1D  Channel 1 data and limit lines.
COLOCH1M  Channel 1 memory.
COLOCH2D  Channel 2 data and limit lines.
COLOCH2M  Channel 2 memory.
COLOGRAT  Graticule.
COLOTEXT  Text.
COLOWARN  Warning.

COLOR[D]

Adjusts the color saturation for the selected display feature.

CONS  Continues the paused sequence.

CONT  Continuous sweep trigger mode.

These commands convert the S-parameter data to:

CONV1DS  Inverted S-parameters.
CONVOFF  Conversion OFF.
CONVYREF  Y:reflection.
CONVYTRA  Y:transmission.
CONVZREF  Z:reflection.
CONVZTRA  Z:transmission.

COPYFRFT  Copies the file titles into the register titles.
COPYFRR  Copy save/recall register titles to the disk register titles.
CORI<ON|OFF>  Turns interpolative error correction ON and OFF.
CORR<ON|OFF>  Turns error correction ON and OFF.
COUC<ON|OFF>  Couples and uncouples the stimulus between the channels.
COUP<ON|OFF>  Couple the power when coupled channels is turned OFF, COUCOFF.
CSWI<ON|OFF>  Selects test set continuous switching (ON) or test set hold (OFF) when there is a 2-port calibration active. Continuous switching is allowed only when the power ranges on both attenuator ports are set the same. When continuous switching is ON, the analyzer measures all four S-parameters each time before displaying the data for a full 2-port cal measurement. In test set hold mode, the analyzer measures all four S-parameters once and then measures the desired parameter continuously. This is known as a quasi 2-port cal measurement and it is less accurate than a full 2-port calibrated measurement.

CWFREQ[D]

Sets the CW frequency for power sweep and CW frequency modes. While the list frequency table segment is being edited, it sets the center frequency of the current segment.
CWTIME
Selects the CW time sweep type.

D1DIVID2<ON|OFF>
This command divides the data in channel 1 by the data in channel 2 and displays the result on channel 2.

DATI
Stores trace in channel memory. OPC-compatible.

DCONV
Selects down converter for mixer measurements.

DEBU<ON|OFF>
Turns the HP-IB debug mode ON and OFF. When ON, the HP 8753D/Option 011 scrolls incoming HP-IB commands across the display.

DECRLOOC
Decrements the sequencing loop counter by 1. NEWSEQ<1> must precede to ensure that a sequence is currently being created or modified.

DEFC
Sets the default colors for all display features.

DEFLTCPIO
Sets up the following default state for copy. HP-IB only command. There is no equivalent front-panel key.

Plotter Type: PLOTTER
Plotter Port: SERIAL
Baud Rate: 9600
Handshake: Xon-Xoff
HP-IB Address: 5
Parallel Port: COPY

Printer Type: DESKJET
Printer Port: PARALLEL
Baud Rate: 19200
Handshake: Xon-Xoff
HP-IB Address: 1

DEFLPRINT
Sets the printer to the following default setup conditions:

Print
Auto-feed
Print Colors:
Ch 1 Data
Ch 1 Memory
Ch 2 Data
Ch 2 Memory
Graticule
Warning
Text
Monochrome
On
Magenta
Green
Blue
Red
Cyan
Black
Black

DEFS[D]
Begins standard definition during cal kit modification. D is the standard number.

DELA
Displays the data formatted as group delay.

DELO
Turns the delta marker mode OFF.
These commands make the indicated marker the delta reference:

**DELR1**  Marker 1.
**DELR2**  Marker 2.
**DELR3**  Marker 3.
**DELR4**  Marker 4.

**DELRFIXM**  Fixed marker.

**DEMOAMPL**  Sets the transform demodulation to amplitude demodulation. Only has an effect with a CW time transform.

**DEMOOFF**  Turns the transform demodulation function OFF.

**DEMOPHAS**  Sets the transform demodulation to phase demodulation.

**DFLT**  Sets the plotter to the following default setup conditions:

<table>
<thead>
<tr>
<th>Plot Data</th>
<th>On</th>
<th>Pen Number</th>
<th>Ch 1</th>
<th>Ch 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plot Mem</td>
<td>On</td>
<td>Data</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Plot Grat</td>
<td>On</td>
<td>Memory</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Plot Text</td>
<td>On</td>
<td>Graticule</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Plot Mkr</td>
<td>On</td>
<td>Text</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Auto-feed</td>
<td>On</td>
<td>Marker</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Scale Plot</td>
<td>Full</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plot Speed</td>
<td>Fast</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line Type</td>
<td></td>
<td>Data</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Memory</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

**DIRS[D]**  Sets the number of files in the directory at disk initialization.

**DISCUNIT[D]**  Specifies which disk in a multiple-disk drive is to be used for disk registers.

**DISCVOLU[D]**  Specifies which volume of a multiple-volume disk drive (e.g. a Winchester) is to be used for disk registers.

**DISM<ON/OFF>**  Displays the response and stimulus values for all markers that are turned on.

These commands display the indicated combinations of data and trace memory on the active channel:

**DISPDATA**  Data only.
**DISPDATM**  Data and memory.
**DISPDDM**  Data divided by memory (linear division, log subtraction).
**DISPDMM**  Data minus memory (linear subtraction).
**DISPMEMO**  Memory only.

**DIVI**  Same as DISPDMM.
DONE

Done with a class of standards, during a calibration. Only needed when multiple standards are measured to complete the class. OPC-compatible.

DONM

Done modifying a test sequence.

DOSEQ<i>

Begin execution of the selected sequence.

DOWN

Decrements the value in the active entry area (down key).

DUAC<ON/OFF>

Dual channel display ON or OFF.

DUPLSEQ[X]SEQ[Y]

Duplicates sequence X to sequence Y.

EDITDONE

Done editing list frequency or limit table. OPC-compatible.

EDITLIML

Begin editing limit table.

EDITLIST

Begin editing list frequency table.

ELED[D]

Sets the electrical delay offset.

EMIB

Send out a beep during a sequence. NEWSEQ<i> must precede to ensure that a sequence is currently being created or modified.

ENTO

Turns the active entry area OFF.

ESB?

Outputs event-status register B.

ESE[D]

Enables the selected event-status register bits to be summarized by bit 5 in the status byte. An event-status register bit is enabled when the corresponding bit in the operand D is set.

ESNB[D]

Enables the selected event-status register B bits to be summarized by bit 2 of the status byte. Much like ESE;.

ESR?

Outputs the value of the event-status register.

EXTD

Selects the external disk as the active storage device. Used with STORESEQ<i>, TITF<i>, PURG<i>, STOR<i>, LOAD<i>, and LOADSEQ<i>.

These commands include the indicated information when a register is stored on disk. See Figure 1-1 for data types:

EXTMDATA<ON/OFF>

Error corrected data.

EXTMDATO<ON/OFF>

Data array only.

EXTMFORM<ON/OFF>

Formatted trace data.

EXTMGRA<ON/OFF>

User graphics.

EXTMRRAW<ON/OFF>

Raw data arrays.

EXTTHIGH

Sets the external trigger polarity high.

EXTTLOW

Sets the external trigger polarity low.

EXTTOFF

Deactivates the external trigger mode. OPC-compatible.

EXTTON

Activates the external trigger on sweep mode. OPC-compatible.

EXTTPOIN

Sets the external trigger to auto trigger on point. OPC-compatible.
**FIXE**
Specifies a fixed load, as opposed to a sliding load, when defining a standard during a cal kit modification.

**FOCU[D]**
Adjusts display focus, 0 to 100 percent.

These commands set the data format for array transfers in and out of the instrument:

**FORM1**
HP 8753D/Option 011 internal format. Preceded by 4 byte header.

**FORM2**
32 bit floating point format. Preceded by 4 byte header.

**FORM3**
64 bit floating point format. Preceded by 4 byte header.

**FORM4**
ASCII format. No header.

**FORM5**
32 bit floating point PC format. Bytes reversed.

These commands define the format to use on disk initializations:

**FORMATDOS**
Selects DOS as the disk format.

**FORMATLIF**
Selects LIF as the disk format.

**FREQOFFS<ON|OFF>**
Activates the frequency offset instrument mode. OPC-compatible.

**FREO**
Frequency blank. Turns OFF frequency notation.

**FRER**
HP-IB free run. Acts the same as CONT; .

**FULP**
Selects full page plotting, as opposed to plotting in one of the four quadrants.

These commands select a forward calibration class, during a 2-port calibration sequence. They are OPC-compatible if there is only one standard in the class:

**FWDI**
Isolation.

**FWDM**
Match.

**FWDT**
Transmission.

These commands control the time domain gate (available only with option 010, time domain):

**GATECENT[D]**
Center time.

**GATEO<ON|OFF>**
Gate ON/OFF.

**GATESPAN[D]**
Span time.

**GATESTAR[D]**
Start time.

**GATESTOP[D]**
Stop time.

These commands set the gate shape:

**GATSMAXI**
Maximum.

**GATSMINI**
Minimum.

**GATSNORM**
Normal.

**GATSWIDE**
Wide.

**GOSUB**
Invokes a sequence as a subroutine. Used with SEQ<↓>. 
These commands activate the harmonic measurement mode, option 002. They are all OPC-compatible:

**HARMOFF**
Turns OFF harmonic mode.

**HARMSSEC**
Measures the second harmonic.

**HARMTTHIR**
Measures the third harmonic.

**HOLD**
Puts the sweep trigger into hold.

**IDN?**
Outputs the identification string: “HEWLETT PACKARD,8755D,0,X.XX”, where X.XX is the firmware revision of the instrument.

**IFBIHIGH**
Tests the specified input GPIO bit (PARAIN). If high, invokes the sequence specified by SEQ<i>.

**IFBILOW**
Tests the specified input GPIO bit (PARAIN). If low, invokes the sequence specified by SEQ<i>.

**IFBW[D]**
Sets the IF bandwidth.

These commands branch an executing sequence to a new sequence if the following condition is satisfied. NEWSEQ<i> must precede to ensure that a sequence is currently being created or modified:

**IFLCEQZESEQ<i>**
If loop counter equals zero, then do sequence <i>.

**IFLCNEZSEQ<i>**
If loop counter does not equal zero, then do sequence <i>.

**IFLTFAILSEQ<i>**
Limit test fails.

**IFLTPASSSEQ<i>**
Limit test passes.

**IMAG**
Selects the imaginary display format.

**INCRLOOC**
Increments the sequencing loop counter by 1. NEWSEQ<i> must precede to ensure that a sequence is currently being created or modified.

**INID**
Initializes the internal disk. All information on the disk will be destroyed.

**INIE**
Initializes the external disk. All information on the disk will be destroyed. Requires pass control when using the HP-IB port.

These commands input individual calibration coefficient arrays. Before sending the array, issue a CALIXXX; command, where XXX specifies the calibration type of the data. Then input the cal arrays. Lastly store the data with SAVE;. The instrument goes into hold, displaying uncorrected data: SING; completes the process by displaying error corrected data. See Table 1-5, for the contents of the different arrays.

**INPUCALCO1[D]**
Array 1.

**INPUCALCO2[D]**
Array 2.

**INPUCALCO3[D]**
Array 3.

**INPUCALCO4[D]**
Array 4.

**INPUCALCO5[D]**
Array 5.

**INPUCALCO6[D]**
Array 6.
INPUCALC08[D] Array 8.
INPUCALC10[D] Array 10.
INPUCALC11[D] Array 11.
INPUCALC12[D] Array 12.
INPUCALK[D] Inputs a cal kit read out with OUTPCALK; . After the transfer, the
data should be saved into the user cal kit area with SAVEUSEK;.
INPUDATA[D] Inputs an error corrected data array, using current format. The
instrument stops sweeping, and then formats and displays the data.
INPUFORM[D] Inputs a formatted data array, using current format. The instrument
stops sweeping and displays the data.
INPULEEAS[D] Inputs a learn string read out by OUTPLEAS;.

These commands input power meter calibration arrays into the instrument. Values should be
entered as 100 x desired source power in dB.

INPUPCM1[D] Channel 1.
INPUPCM2[D] Channel 2.

These commands input a raw data array using the current format. See OUTPRAW for the
meaning of the arrays. The instrument stops sweeping, error corrects the data, then formats
and displays the data.

INPURAW1[D] Array 1.
INPURAW2[D] Array 2.
INPURAW3[D] Array 3.

These commands select the instrument mode. They are all OPC-compatible.

INSMMEXSA External source, auto.
INSMMEXSM External source, manual.
INSMMNETA Standard network analyzer.
INSMCNTNR Tuned receiver.

INTD Selects the internal disk as the active storage device. Used with
STORESEQ<1>, TITF<1>, PURG<1>, STOR<1>, LOAD<1>, and
LOADSEQ<1>.

INTE[D] Sets the display intensity, 0 to 100 percent.

INTM Selects the internal memory as the active storage device.

ISOD Done with isolation subsequence in a 2-port calibration.
OPC-compatible.

ISOL Begins the isolation subsequence step in a 2-port calibration.

ISOOP Begins the isolation subsequence for one path, two port calibration.
KEY[D] Sends a keycode, equivalent to actually pressing the key. It does not matter if the front-panel is in remote mode. See Figure 1-2 for the key codes.

KITD Calibration kit done: the last step in modifying a cal kit.

KOR? Outputs a two byte key code/knob count. If the number is positive, it is a key code. If it is negative, then set bit 15 equal to bit 14. The resulting integer is the knob count, either positive or negative, depending on the direction of turn (negative for clockwise.) There are approximately 120 counts per knob turn.

These commands enter labels for the standard classes during a cal kit modification:

LABEFWDM[$] Forward match.
LABEFWDT[$] Forward transmission.
LABERESI[$] Response, response and isolation.
LABERESP[$] Response.
LABEREV[$] Reverse match.
LABEREVT[$] Reverse transmission.
LABES11A[$] S1A (opens).
LABES11B[$] S1B (shorts).
LABES11C[$] S1C (loads).
LABES22A[$] S22A (opens).
LABES22B[$] S22B (shorts).
LABES22C[$] S22C (loads).
LABETLFM[$] TRL, Line, Forward, Match
LABETLFT[$] TRL, Line, Forward, Trans
LABETLRT[$] TRL, Line, Reverse, Match
LABETLRT[$] TRL, Line, Reverse, Trans
LABETRFM[$] TRL, Reflect, Forward, Match
LABETRMR[$] TRL, Reflect, Reverse, Match
LABETTFM[$] TRL, Thru, Forward, Match
LABETTFT[$] TRL, Thru, Forward, Trans
LABETTRM[$] TRL, Thru, Reverse, Match
LABETTRT[$] TRL, Thru, Reverse, Trans
LABK[$] Enters a cal kit label during a cal kit modification.
LABS[$] Enters a standard's label during standard definition.

LEFL Selects a plot in the left lower quadrant.
LEFU Selects a plot in the left upper quadrant.
LIMIAMP[D] Enters the limit line amplitude offset.
LIMILINE<ON|OFF> Turns the display of the limit lines ON and OFF.
LIMIMAOF[D]  Marker to limit offset. Centers the limit lines about the current marker position using the limit amplitude offset function.

LIMISTIO[D]  Enters the stimulus offset of the limit lines.

LIMTEST<ON|OFF>  Turns limit testing ON and OFF.

These commands edit a limit test segment. The limit table editing is begun with EDITLML; and a segment is brought up for editing with either SADD; or SEDIN; . The segment is closed with SDON; , the table is closed with EDITDONE;

LIMD[D]  Sets the limit delta value while editing a limit line segment.

LIML[D]  Sets the lower limit value.

LIMM[D]  Sets the middle limit value.

LIMS[D]  Sets the limit stimulus break point.

LIMITFL  Make the segment a flat line.

LIMITSL  Make the segment a sloping line.

LIMITSP  Make the segment a single point.

LIMU[D]  Set the upper limit value.

LINFREQ  Selects a linear frequency sweep.

LINM  Selects the linear magnitude display format.

LINTDATA[D]  Enters the line type for plotting data.

LINTMEMO[D]  Enters the line type for plotting memory.

LISFREQ  Selects the list frequency sweep mode.

LISV  Activates the list values function. The next page of values can be called with NEXP; . The current page can be plotted or printed, in raster graphics mode, with PLOT; , or PRINALL; . The entire list can be printed, in ASCII text mode, with PRINTALL;

These commands load the file from disk with the name indicated by the previous TITFn command. The actual file recalled depends on the file title in the file position specified. Requires pass control mode when using the HP-IB port.

LOAD1  Loads the file from disk using the file name provided by the preceding TITF1; command.

LOAD2  Loads the file from disk using the file name provided by the preceding TITF2; command.

LOAD3  Loads the file from disk using the file name provided by the preceding TITF3; command.

LOAD4  Loads the file from disk using the file name provided by the preceding TITF4; command.

LOAD5  Loads the file from disk using the file name provided by the preceding TITF5; command.
These commands load the indicated sequence from disk. Requires pass control mode when using the HP-IB port.

LOADSEQ1 Loads sequence 1 from disk.
LOADSEQ2 Loads sequence 2 from disk.
LOADSEQ3 Loads sequence 3 from disk.
LOADSEQ4 Loads sequence 4 from disk.
LOADSEQ5 Loads sequence 5 from disk.
LOADSEQ6 Loads sequence 6 from disk.

LOGFREQ Selects a log frequency sweep.
LOGM Selects the log magnitude display format.
LOOC[D] Sets the value of the sequencing loop counter. NEWSEQ<1> must precede to ensure that a sequence is currently being created or modified.
LOWPIMPUL Turns ON the low pass impulse transform (option 010).
LOWPSTEP Turns ON the low pass step transform (option 010).
LRN? Same as OUTPLEAS.
LRN[D] Same as INPULEAS.

MANTRIG Sets the external trigger to manual trigger on point. OPC-compatible.

These commands make the indicated marker active and set its stimulus:

MARK2[D] Marker 2.
MARK3[D] Marker 3.

MARKBUCK[D] Places the marker on a specific sweep point (bucket). D is the bucket number, ranging from 0 to number of points less 1.
MARKCENT Enters the marker stimulus as the center stimulus.
MARKCONT Places the markers continuously on the trace, not on discrete sample points.
MARKCOUP Couples the markers between the channels, as opposed to MARKUNCO.
MARKCW Sets the CW frequency to the marker frequency.
MARKDELA Sets electrical length so group delay is zero at the marker stimulus.
MARKDISC Places the markers in discrete placement mode.
MARKFAUV[D] Sets the auxiliary value of the fixed marker position. Works in coordination with MARKFVAL and MARKFSTI.
MARKFSTI[D]  Sets the stimulus position of the fixed marker.
MARKFVAL[D]  Sets the value of the fixed marker position. See Table 1-6 for the meaning of value and auxiliary value as a function of display format.
MARKMAXI    Same as SEAMAX.
MARKMIDD    During a limit segment edit, makes the marker amplitude the limit segment middle value.
MARKMINI    Same as SEAMIN.
MARKOFF     Turns all markers and marker functions OFF.
MARKREF     Enters the marker amplitude as the reference value.
MARKSPAN    Enters the span between the active marker and the delta reference as the sweep span.
MARKSTAR    Enters the marker stimulus as the start stimulus.
MARKSTIM    During a limit segment edit, enters the marker stimulus as the limit stimulus break point.
MARKSTOP    Enters the marker stimulus as the stop stimulus.
MARKUNCO    Uncouples the markers between channels, as opposed to MARKCOUP.
MARKZERO    Places the fixed marker at the active marker position and makes it the delta reference.
MAXF[D]     Sets the maximum valid frequency of a standard being defined during a cal kit modification.
MEASA       Measures and displays input A on the active channel.
MEASB       Measures and displays input B on the active channel.
MEASR       Measures and displays input R on the active channel.
MEASTAT<ON|OFF> Turns trace statistics ON and OFF.
MENU<ON|OFF> Blanks the softkey menu.

These commands bring up the menu associated with the indicated front-panel key:

MENU AVG
MENUCAL CAL
MENUCOPY COPY
MENUDISP DISPLAY
MENUFORM FORMAT
NUMMARK MARKER
NUMMEAS MEAS
NUMMRKF MARKER FCTN
NUMRECA SAVE RECALL
NUMSCL SCALE
NUMSEQE SEQ
MENUSTIM  STIMULUS MENU
MENUSYST  SYSTEM
MINF[D]  Sets the minimum valid frequency of a standard being defined during a cal kit modification.
MINU  Displays data minus memory, the same as DISPDMM.
MODII  Begins the modify cal kit sequence.

NEWSEQ<I>  Begin modifying a sequence.
NEXP  Displays the next page of the operating parameters list.
NOOP  No operation.  OPC-compatible.
NUMG[D]  Activates D number of groups of sweeps.  A group is whatever is needed to update the current parameter once.  This function restarts averaging if ON.  OPC-compatible.
NUMR[D]  Sets the number of power meter readings per point used during a power meter calibration.

These commands specify the offset value for the indicated parameter for a standard being defined during a cal kit modification:

OFSD[D]  Delay offset.
OFSL[D]  Loss offset.
OFSZ[D]  Impedance offset.

OMII  Omits the isolation step of a calibration sequence.
OPC  Operation complete.  Reports the completion of the next command received by setting bit 0 in the event-status register, or by replying to an interrogation if OPC?; is issued.  See “Command Interrogate” earlier in this chapter.

OPEP  Presents a list of key operating parameters.  NEXP; scrolls to the next page of parameters.  Requesting a plot or print copies the current page.  The current page can be plotted or printed, in raster graphics mode, with PLCT;; or PRINALL;;.  The entire list can be printed, in ASCII text mode, with PRINTALL;;.

OUTPACTI  Outputs the value of the active function, or the last active function if the active entry area is OFF.

OUTPAPER  Outputs the smoothing aperture in stimulus units, rather than as a percentage.

These commands output the error correction arrays for the active calibration on the active channel.  See Table 1-5, for the contents of the arrays.  Each array comes out in the current output format.  They contain real/imaginary pairs, the same number of pairs as points in the sweep.

OUTPCALC01  Array 1.
OUTPCALC02  Array 2.
OUTPCALC03  Array 3.
OUTPCALC04 Array 4.
OUTPCALC05 Array 5.
OUTPCALC06 Array 6.
OUTPCALC07 Array 7.
OUTPCALC08 Array 8.
OUTPCALC09 Array 9.
OUTPCALC10 Array 10.
OUTPCALC11 Array 11.
OUTPCALC12 Array 12.

These commands output over HP-IB the interpolated calibration coefficient arrays for the active calibration on the active channel.

OUTPICALC01 Array 1.
OUTPICALC02 Array 2.
OUTPICALC03 Array 3.
OUTPICALC04 Array 4.
OUTPICALC05 Array 5.
OUTPICALC06 Array 6.
OUTPICALC07 Array 7.
OUTPICALC08 Array 8.
OUTPICALC09 Array 9.
OUTPICALC10 Array 10.
OUTPICALC11 Array 11.
OUTPICALC12 Array 12.

These commands output over HP-IB the interpolated power meter calibration arrays for channels 1 and 2.

OUTPIPMCAL1 Channel 1.
OUTPIPMCAL2 Channel 2.

OUTPCALK Outputs the currently active calibration kit, as a less than 1000 byte string. The data is in form 1.

OUTPDATA Outputs the error corrected data from the active channel in the current format. See Figure 1-1.

OUTPERRO Outputs the oldest error message in the error queue. Sends first the error number, and then the error message itself as a string no longer than 50 characters.

OUTPFORM Outputs the formatted display data array from the active channel in the current format. See Table 1-6 for the contents of the array positions as a function of display format.

OUTPIDEN Outputs the identification string for the analyzer: “HEWLETT PACKARD,8753D,0,X.XX” where X.XX is the firmware revision.
OUTKEY

Outputs the key code of the last key pressed. An invalid key is reported with a 63, a knob turn with a -1. See Figure 1-2 for the front-panel key codes.

OUTPLEAS

Outputs the learn string, which contains the entire front panel state, the limit table, and the list frequency table. It is always in form 1.

These commands output the limit test results. The results consist of four fields. First is the stimulus value for the point. Second is an integer indicating test status. Third is the upper limit at that point. Fourth is the lower limit at that point. If there are no limits at that point, the third and fourth fields are zero. The test status is -1 for no test, 0 for fail, and 1 for pass.

OUTPLIMF

Outputs the limit test results for each failed point.

OUTPLIML

Outputs the limit test results for each point in the sweep. This is a form 4 transfer.

OUTPLIMM

Outputs the limit test results at the marker.

OUTPMARK

Outputs the marker values. The first two numbers are the marker response values, and the last is the stimulus value. See Table 1-6 for the meaning of the response values as a function of display format.

OUTPMEMO

Outputs the memory trace from the active channel. The data is in real/imaginary pairs, and can be treated the same as data read with the OUTPDATA command.

OUTPMSTA

Outputs the marker statistics: mean, standard deviation, and peak-to-peak variation in that order. If statistics is not ON, it is turned ON to generate current values and turned OFF again.

OUTPMWID

Outputs the marker bandwidth search results: bandwidth, center, and Q in that order. If widths is not ON, it is turned ON to generate current values and turned OFF again.

OUTPMWIL

Outputs the marker bandwidths search results: bandwidth, center, loss, and Q in that order. If widths is not ON, it is turned ON to generate current values and turned OFF again.

OUTPPLOT

Outputs the plot string to the HP-IB ports. Can be directed to a plotter, or read into the computer. PSOFT<ON|OFF> controls whether the softkeys are included in the plot.

These commands output the power meter calibration array. Values are sent as 100 times the source power in dB. A default array is used if a power meter calibration sweep, TAKCS, has not been taken:

OUTPPMCAL1

Channel 1.

OUTPPMCAL2

Channel 2.

OUTPPRIN

Outputs to the HP-IB port a raster dump of the display, intended for a graphics printer. PSOFT<ON|OFF> controls whether the soft keys are included in the printout.

OUTPPRNALL

Prints all list values or operating and marker parameters in text mode to HP-IB. Activate desired function with LISV for values or OPEP for operating parameters.
These commands output the raw measurement data. See Figure 1-1 for the meaning of the data. Normally, array 1 holds the current parameter. If a 2-port calibration is active, the arrays hold S11, S21, S12, and S22, respectively:

**OUTPRAW1** Array 1.
**OUTPRAW2** Array 2.
**OUTPRAW3** Array 3.
**OUTPRAW4** Array 4.

**OUTPRFR** Outputs the external source RF frequency. The instrument must be in external source mode, either INSMECSA or INSEXSM.

**OUTPSEQ<i>** Outputs a sequence listing over HP-IB.
**OUTPSTAT** Outputs the status byte.
**OUTPTITL** Outputs the display title.

**PARAIN[D]** Specified the input GPIO bit to be used by IFBIHIGH and IFBILOW tests.

**PARAL<GPIO|CPY>** Selects use of the parallel port: for general purpose I/O or for the copy function.

**PARAOUT[D]** Programs all GPIO output bits (0 to 255) at once.

**PAUS** Inserts a pause into a sequence. NEWSEQ<i>** must precede to ensure that a sequence is currently being created or modified.

**PCB[D]** Same as ADDRCONT. Indicates where control will be passed in pass control mode.

These commands select the color for the indicated display feature where **<COLOR>** equals one of the following colors: white, cyan, magenta, blue, yellow, green, red, or black.

**PCOLDATA1<COLOR>** Channel 1 data.
**PCOLDATA2<COLOR>** Channel 2 data.
**PCOLMEMO1<COLOR>** Channel 1 memory.
**PCOLMEMO2<COLOR>** Channel 2 memory.
**PCOLGRAT<COLOR>** Graticule.
**PCOLTEXT<COLOR>** Display text.
**PCOLWARN<COLOR>** Warning text.
**PDATA<ON|OFF>** Selects whether trace data is plotted.

These commands select the pen for plotting the indicated display feature for the active channel:

**PENNDATA[D]** Data trace.
**PENNGRAT[D]** Graticule.
**PENNMARK[D]** Markers and marker text.
**PENNMEMO[D]** Memory trace.
**PENNTEXT[D]** Text and user graphics.
PGRAT<ON|OFF>  Selects whether the graticule is plotted.
PHA[O|D]      Sets the phase offset.
PHAS          Selects the phase display format.
PLOS<SL[OW|AST] Selects the pen speed for plotting.
PLOT          Initiates a plot. Requires pass control mode when using the HP-IB port.
PLTPTD<P>     Plots measurement results to a disk for later retrieval.
PLTNDSHK<DTR|XON> Selects the plotter handshake mode as either Xon-Xoff or DTR-DSR.
PLTPRTP<IB>   Sets the plotter port to HP-IB.
PLTPR<PARA   Sets the plotter port to parallel.
PLTPRTSERI   Sets the plotter port to serial.
PLTTRAUTO<ON|OFF> Turns ON and OFF the plotter auto feed.
PLTTRAU<BAUD|D> Sets the plotter baud rate.
PLTTRFO<RF>   Sends a form feed to the plotter.
PLTTYPHG<GL>  Selects HPGL-compatible printer as the plotter type.
PLTTYPP<PLTR> Selects plotter as the plotter type.
PMEM<ON|OFF>   Selects whether memory is plotted.
PMKR<ON|OFF>   Selects whether markers are plotted.
PMTRTTTTT     Reads power meter/HP-IB value into title string. NEWSEQ<1> must precede to ensure that a sequence is currently being created or modified.
POIN[D]       Sets the number of points in the sweep. If a list frequency segment is being edited, sets the number of points in the segment.
POLA          Selects the polar display format.

These commands select the marker readout format for polar display:
POLMLIN       Linear markers.
POLLOG        Log markers.
POLMRI        Real/imaginary markers.

PORE<ON|OFF>   Turn port extensions ON and OFF.

These commands set the port extension length for the indicated port or input. Ports 1 and 2 refer to the test set ports:
PORT1[D]      Port 1.
PORT2[D]      Port 2.
PORTA[D]      Input A.
PORTB[D]      Input B.
PORTP<CPLD|UNCPLD> Selects either coupled or uncoupled for the port powers for a given channel.
These commands set the port extension length for the indicated port or input. Ports 1 and 2 refer to the test set ports:

**PORTR**  
Reflection port (equivalent to PORT1).

**PORTT**  
Transmission port (equivalent to PORT2).

**POWE[D]**  
Sets the output power level.

**POWLFREQ[D]**  
Selects the frequency for which a power loss correction is entered. This must be followed by a POWLLOSS[D], which sets the value.

**POWLLIST**  
 Begins editing a power loss list for a power meter calibration.

**POWLLOSS[D]**  
Sets the loss value for a particular frequency, POWLFREQ[D], in the power loss list.

**POWM<ON|OFF>**  
Selects whether the HP 436A (ON) or the HP 437B/438A (OFF) is to be used as the power meter in service procedures.

**POWS**  
Selects power sweep, from the sweep type menu.

**POWT<ON|OFF>**  
Turning power trip OFF clears a power trip after an overload condition is detected at one of the input ports.

**PRAN0**  
Selects power range 0 when in manual power range. Used with PWRR and PMAN.

**PRAN1**  
Selects power range 1 when in manual power range. Used with PWRR and PMAN.

**PRAN2**  
Selects power range 2 when in manual power range. Used with PWRR and PMAN.

**PRAN3**  
Selects power range 3 when in manual power range. Used with PWRR and PMAN.

**PRAN4**  
Selects power range 4 when in manual power range. Used with PWRR and PMAN.

**PRAN5**  
Selects power range 5 when in manual power range. Used with PWRR and PMAN.

**PRAN6**  
Selects power range 6 when in manual power range. Used with PWRR and PMAN.

**PRAN7**  
Selects power range 7 when in manual power range. Used with PWRR and PMAN.

**PRES**  
Presets the analyzer to the factory preset state.

**PRIC**  
Selects color print.

**PRINALL**  
Copies the display, in raster graphics mode, to a printer. Requires pass control mode when using the HP-IB port.

**PRINSEQ<1>|>**  
Begins printing the sequence selected. Requires pass control mode when using the HP-IB port.

**PRINTALL**  
Prints all list values or operating and marker parameters in ASCII text mode. Requires pass control mode when using the HP-IB port.

**PRIS**  
Selects standard (monochrome) print.

**PRNHNDSHK<DTR|XON>**  
Selects the printer handshake mode as either Xon-Xoff or DTR-DSR.
PRNPRTPIB    Sets the printer port to HP-IB.
PRNPRTPARA   Sets the printer port to parallel.
PRNPRTSERI   Sets the printer port to serial.
PRNTRAUTF<ON|OFF>  Turns ON and OFF the printer auto feed.
PRNTRBAUD[D]  Sets the printer baud rate.
PRNRFRF      Sends a form feed to the printer.
PRNTYPDJ     Selects the DeskJet printer as the printer type.
PRNTYPEP     Selects the Epson ESC/P2 printer control language-compatible
             printer as the printer type.
PRNTYPLJ     Selects the LaserJet printer as the printer type.
PRNTYPJPJ    Selects the PaintJet printer as the printer type.
PRNTYPTJ     Selects the ThinkJet printer as the printer type.
PSOFT<ON|OFF> Controls whether softkeys are included in the hardcopy print
             or plot when using one of the following commands: PLOT;,
             PRIMALL; OUTPLOT; or OUTPPRINT;.
PTEXT<ON|OFF> Selects whether text is plotted.
PTOS         Pauses the sequence to be followed by selection one of the 6
             sequences (SEQ<I>).

These commands purge the indicated file from disk. Requires pass control mode when using the
HP-IB port.
PURG1        File 1.
PURG2        File 2.
PURG3        File 3.
PURG4        File 4.
PURG5        File 5.

These commands select the type of power meter calibration desired. A calibration sweep
should be taken, TAKCS, after selecting a "one sweep" Power meter calibration, to ensure a
valid calibration. No calibration sweep is needed for "each sweep" power meter calibrations.
PWMCEACS[D]  Each sweep.
PWMCOFF[D]   Off.
PWMCONES[D]  One sweep.
PWRLOSS<ON|OFF> Selects whether or not to use the power loss table for a power meter
             calibration.
PWRMCMAL[D]  Displays the power meter cal menu and sets the drive port cal
             power.
PWRRA<PAUTO|PMAN> Select the source power range auto or manual mode.
RAID         Completes the response and isolation cal sequence. OPC-compatible.
RAISOL
Calls the isolation class for the response and isolation calibration.
OPC-compatible if only one standard in class.

RAIRESP
Calls the response class for the response and isolation calibration.
OPC-compatible if only one standard in class.

READDRATE
Outputs the date in the following string format: DD MMM YYYY.
HP-IB only command. There is no front-panel equivalent.

READTIME
Outputs the time in the following string format: HH:MM:SS. HP-IB
only command. There is no front-panel equivalent.

REAL
Selects the real display format.

RECO
Recalls previously saved display colors.

These commands recall the indicated internal register. They are all OPC-compatible:

RECA1
Register 1.

RECA2
Register 2.

RECA3
Register 3.

RECA4
Register 4.

RECA5
Register 5.

RECAREG<1>
Recalls save/recall registers 01 through 31. RECAREG01 through
RECAREG05 are the same as RECA1 through RECA5.

REFD
Completes the reflection calibration subsequence of a 2-port
calibration. OPC-compatible.

REFL
Begins the reflection calibration subsequence of a 2-port calibration.

REFOP
Begins the reflection calibration subsequence for one path, two port
calibration.

REFP[D]
Enters the reference position. 0 is the bottom, 10 is the top of the
graticule.

REFT
Recall file titles from disk. Requires pass control mode when using
the HP-IB port.

REFV[D]
Enters the reference line value.

RESC
Resume cal sequence.

RESD
Restores the measurement display after viewing the operating
parameters or list values.

RESPDONE
Completes the response calibration sequence. OPC-compatible.

REST
Measurement restart.

These commands call the reverse calibration classes, during a full 2-port calibration. They are
OPC-compatible if there is only one standard in the class:

REVI
Isolation.

REVM
Match.

REVX
Transmission.
These commands are used in mixer measurements.

**RFGTLO**  
Sets RF greater than LO.

**RFLTLO**  
Sets RF less than LO.

**RIGL**  
Selects a plot in the lower right quadrant.

**RIGU**  
Selects a plot in the upper right quadrant.

**RSCO**  
Resets colors for the selected group.

**RST**  
Presets the instrument. OPC-compatible.

These commands select the parameter displayed on the active channel:

**S11**

**S12**

**S21**

**S22**

**SADD**  
During either a list frequency or limit table edit, adds a new segment to the table.

**SAMC<ON/OFF>**  
Turns sampler correction ON and OFF. Sampler correction is only turned off to take data for custom calibration coefficients.

**SAV1**  
Completes the 1-port calibration sequence. OPC-compatible.

**SAV2**  
Completes the 2-port calibration sequence. OPC-compatible.

**SAVC**  
Completes the transfer of error correction coefficients back into the instrument. OPC-compatible.

These commands store the current instrument state in the indicated internal register. These commands are all OPC-compatible.

**SAVE1**  
Register 1.

**SAVE2**  
Register 2.

**SAVE3**  
Register 3.

**SAVE4**  
Register 4.

**SAVE5**  
Register 5.

**SAVEREG<i>**  
Saves to save/recall registers 01 through 31. SAVEREG01 through SAVEREG05 are the same as SAVE1 through SAVE5.

The following commands define the format for saving files to disk.

**SAVUASC**  
Selects ASCII format for saving to disk. Also known as citifile format.

**SAVUBFINA**  
Selects binary format for saving to disk.

**SAVEUSEK**  
Stores the active calibration kit as the user kit.

**SCAL[D]**  
Sets the trace scale factor.

**SCAP<FULL|GRAT>**  
Selects a full plot, or a plot where the graticule is expanded to P1 and P2.
SDEL
During either a list frequency or a limit table edit, deletes the current segment.

SDON
During either a list frequency or a limit table edit, closes a segment after editing.

These commands control the marker searches. The marker searches place the active marker according to the indicated search criteria. The search is continuously updated if tracking is ON:

SEAL
Search left for next occurrence of the target value.

SEAMAX
Trace maximum.

SEAMIN
Trace minimum.

SEAOFF
Turns the marker search OFF.

SEAR
Search right for next occurrence of the target value.

SEATARG[D]
Arbitrary target amplitude.

SED[D]
During either a frequency or a limit table edit, selects segment D for editing.

SELL[D]
Selects the learnstring revision (LRN) to input to or output from the analyzer. The valid parameters are:

0: Defaults to current revision.
201: Revision 8753B 2.01
300: Revision 8753B 3.00
401: Revision 8753C 4.01
402: Revision 8753C 4.02
412: Revision 8753C 4.12
413: Revision 8753C 4.13

This is an HP-IB only command. There is no front-panel equivalent.

SEQ<I>
Selects sequence 1 through 6.

SEQWAIT[D]
Tells the instrument to wait D seconds during a sequence. NEWSEQ<I> must precede to ensure that a sequence is currently being created or modified.

SETBIT[D]
Sets the specified bit (0 to 7) on the GPIO.

SETDATE[$]
Sets the date in the following format: DD MMM YYYY, where DD is the day and must be 2 digits, MMM is the month and must be three alpha characters (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC), and YYYY is the year and must be 4 digits.

SETF
Set frequency for low pass transform, option 010.

SETTIME[$]
Sets the time in the following format: HH:MM:SS, where HH is the hour, MM is minutes, SS is seconds, and each must be 2 digits.

SETZ[D]
Set the characteristic impedance of the measurement system.

SHOM
Displays the desired softkey menu during a sequence. NEWSEQ<I> must precede to ensure that a sequence is currently being created or modified.
SING  Single sweep. OPC-compatible.
SLID  Sliding load done.
SLIL  Specifies the standard as a sliding load during a standard definition as part of a cal kit modification.
SLIS  Sliding load set.
SLOPE[D]  Enters the power slope value.
SLOPO<ON|OFF>  Turns the power slope ON and OFF.
SMIC  Select Smith chart display format.
The following commands select the marker readout format on a Smith chart:
SMIMGB  G+jB.
SMIMLIN  Linear.
SMIMLOG  Log.
SMIMRI  Real/imaginary pairs.
SMIMRX  R+jX.
SMOOAPER[D]  Sets the smoothing aperture as a percent of the trace.
SMOOO<ON|OFF>  Turns smoothing ON and OFF.
The following commands press the indicated soft key:
SOFT1  Softkey 1.
SOFT2  Softkey 2.
SOFT3  Softkey 3.
SOFT4  Softkey 4.
SOFT5  Softkey 5.
SOFT6  Softkey 6.
SOFT7  Softkey 7.
SOFT8  Softkey 8.
SOFR  Displays the firmware revision on the screen.
SOUP<ON|OFF>  Turns the source power ON and OFF.
SPAN[D]  Sets the stimulus span. If a list frequency segment is being edited, sets the span of the segment.
SPEG  Displays the specify gate menu.
The following commands initiate the SPECIFY-CLASS part of modifying a cal kit. After issuing each command, send the analyzer a series of standard numbers to be included in the class, separated by commas. When the class is full, send CLAD; to terminate the sequence.
SPECFWDM  Forward match.
SPECFWDT  Forward transmission.
SPECRESP  Response.
SPECRESI      Resp & Isol, response.
SPECREVQ      Reverse match.
SPECREVTV     Reverse transmission.
SPECXS11A     S1A (opens).
SPECXS11B     S1B (shorts).
SPECXS11C     S1C (loads).
SPECXS22A     S22A (opens).
SPECXS22B     S22B (shorts).
SPECXS22C     S22C (loads).
SPECTRFM      TRL, Reflect, Forward, Match
SPECTRRM      TRL, Reflect, Reverse, Match
SPECTRLFM     TRL, Line, Forward, Match
SPECTRLFT     TRL, Line, Forward, Trans
SPECTRLRM     TRL, Line, Reverse, Match
SPECTRLRT     TRL, Line, Reverse, Trans
SPECTTTFM     TRL, Thru, Forward, Match
SPECTTFT      TRL, Thru, Forward, Trans
SPECTTRRM     TRL, Thru, Reverse, Match
SPECTTRT      TRL, Thru, Reverse, Trans

SPLD<ON|OFF>  Turns the split display mode ON and OFF.
SRE[D]        Service request enable. A bit set in D enables the corresponding bit in the status byte to generate an SRQ.
SSEG[D]       Selects the desired segment of the frequency list for a list frequency sweep.
STB?          Outputs the status byte.

The following commands select a standard from a class during a calibration sequence. If a class is requested, as in CLASS11A (open, S11 1-port cal.) the analyzer will do one of two things. If there is only one standard in the class, it will measure that standard automatically. If there are several standards in the class, then one of the following commands must be used to select one, causing it to be measured. All of these commands are OPC-compatible:

STANA         Standard listed under softkey 1.
STANB         Standard listed under softkey 2.
STANC         Standard listed under softkey 3.
STAND         Standard listed under softkey 4.
STANE         Standard listed under softkey 5.
STANF         Standard listed under softkey 6.
STANG         Standard listed under softkey 7.
STAR[D]  Enters the start stimulus value. If a list frequency segment is being edited, sets the start of the segment.

STDD  Standard done, define standard sequence, while modifying a cal kit.

The following commands select the standard type after the standard number has been entered during a modify cal kit sequence:

STDARBI  Arbitrary impedance.

STDDELA  Delay/thru.

STIDLOAD  Load.

STIDOPEN  Open.

STIDSHOR  Short.

STOP[D]  Sets the stimulus stop value. If a list frequency segment is being edited, sets the stop of the segment.

These commands store the indicated file on disk. Used with the INTD and EXTD commands to designate the internal or external disk. Requires pass control mode when using the HP-IB port.

STOR1  Stores the current instrument state to disk using the file name provided by the preceding TITF1; command.

STOR2  Stores the current instrument state to disk using the file name provided by the preceding TITF2; command.

STOR3  Stores the current instrument state to disk using the file name provided by the preceding TITF3; command.

STOR4  Stores the current instrument state to disk using the file name provided by the preceding TITF4; command.

STOR5  Stores the current instrument state to disk using the file name provided by the preceding TITF5; command.

These commands store the instrument state to the indicated sequence to disk. Used with the INTD and EXTD commands to designate the internal or external disk. Requires pass control mode when using the HP-IB port.

STORSEQ1  Sequence 1.

STORSEQ2  Sequence 2.

STORSEQ3  Sequence 3.

STORSEQ4  Sequence 4.

STORSEQ5  Sequence 5.

STORSEQ6  Sequence 6.

STPSIZE  While editing a list frequency segment, sets step size.

SVCO  Saves display colors.

SWET[D]  Sets the sweep time. Sets to 0 for auto sweep time.

SWR  Selects the SWR display format.
TAKCS
    Begins a power meter calibration sweep. Requires pass control.

TALKLIST
    Puts the analyzer in talker listener mode.

TERI[D]
    Specifies the terminal impedance of an arbitrary impedance standard
during a cal kit modification.

TESS?
    Returns a one on the standard analyzer. (The standard analyzer does
not recognize external test sets.) For option 011, returns a one if an
HP 85046A/B S-parameter test set is present. Returns a two if an
HP 85047A S-parameter test set is present.

TIMDTRAN<ON|OFF>
    Turns the option 010 (time domain) transform ON and OFF.

TIMESTAM<ON|OFF>
    Turns on the clock time for prints and plots.

TINT[D]
    Adjusts the tint for the selected display feature.

TITF0
    Titles the SAVE STATE filename in sequence mode.

These commands title the indicated file numbers:

TITF1[$]       File 1.
TITF2[$]       File 2.
TITF3[$]       File 3.
TITF4[$]       File 4.
TITF5[$1      File 5.

TITL[$]
    Enters a new CRT title. A maximum of 48 characters (alphanumeric
and mathematical symbols) are allowed.

TITP
    Titles the plot file.

These commands title the indicated internal register:

TITR1[$]       Register 1.
TITR2[$]       Register 2.
TITR3[$]       Register 3.
TITR4[$]       Register 4.
TITR5[$]       Register 5.

TITREG<1>[$]
    Titles save/recall registers 01 through 31. TITREG01 through
    TITREG05 are the same as TITR1 through TITR5.

TITSEQ<1>[$]
    Selects the sequence to be titled.

TITTMEM
    Sends the title string to trace memory. NEWSEQ<1> must precede
to ensure that a sequence is currently being created or modified.

TITTPMTR
    Sends the title string to the power meter address. NEWSEQ<1> must
precede to ensure that a sequence is currently being created or
modified.

TITTPERI
    Sends the title string to the peripheral address. NEWSEQ<1> must
precede to ensure that a sequence is currently being created or
modified.
TTTPRIN  Sends the title string to the printer address. NEWSEQ<1> must precede to ensure that a sequence is currently being created or modified.

TRACK<ON|OFF>  Turns marker search tracking ON and OFF.

TRAD  Completes the transmission calibration subsequence of a 2-port calibration. OPC-compatible.

TRAN  Begins the transmission calibration subsequence of a 2-port calibration.

TRAOP  Begins the transmission calibration subsequence for one path, two port calibration.

TRIG  HP-IB trigger. Puts instrument into hold mode.

TST?  Causes a self test and returns a zero if the test is passed.

TSTOFWD[D]  Defines 3 bits, D0 through D2, on the test set connector I/O for the channel 1 and channel 2 forward settings. These bits can be set to values of 0 through 7. Be careful that you do not also set a value to ATTP1 and ATTP2 as there is interaction between these commands and they will change the values you have set for D0 through D2 and will couple the channels together. Values for ATTP1 and ATTP2 translate to the following values for D0 through D2:

<table>
<thead>
<tr>
<th>ATTP1/ATTP2</th>
<th>D0 - D2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 dB</td>
<td>7</td>
</tr>
<tr>
<td>10 dB</td>
<td>6</td>
</tr>
<tr>
<td>20 dB</td>
<td>5</td>
</tr>
<tr>
<td>30 dB</td>
<td>4</td>
</tr>
<tr>
<td>40 dB</td>
<td>3</td>
</tr>
<tr>
<td>50 dB</td>
<td>2</td>
</tr>
<tr>
<td>60 dB</td>
<td>1</td>
</tr>
<tr>
<td>70 dB</td>
<td>0</td>
</tr>
</tbody>
</table>
TSTIOREV[D]

Defines 3 bits, D0 through D2, on the test set connector I/O for the channel 1 and channel 2 reverse settings. These bits can be set to values of 0 through 7. Be careful that you do not also set a value to ATTP1 and ATTP2 as there is interaction between these commands and they will change the values you have set for D0 through D2 and will couple the channels together. Values for ATTP1 and ATTP2 translate to the following values for D0 through D2:

<table>
<thead>
<tr>
<th>ATTP1/ATTP2</th>
<th>D0 - D2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 dB</td>
<td>7</td>
</tr>
<tr>
<td>10 dB</td>
<td>6</td>
</tr>
<tr>
<td>20 dB</td>
<td>5</td>
</tr>
<tr>
<td>30 dB</td>
<td>4</td>
</tr>
<tr>
<td>40 dB</td>
<td>3</td>
</tr>
<tr>
<td>50 dB</td>
<td>2</td>
</tr>
<tr>
<td>60 dB</td>
<td>1</td>
</tr>
<tr>
<td>70 dB</td>
<td>0</td>
</tr>
</tbody>
</table>

TSTP<P1|P2>

Selects test port 1 or 2 for non-S-parameter measurements.

These commands set the TTL output and end of sweep pulse:

TTLHPULS
Rear panel “TEST SEQ” BNC output normally low, high pulse at end of sweep.

TTLLPULS
Rear panel “TEST SEQ” BNC output normally high, low pulse at end of sweep.

TTLOH
Sets Rear panel “TEST SEQ” BNC output continuously high.

TTLOL
Sets Rear panel “TEST SEQ” BNC output continuously low.

UCONV
Selects up converter for mixer measurements.

UP
Increments the value in the active entry area (up key).

USEPASC
Puts the analyzer in pass control mode.

These commands select the sensor input being used with the HP 438A power meter. For the HP 436A or 437B, the A sensor is always used:

USESENSA
Sensor A.

USESENSB
Sensor B.

VELOFACT[D]
Enters the velocity factor of the transmission medium.

VIEM<ON|OFF>
Displays the measurement trace (ON) or the mixer setup (OFF).

VOFF[D]
Sets the frequency offset value.
WAIT                    Waits for a clean sweep. OPC-compatible.
WAVE                    Specifies a waveguide standard while defining a standard as part of a cal kit modification.
WIDT<ON|OFF>            Turns the bandwidth search ON and OFF.
WIDV[D]                 Enters the widths search parameter.

These commands set the window for the transform (option 010, time domain):

WINDMAXI                Maximum.
WINDMINI                Minimum.
WINDNORM                Normal.
WINDOW[D]               Enters arbitrary window.
WINDUSEM<ON|OFF>        Turns the trace memory ON as the window shape.

These commands enter new softkey labels into the indicated softkey positions:

WRSK1[$]                Softkey 1.
WRSK2[$]                Softkey 2.
WRSK3[$]                Softkey 3.
WRSK4[$]                Softkey 4.
WRSK5[$]                Softkey 5.
WRSK6[$]                Softkey 6.
WRSK7[$]                Softkey 7.
WRSK8[$]                Softkey 8.
Index

Special characters

\$ , 1-8

A

A/B, 1-42
AB, 1-42
acceptor handshake, 1-2
ADDRCONT[D], 1-42
ADDRDISC[D], 1-42
address
controller, 1-42
disk drive, 1-42
peripheral, 1-42
plotter, 1-42
power meter, 1-42
printer, 1-42
ADDRPERI[D], 1-42
ADDRPLOT[D], 1-42
ADDRPOW[M][D], 1-42
ADDRPRIN[D], 1-42
adjust brightness, 1-43
adjust color, 1-45
adjust tint, 1-69
AF, 1-36
AH1, 1-2
ALTAB, 1-42
alternate inputs, 1-42
ANAB, 1-42
ANA1, 1-42
analog bus, 1-42
analog input, 1-42
analyzer HP-IB capabilities, 1-2
analyzer identification, 1-4
appendage, 1-5
AR, 1-42
ASCII
save format, 1-64
ASEG, 1-42
assert sequence, 1-42
ASSS, 1-42
ATTP1, 1-42
ATTP2, 1-42
AUTO, 1-42
auto feed
plotter, 1-60
printer, 1-62
auto scale, 1-42
averaging, 1-42
restart, 1-42
averaging factor, 1-42
AVERFACT[D], 1-42
AVERO, 1-42
AVERREST, 1-42
B

BACI[D], 1-42
background intensity, 1-42
BANDPASS, 1-42
basic listener, 1-2
basic talker, 1-2
baud rate
plotter, 1-60
printer, 1-62
beep
emit, 1-48
BEEPDONE, 1-42
beeper on done, 1-42
beeper on warning, 1-42
BEEPFAIL, 1-42
BEEPWARN, 1-42
begin cal sequence, 1-43
binary
save format, 1-64
BR, 1-42
C

C1, 1-2
C10, 1-2
C1[D], 1-43
C2, 1-2
C2[D], 1-43
C3, 1-2
C3[D], 1-43
CALI, 1-43
CALFCALF[D], 1-43
CALFFREQ[D], 1-43
CALFSENA, 1-43
CALFSENB, 1-43
calibration
power meter, 1-62
calibration arrays, 1-34
calibration/classes relationship, 1-34
calibration command sequence, 1-34
Calibration kits, 1-43
Calibration type off, 1-43
CALIFUL2, 1-43
CALIONE2, 1-43
CALIRAI, 1-43
CALIRESP, 1-43
CALIS111, 1-43
CALIS221, 1-43
CALITRL2, 1-43
CALK35MD, 1-43
CALK35MM, 1-43
CALK7MM, 1-43
cal kit done, 1-52
CALKN50, 1-43
CALKN75, 1-43
CALKUSED, 1-43
CALN, 1-43
cal power
  set port 1, 1-62
cal sensor table
  edit, 1-43
cal sequence
  begin, 1-43
  resume, 1-63
carriage returns, 1-6
case distinctions, 1-5
CBRI[D], 1-43
CENT[D], 1-43
center, 1-43
CHAN1, 1-43
CHAN2, 1-43
channels
  coupled, 1-45
CHOPAB, 1-43
chritile
  save format, 1-64
CLAD, 1-44
CLASS11A, 1-44
CLASS11B, 1-44
CLASS11C, 1-44
CLASS22A, 1-44
CLASS22B, 1-44
CLASS22C, 1-44
class done, 1-44
CLEABIT, 1-44
CLEA<i>, 1-44
CLEAL, 1-44
CLEARALL, 1-44
CLEAREG<i>, 1-44
clear list, 1-44
clear register, 1-44
clear sequence, 1-44
CLOSESEQ<i>, 1-44
CLEL, 1-44
CLES, 1-44
CLS, 1-44
COAX, 1-44
CO[D], 1-43
COLOCH1D, 1-45
COLOCH1M, 1-45
COLOCH2D, 1-45
COLOCH2M, 1-45
COLOGRAT, 1-45
color
  data channel 1, 1-59
data channel 2, 1-59
  graticule, 1-59
  memory channel 1, 1-59
memory channel 2, 1-59
text, 1-59
warning, 1-59
COLOR[D], 1-45
colors, 1-59
COLOTEXT, 1-45
COLOWARN, 1-45
command interrogate, 1-3
command naming, 1-7
commands
HP-IB, 1-1
command structure, 1-5
CONS, 1-45
CONT, 1-45
continue sequence, 1-45
creator
  address, 1-42
CONV1DS, 1-45
CONVOFF, 1-45
CONVRER, 1-45
CONVYTRA, 1-45
CONVZTRA, 1-45
copy display, 1-58, 1-60, 1-61
COPYFRFT, 1-45
COPYFRRT, 1-45
CORI, 1-45
CORR, 1-45
correction, 1-45
  interpolative, 1-45
COUC, 1-45
COUP, 1-45
coupled channels, 1-45
CRT focus, 1-49
CRT intensity, 1-51
CRT title, 1-69
CS, 1-36
CSWI, 1-45
cW freq, 1-45
CWREQ[D], 1-45
cW time, 1-45
CWTIME, 1-45
D
[D], 1-8
D1DIVD2, 1-46
data
  include with disk files, 1-48
data channel 1
  color, 1-59
data channel 2
  color, 1-59
data only
  include with disk files, 1-48
data processing chain, 1-38
data units, 1-4
date, 1-65
DATI, 1-46
DC1, 1-2
DCONV, 1-46
DEBU, 1-46
debug, 1-46
decimal point, 1-6
decrement loop counter, 1-46
DECRLOOC, 1-46
default calibration kits, 1-43
default colors, 1-46
DEF, 1-46
DEFLPRINT, 1-46
DEFTCPNO, 1-46
DEFS[D], 1-46
DELA, 1-46
delay, 1-46, 1-48
  set to mkr, 1-54
delete segment, 1-64
DEL<&>, 1-47
DELO, 1-46
DELRFIXM, 1-47
delta limits, 1-53
delta reference, 1-46, 1-47
DEMOAMPL, 1-47
demodulation off, 1-47
DEMOFF, 1-47
DEMOPHAS, 1-47
DeskJet, 1-62
device clear, 1-2
DF, 1-36
DFLT, 1-47
DI, 1-36
directory size
  LIF, 1-47
DIRS[D], 1-47
DISCOUNT[D], 1-47
DISCVOLU[D], 1-47
disk
  load file, 1-53
disk drive
  address, 1-42
disk drive unit, 1-47
disk drive volume, 1-47
disk file names, 1-40
disk format, 1-49
DISM, 1-47
DISPDATA, 1-47
DISPDATM, 1-47
DISPDM, 1-47
display A/B, 1-42
display A/R, 1-42
display B/R, 1-42
display data, 1-47
display data — mem, 1-47
display data & mem, 1-47
display data/mem, 1-47
display data to mem, 1-46
display graphics, 1-35
display memory, 1-47
DISPMEMO, 1-47
DISPM, 1-47
DIV, 1-47
done
  with class, 1-47
  with isolation, 1-251
  with reflection, 1-63
  with transmission, 1-70
DONE, 1-47
done modify sequence, 1-48
DONM, 1-48
DOSEQ<X>, 1-48
do sequence, 1-48
DOS format, 1-49
DOWN, 1-48
down converter, 1-46
DTI, 1-2
DTR, 1-61
DUAC, 1-48
dual channels, 1-48
duplicate sequence, 1-48
DUPLSEQ<X>SEQ<X>, 1-48
E
E2, 1-2
ediit cal sensor table, 1-43
EDITDONE, 1-48
edit limit table, 1-48
EDITLIML, 1-48
EDITLIST, 1-48
edit power loss range, 1-61
edit power loss table, 1-61
edit segment, 1-65
ELED[D], 1-48
EMIB, 1-48
emit beep, 1-48
end or identify, 1-5
ENI, 1-48
entry off, 1-48
EOI, 1-5
Epson-P2, 1-62
ESB?, 1-48
ESD[D], 1-48
ESNB[D], 1-48
ESR?, 1-48
EXTD, 1-48
EXTDATA, 1-48
external source mode, 1-51
external trigger, 1-48
EXTMDATO, 1-48
EXTMFORM, 1-48
EXTMGRAP, 1-48
EXTMRAW, 1-48
EXTHIGH, 1-48
EXTLOW, 1-48
EXTTOFF, 1-48
EXTTON, 1-48
EXTPOIN, 1-48

F
file names
  disk, 1-40
file titles
  recall, 1-63
firmware revision identification, 1-4
FIXE, 1-48
fixed load, 1-48
fixed marker, 1-47
flat line type, 1-53
FOCU[D], 1-49
FORM1, 1-49
FORM2, 1-49
FORM3, 1-49
FORM4, 1-49
FORM5, 1-49
format
  disk, 1-49
FORMATDOS, 1-49
FORMATLIF, 1-49
formatted data
  include with disk files, 1-48
form feed
  plotter, 1-60
  printer, 1-62
forward calibration class, 1-49
FREO, 1-49
FREQOFFS, 1-49
frequency notation, 1-49
frequency offset, 1-49
frequency offset value, 1-71
FULP, 1-49
FWDI, 1-49
FWDM, 1-49
FWDT, 1-49

G
GATE, 1-49
GATECENT[D], 1-49
gate center time, 1-49
gate on/off, 1-49
gate shape, 1-49
  maximum, 1-49
  minimum, 1-49
  normal, 1-49
  wide, 1-49
GATESPAN[D], 1-49
gate span time, 1-49
GATESVAR[D], 1-49
gate start time, 1-49
GATSTOP[D], 1-49
gate stop time, 1-49
GATSMAXI, 1-49
GATSMINI, 1-49
GATSNORM, 1-49
GATSWIDE, 1-49
GOSUB, 1-49
gosub sequence, 1-49
GPIO, 1-59
GPIO input bit, 1-59
GPIO output bits, 1-59

graphics
  character size, 1-37
  default values, 1-36
  display off, 1-36
  display on, 1-37
  draw to x,y, 1-36
  erase display, 1-36, 1-37
  label display, 1-36
  line type, 1-36
  output scaling limits, 1-36
  pen down, 1-36
  pen up, 1-37
  plot relative, 1-37
  select pen, 1-37
graphics commands, 1-36
graticule
  color, 1-59

H
handshake
  plotter, 1-60
  printer, 1-61
HARMOFF, 1-50
harmonic mode off, 1-50
HARSEC, 1-50
HARMTHIR, 1-50
held commands, 1-2
HOLD, 1-50
HP-GL
character size, 1-37
commands accepted but ignored, 1-37
default values, 1-36
display off, 1-36
display on, 1-37
draw to x,y, 1-36
erase display, 1-36, 1-37
label display, 1-36
line type, 1-36
output scaling limits, 1-36
pen down, 1-36
pen up, 1-37
plot relative, 1-37
select pen, 1-37
HP-GL subset, 1-36
HP-IB
acceptor handshake, 1-2
analyzer capabilities, 1-2
basic listener, 1-2
basic talker, 1-2
device clear, 1-2
parallel poll, 1-2
serial poll, 1-2
source handshake, 1-2
HP-IB commands, 1-1
HP-IB only commands, 1-29

I
<M>, 1-8
identification
of analyzer, 1-4
of firmware revision, 1-4
IDN?, 1-4, 1-50
IF bandwidth, 1-50
IFBIHIGH, 1-50
IFBILOW, 1-50
IFBW[D], 1-50
IFLCEQZESEQ<i>, 1-50
IFLCNEZESEQ<i>, 1-50
IFLTFAILSEQ<i>, 1-50
IFLTPASSSEQ<i>, 1-50
IM, 1-37
IMAG, 1-50
imaginary, 1-50
increment loop counter, 1-50
INCRLOOC, 1-50
INID, 1-50
INIE, 1-50
initialize disk, 1-50
INFUCALC<i>[D], 1-50
INFUCALK[D], 1-51
INFUDATA[D], 1-51
INFUFORM[D], 1-51
INPULEAS[D], 1-51
INPUPMCAL<i>, 1-51
INPURAW<i>, 1-51
input syntax, 1-5
INSMEXSA, 1-51
INSMEXSM, 1-51
INSMNETA, 1-51
INSMTUNR, 1-51
INTD, 1-51
INTE[D], 1-51
intensity
background, 1-42
interpolative correction, 1-45
interrogate syntax, 1-6
interrogating commands, 1-3
INTM, 1-51
IP, 1-37
ISOD, 1-51
ISOOP, 1-51
IW, 1-37

K
key codes, 1-41
KEY[D], 1-51
key select codes, 1-8
KITD, 1-52
kit done, 1-52

L
L4, 1-2
LABEFWDM[$], 1-52
LABEFWDT[$], 1-52
label cal kit, 1-52
label class, 1-52
label standard, 1-52
LABERESI[$], 1-52
LABERESP[$], 1-52
LABEREVM[$], 1-52
LABEREVT[$], 1-52
LABES1A[$], 1-52
LABES1B[$], 1-52
LABES1C[$], 1-52
LABES22A[$], 1-52
LABES22B[$], 1-52
LABES22C[$], 1-52
LABETLFM[$], 1-52
LABETLFT[$], 1-52
LABETLRM[$], 1-52
LABETLRT[$], 1-52
LABETRFM[$], 1-52
LABETTRM[$], 1-52
LABETTRT[$], 1-52

Index-5
LABK[$], 1-52
LABS[$], 1-52
LaserJet, 1-62
LB, 1-36
LEFL, 1-52
LEFU, 1-52
LIF
  directory size, 1-47
  LIF format, 1-49
LIM[D], 1-53
LIMIAMPO[D], 1-52
LIMLINE, 1-52
LIMMAOF[D], 1-52
LIMISTO[D], 1-53
LIMITEST, 1-53
  limit line, 1-52
  limit line amplitude offset, 1-52
  limit line stimulus offset, 1-53
  limit table edit, 1-48
  limit test, 1-53
  limit test beeper, 1-42
  limit test fail, 1-50
  limit test pass, 1-50
LIML[D], 1-53
LIMM[D], 1-53
LIMS[D], 1-53
LIMTPL, 1-53
LIMTSL, 1-53
LIMTSP, 1-53
LIMU[D], 1-53
  linear sweep, 1-53
  line feeds, 1-5, 1-6
  line type data, 1-53
    memory, 1-53
LINFREQ, 1-53
LINM, 1-53
lin mag, 1-53
LINTDATA[D], 1-53
LINTMEMO[D], 1-53
LISFREQ, 1-53
  list clear, 1-44
  list sweep, 1-53
  list values, 1-53
    print, 1-61
LISV, 1-53
LOAD<$>, 1-53
LOADSEQ<$>, 1-54
local lockout, 1-2
LOGFREQ, 1-54
LOGM, 1-54
log mag, 1-54
log sweep, 1-54
LOOC[D], 1-54
  loop counter decrement, 1-46
  increment, 1-50
  loop counter value, 1-54
  lower case, 1-5, 1-6
  lower limit segment, 1-53
  low pass frequency, 1-65
  low pass impulse, 1-54
  low pass step, 1-54
LOWPIMPU, 1-54
LOWPSTEP, 1-54
LIta, 1-36
M
MANTRIG, 1-54
MARKCENT, 1-54
MARKCONT, 1-54
MARKCOUP, 1-54
MARKCW, 1-54
MARKDELA, 1-54
MARKDISC, 1-54
  marker bandwidth search, 1-72
  marker parameters print, 1-61
  marker range, 1-54
  markers continuous, 1-54
discrete, 1-54
displayed, 1-47
  markers coupled, 1-54
  marker search left, 1-65
maximum, 1-65
minimum, 1-65
off, 1-65
right, 1-65
target, 1-65
  tracking, 1-70
  markers off, 1-55
  marker statistics, 1-55
  markers uncoupled, 1-55
  marker to CW frequency, 1-54
  marker to limit offset, 1-52
  marker to middle segment, 1-55
  marker to stimulus segment, 1-55
  marker width, 1-72
  marker zero, 1-55
MARKFAUV[D], 1-54
MARKFSTI[D], 1-54
MARKFVAL[D], 1-55
MARK&<$>[D], 1-54
MARKMIDD, 1-55
MARKMINI, 1-55
MARKOFF, 1-55
MARKREF, 1-55
MARKSPAN, 1-55
MARKSTAR, 1-55
MARKSTIM, 1-55
MARKSTOP, 1-55
MARKUNCO, 1-55
MARKZERO, 1-55
MAXF[D], 1-55
MEASA, 1-55
MEASB, 1-55
MEASR, 1-55
MEASTAT, 1-55
measurement calibration, 1-34
measurement restart, 1-63
memory channel 1
  color, 1-59
memory channel 2
  color, 1-59
MENU, 1-55
MENUVAV, 1-55
MENUCAL, 1-55
MENUCOPY, 1-55
MENUDISP, 1-55
MENUFORM, 1-55
MENUMARK, 1-55
MENUMEAS, 1-55
MENUMRF, 1-55
MENURECA, 1-55
MENUSCAL, 1-55
MENUSEQU, 1-55
MENUSIM, 1-55
MENUSYST, 1-56
middle value
  segment, 1-53
MINF[D], 1-56
MODI, 1-56
modify cal kit, 1-56
modify colors, 1-45
modify sequence, 1-56

N
  network analyzer mode, 1-51
NEWSEQ<i>, 1-56
new sequence, 1-56
NEXP, 1-56
next page, 1-56
NOOP, 1-56
number of readings, 1-56
NUMG[D], 1-56
NUMR[D], 1-56

O
  OC, 1-37
  OE, 1-37
  OFSD[D], 1-56
  OFSL[D], 1-56
  OFSZ[D], 1-56
  OI, 1-37
  OMII, 1-56
  OP, 1-36
  OPC, 1-3, 1-56
  OPC-compatible commands, 1-3
  open capacitance values, 1-43
  OPEP, 1-56
  operating parameters, 1-56
  operation complete, 1-3
  OS, 1-37
  OUTPCAL<i>, 1-56
  OUTPCALK, 1-57
  OUTPDATA, 1-57
  OUTPERRO, 1-57
  OUTPFORM, 1-57
  OUTPGCAL<i>, 1-57
  OUTPIDEN, 1-57
  OUTPIMCAL<i>, 1-57
  OUTPKAY, 1-57
  OUTPLEAS, 1-58
  OUTPLIMF, 1-58
  OUTPLIML, 1-58
  OUTPLIMM, 1-58
  OUTPMARK, 1-58
  OUTPMEM, 1-58
  OUTPMSTA, 1-58
  OUTPMWID, 1-58
  OUTPMWIL, 1-58
  OUTPPLOT, 1-58
  OUTPPMCAL<i>, 1-58
  OUTPRIN, 1-58
  OUTPRRNALL, 1-58
  OUTPRAW<i>, 1-59
  OUTPRFR, 1-59
  OUTPSEQ<i>, 1-59
  OUTPSSTAT, 1-59
  OUTPTITL, 1-59
  output
    plot string, 1-58
    output queue, 1-4

P
  PaintJet, 1-62
  PARAIN[D], 1-59
  PARAL, 1-59
  parallel poll, 1-2
  parallel port configure, 1-59
  PARAOUT[D], 1-59
  pass control, 1-71
PAUS, 1-59
pause, 1-59
pause to select sequence, 1-62
PAx,y, 1-36
PCB[D], 1-59
PCOLDATA1, 1-59
PCOLDATA2, 1-59
PCOLGRAT, 1-59
PCOLMEMO1, 1-59
PCOLMEMO2, 1-59
PCOLTEXT, 1-59
PCOLWARN, 1-59
PD, 1-35
PDATA, 1-59
PENNDATA[D], 1-59
PENNGRAT[D], 1-59
PENNMARK[D], 1-59
PENNMEMO[D], 1-59
PENNTXT[D], 1-59
pen number
  data, 1-59
  graticule, 1-59
  markers, 1-59
  memory, 1-59
  text, 1-59
peripheral
  address, 1-42
PG, 1-37
PGRAT, 1-60
PHAQ[D], 1-60
PHAS, 1-60
phase, 1-60
phase offset, 1-60
PLOS, 1-60
PLOT, 1-60
plot data, 1-59
plot graticule, 1-60
plot markers, 1-60
plot memory, 1-60
plot quadrant, 1-52, 1-64
plot scale, 1-64
plot softkeys, 1-62
plot speed, 1-60
plot string
  output, 1-58
plotter
  address, 1-42
  auto feed, 1-60
  baud rate, 1-60
  disk, 1-60
  form feed, 1-60
  handshake, 1-60
plotter default setup, 1-47
plotter port
HP-IB, 1-60
parallel, 1-60
serial, 1-60
plotter type, 1-60
plot text, 1-62
PLTNDSHK, 1-60
PLTRDTRISK, 1-60
PLTRTHPIS, 1-60
PLTRTPARA, 1-60
PLTRPTSERI, 1-60
PLTRAUTF, 1-60
PLTRBAUD[D], 1-60
PLTRFORE, 1-60
PLTYPHGL, 1-60
PLTYPPLTR, 1-60
plus sign, 1-6
PMEM, 1-60
PMKR, 1-60
PMTRTIT, 1-60
POIN[D], 1-60
points
  specify, 1-60
POLA, 1-60
polar, 1-60
polar markers, 1-60
POLMLN, 1-60
POLIMLOG, 1-60
POLMR, 1-60
PORE, 1-60
PORT 1 attenuator, 1-42
PORT1[D], 1-60
PORT 2 attenuator, 1-42
PORT2[D], 1-60
PORTA[D], 1-60
PORTB[D], 1-60
port extensions, 1-60
PORTP, 1-60
port power coupling, 1-60
PORTR, 1-61
PORTT, 1-61
POW[E][D], 1-61
power level, 1-61
power loss range
  edit, 1-61
power loss table, 1-52
  edit, 1-61
power meter
  address, 1-42
power meter cal factor, 1-43
power meter calibration, 1-62
power meter into title string, 1-60
power meter type, 1-61
power ranges, 1-61
power slope, 1-66
power sweep, 1-61
power trip, 1-61
POWLFREQ(D), 1-61
POWLLIST, 1-61
POWLLOSS(D), 1-61
POWM, 1-61
POWS, 1-61
POWT, 1-61
PPO, 1-2
PRAN, 1-61
PRIC, 1-61
PRINALL, 1-61
PRINSEQ<1>, 1-61
PRINTALL, 1-61
print color, 1-61
printer
address, 1-42
auto feed, 1-62
baud rate, 1-62
form feed, 1-62
handshake, 1-61
printer default setup, 1-46
printer port
HP-IB, 1-61
parallel, 1-62
serial, 1-62
print monochrome, 1-61
print sequence, 1-61
print softkeys, 1-62
PRIS, 1-61
PRRHNDSHK, 1-61
PRNPRTRHPB, 1-61
PRNPRTPARA, 1-62
PRNPRTSERI, 1-62
PRNTAUTF, 1-62
PRNTBAUD[1], 1-62
PRTRFORF, 1-62
PRNTPDJ, 1-62
PRNTYPEP, 1-62
PRNTYPLJ, 1-62
PRNTYPJP, 1-62
PRNTYPTJ, 1-62
PRxy, 1-37
PSOFT, 1-62
PTEXT, 1-62
PTOS, 1-62
PU, 1-37
purge file, 1-62
PURG<1>, 1-62
PWMCEACS[D], 1-62
PWMCOFF[D], 1-62
PWMCONES[D], 1-62
PWRLoss, 1-62
PWRMCAL[D], 1-62
PWR, 1-62

Q
quasi 2-port cal, 1-45
query, 1-3
question mark
using, 1-3

R
RAID, 1-62
RAIJSOL, 1-62
RAIRESP, 1-63
raw data
include with disk files, 1-48
REAL, 1-63
RECA<1>, 1-63
recall colors, 1-63
recall register, 1-63
recall sequence, 1-54
RECAREG<1>, 1-63
RECO, 1-63
REFD, 1-63
reference line value, 1-63
reference position, 1-63
set to mkr, 1-55
REFL, 1-63
reflection, 1-44
REFOP, 1-63
REFP[D], 1-63
REFT, 1-63
REPV[D], 1-63
remote lockout, 1-2
RESC, 1-63
RESD, 1-63
reset color, 1-64
RESPDONE, 1-63
response cal done, 1-63
REST, 1-63
restart averaging, 1-42
restore display, 1-63
resume cal sequence, 1-63
REVI, 1-63
REVM, 1-63
REV, 1-63
RFGLTO, 1-64
RF < LO, 1-64
RF > LO, 1-64
RFILTO, 1-64
RIGL, 1-64
RIGU, 1-64
RL1, 1-2
RS, 1-37
RSCO, 1-64
RST, 1-64

Index-9
S
S11, 1-64
S12, 1-64
S21, 1-64
S22, 1-64
SADD, 1-64
SAV1, 1-64
SAV2, 1-64
SAVC, 1-64
save cal kit, 1-64
save colors, 1-68
save format, 1-64
SAVE<i>, 1-64
SAVEREG<i>, 1-64
save register, 1-64
save sequence, 1-68
SAVEUSEK, 1-64
SAVUASC1, 1-64
SAVUBINA, 1-64
SCAL[D], 1-64
scale
    auto, 1-42
SCAP, 1-64
SDEL, 1-64
SDON, 1-65
SEAL, 1-65
SEAMAX, 1-65
SEAMIN, 1-65
SEAOFF, 1-65
SEAR, 1-65
SEATARG[D], 1-65
second harmonic, 1-50
SEDI[D], 1-65
segment
    add, 1-64
    delete, 1-64
    edit, 1-65
segment edit done, 1-48
segment select, 1-67
select sequence, 1-65
select standard, 1-67
semicolon, 1-6
sensor input selection, 1-71
SEQ<i>, 1-65
sequence wait, 1-65
SEQWAIT[D], 1-65
serial poll, 1-2
set bandwidth, 1-50
SETBIT[D], 1-65
SETDATE[$], 1-65
SETP, 1-65
SETTIME[$], 1-65
SETZ[D], 1-65
SH1, 1-2
SHOM, 1-65
show menus, 1-65
SSH,w, 1-37
SING, 1-65
single point type, 1-53
SL, 1-37
SLID, 1-66
sliding load, 1-66
    done, 1-66
    set, 1-66
SLIL, 1-66
SLIS, 1-66
SLOPE[D], 1-66
sloping line type, 1-53
SMIC, 1-66
SMIMGB, 1-66
SMIMLIN, 1-66
SMIMLOG, 1-66
SMIMRI, 1-66
SMIMRX, 1-66
Smith chart, 1-66
Smith markers, 1-66
SMOOAPER[D], 1-66
SMOOO, 1-66
smoothing, 1-66
smoothing aperture, 1-66
SOF, 1-66
SOFT<i>, 1-66
SOUP, 1-66
source handshake, 1-2
source power on/off, 1-66
spaces, 1-6
    in commands, 1-5
SPAN[D], 1-66
S-parameters, 1-64
SPECFWDM[I], 1-66
SPECFWDT[I], 1-66
specify class, 1-65
specify gate menu, 1-66
specify points, 1-60
SPECRES[I], 1-66
SPECRESP[I], 1-66
SPECREV[M[I], 1-67
SPECREV[T[I], 1-67
SPECSS11A[I], 1-67
SPECSS11B[I], 1-67
SPECSS11C[I], 1-67
SPECSS22A[I], 1-67
SPECSS22B[I], 1-67
SPECSS22C[I], 1-67
SPECTLFT[I], 1-67
SPECTLRFM[I], 1-67
SPECTLM[I], 1-67
SPECTLRLT[I], 1-67
SPECTRFM[I], 1-67
SPECTRRM[I], 1-67
SPECTTTFM[I], 1-67

Index-10
SPECTTFT[I], 1-67
SPECTTRM[I], 1-67
SPECTTRT[I], 1-67
SPEG, 1-66
SPLD, 1-67
split display, 1-67
SPn, 1-37
SR, 1-37
SR1, 1-2
SSEG[D], 1-67
STANA, 1-67
STANB, 1-67
STANC, 1-67
STAND, 1-67
standard defined, 1-68
standard definition, 1-46
standard labelling, 1-52
standard offsets, 1-56
standard type, 1-68
STANE, 1-67
STANF, 1-67
STANG, 1-67
STAR[D], 1-67
statistics
  marker, 1-55
STIP?, 1-67
STD, 1-68
STDARBI, 1-68
STDDEL, 1-68
STDLOAD, 1-68
STDPEN, 1-68
STDSHOR, 1-68
step down, 1-48
step up, 1-71
stimulus value
  segment, 1-53
STOP[D], 1-68
storage
  disk, 1-48, 1-51
  internal memory, 1-51
store to disk, 1-58
STOR<i>, 1-68
STORESEQ<i>, 1-68
STPSIZE[D], 1-68
structure
  of commands, 1-5
SVCO, 1-68
SWET[D], 1-68
SWR, 1-68
syntax types, 1-5, 1-6

T
T6, 1-2
TAKCS, 1-68
take cal sweep, 1-68
talker/listener, 1-69
TALKLIST, 1-69
TER[I], 1-69
terminal impedance, 1-69
terminators, 1-5
TESS?, 1-69
test port selection, 1-71
test set switching, 1-45
text
  color, 1-59
ThinkJet, 1-62
third harmonic, 1-50
TIMDTRAN, 1-69
time, 1-65
time domain bandpass, 1-42
time domain gate, 1-49
time specify, 1-68
TIMESTAMP, 1-69
time stamp, 1-69
TINT, 1-69
TITFO, 1-69
TITF<i>, 1-69
TITL[i], 1-69
title
  CRT, 1-69
title disk file, 1-69
title register, 1-69
title sequence, 1-69
title string to trace memory, 1-69
title to peripheral, 1-69
title to printer, 1-69
TITP, 1-69
TITREG<i>, 1-69
TITR<i>, 1-69
TITSEQ<i>[i], 1-69
TITTMEM, 1-69
TITTPERI, 1-69
TITTPRIN, 1-69
TRACK, 1-70
TRAD, 1-70
TRAN, 1-70
transform, 1-69
TRAOP, 1-70
TRIG, 1-70
trigger
  continuous, 1-45
  external, 1-48
  hold, 1-50
  number of groups, 1-56
  single, 1-65
  tri-state drivers, 1-2

Index-11
TST?, 1-70
TSTOFWD[D], 1-70
TSTOREV[D], 1-71
TSTP, 1-71
TTLHPULS, 1-71
TTLFPULS, 1-71
TTLOH, 1-71
TTLOL, 1-71
TTL out high, 1-71
TTL out low, 1-71
tuned receiver mode, 1-51

U
UCONV, 1-71
units, 1-4
UP, 1-71
up converter, 1-71
upper case, 1-5, 1-6
upper limit
  segment, 1-53
USEPASC, 1-71
user-defined cal kits, 1-43
user-defined kit
  save, 1-64
user graphics
  include with disk files, 1-48
USESENSA, 1-71
USESENSB, 1-71
use sensor A, 1-71
use sensor B, 1-71

V
velocity factor, 1-71
VELOFACT[D], 1-71
VIEM, 1-71
view measurement, 1-71
VOFF[D], 1-71

W
WAIT, 1-71
warning
  color, 1-59
warning beeper, 1-42
WAVE, 1-72
WIDT, 1-72
WIDV[D], 1-72
WINDMAXI, 1-72
WINDMINI, 1-72
WINDNORM, 1-72
window
  maximum, 1-72
  minimum, 1-72
  normal, 1-72
  shape, 1-72
  value, 1-72
WINDOW[D], 1-72
WINDUSEM, 1-72
WRSK<1>[8], 1-72

X
Xon, 1-61

Z
Z0, 1-65
## Contents

2. HP-IB Programming Reference
   Where to Look for More Information ........................................ 2-2
   HP-IB Operation ................................................................. 2-2
      Device Types ................................................................. 2-2
      Talker .............................................................................. 2-2
      Listener ............................................................................ 2-3
      Controller ......................................................................... 2-3
   HP-IB Bus Structure ............................................................... 2-3
      Data Bus ............................................................................ 2-4
      Handshake Lines ................................................................. 2-4
      Control Lines ........................................................................ 2-4
   HP-IB Requirements .................................................................. 2-5
   HP-IB Operational Capabilities .................................................. 2-5
      HP-IB Status Indicators ....................................................... 2-6
   Bus Device Modes .................................................................... 2-6
      System-Controller Mode ....................................................... 2-7
      Talker/Listener Mode ........................................................... 2-7
      Pass-Control Mode ............................................................... 2-7
      Analyzer Bus Modes ............................................................ 2-7
   Setting HP-IB Addresses .......................................................... 2-8
   Response to HP-IB Meta-Messages (IEEE-488 Universal Commands) 2-8
      Abort .................................................................................. 2-8
      Device Clear ................................................................. 2-8
      Local .................................................................................. 2-8
      Local Lockout ................................................................. 2-9
      Parallel Poll ................................................................. 2-9
      Pass Control ................................................................. 2-9
      Remote .............................................................................. 2-9
      Serial Poll ................................................................. 2-9
      Trigger .............................................................................. 2-9
   Analyzer Command Syntax ...................................................... 2-10
      Code Naming Convention .................................................... 2-10
      Valid Characters ................................................................... 2-11
      Units and Terminators ........................................................ 2-11
      Command Formats ............................................................. 2-11
         General Structure: ......................................................... 2-11
         Syntax Types .................................................................... 2-12
   Analyzer Operation ................................................................. 2-13
      Held Commands ................................................................. 2-13
      Operation Complete .......................................................... 2-13
   Reading Analyzer Data ........................................................... 2-14
      Output Queue ....................................................................... 2-14
      Command Interrogate ....................................................... 2-14
      Output Syntax ..................................................................... 2-15
      Marker data ........................................................................ 2-16
      Array-Data Formats ........................................................... 2-18
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace-Data Transfers</td>
<td>2-19</td>
</tr>
<tr>
<td>Frequency-Related Arrays</td>
<td>2-20</td>
</tr>
<tr>
<td>Data-Processing Chain</td>
<td>2-21</td>
</tr>
<tr>
<td>Data Arrays</td>
<td>2-21</td>
</tr>
<tr>
<td>Data Levels</td>
<td>2-22</td>
</tr>
<tr>
<td>Learn String and Calibration-Kit String</td>
<td>2-23</td>
</tr>
<tr>
<td>Error Reporting</td>
<td>2-24</td>
</tr>
<tr>
<td>Status Reporting</td>
<td>2-24</td>
</tr>
<tr>
<td>The Status Byte</td>
<td>2-26</td>
</tr>
<tr>
<td>The Event-Status Register and Event-Status-Register B</td>
<td>2-27</td>
</tr>
<tr>
<td>Error Output</td>
<td>2-27</td>
</tr>
<tr>
<td>Measurement Process</td>
<td>2-28</td>
</tr>
<tr>
<td>Step 1. Setting Up the Instrument</td>
<td>2-28</td>
</tr>
<tr>
<td>Step 2. Calibrating the Test Setup</td>
<td>2-28</td>
</tr>
<tr>
<td>Step 3. Connecting the Device under Test</td>
<td>2-28</td>
</tr>
<tr>
<td>Step 4. Taking the Measurement Data</td>
<td>2-29</td>
</tr>
<tr>
<td>Step 5. Post-Processing the Measurement Data</td>
<td>2-29</td>
</tr>
<tr>
<td>Step 6. Transferring the Measurement Data</td>
<td>2-29</td>
</tr>
<tr>
<td>Programming Examples</td>
<td>2-30</td>
</tr>
<tr>
<td>Program Information</td>
<td>2-31</td>
</tr>
<tr>
<td>Analyzer Features Helpful in Developing Programming Routines</td>
<td>2-31</td>
</tr>
<tr>
<td>Analyzer-Debug Mode</td>
<td>2-31</td>
</tr>
<tr>
<td>User-Controllable Sweep</td>
<td>2-31</td>
</tr>
<tr>
<td>Example 1: Measurement Setup</td>
<td>2-32</td>
</tr>
<tr>
<td>Example 1A: Setting Parameters</td>
<td>2-32</td>
</tr>
<tr>
<td>Tutorial Program Explanation</td>
<td>2-32</td>
</tr>
<tr>
<td>Initializing the System</td>
<td>2-32</td>
</tr>
<tr>
<td>Main Program</td>
<td>2-33</td>
</tr>
<tr>
<td>Running the Program</td>
<td>2-34</td>
</tr>
<tr>
<td>Example 1B: Verifying Parameters</td>
<td>2-35</td>
</tr>
<tr>
<td>Tutorial Program Explanation</td>
<td>2-35</td>
</tr>
<tr>
<td>Initializing the System</td>
<td>2-35</td>
</tr>
<tr>
<td>Main Program</td>
<td>2-36</td>
</tr>
<tr>
<td>Running the Program</td>
<td>2-36</td>
</tr>
<tr>
<td>Example 2: Measurement Calibration</td>
<td>2-37</td>
</tr>
<tr>
<td>Calibration kits</td>
<td>2-37</td>
</tr>
<tr>
<td>Example 2A: S&lt;sub&gt;11&lt;/sub&gt; 1-Port Measurement Calibration</td>
<td>2-37</td>
</tr>
<tr>
<td>Tutorial Program Explanation</td>
<td>2-38</td>
</tr>
<tr>
<td>Initializing the System</td>
<td>2-38</td>
</tr>
<tr>
<td>Main Program</td>
<td>2-38</td>
</tr>
<tr>
<td>Main Program Resumes</td>
<td>2-39</td>
</tr>
<tr>
<td>Running the Program</td>
<td>2-41</td>
</tr>
<tr>
<td>Example 2B: Full 2-Port Measurement Calibration</td>
<td>2-41</td>
</tr>
<tr>
<td>Tutorial Program Explanation</td>
<td>2-41</td>
</tr>
<tr>
<td>Initializing the System</td>
<td>2-41</td>
</tr>
<tr>
<td>Reflection Portion of the Full 2-Port Measurement Calibration Program</td>
<td>2-42</td>
</tr>
<tr>
<td>Reflection Portion of the Full 2-Port Measurement Calibration Program</td>
<td>2-42</td>
</tr>
<tr>
<td>Resumes</td>
<td>2-42</td>
</tr>
<tr>
<td>Transmission Portion of the Full 2-Port Measurement Calibration Program</td>
<td>2-44</td>
</tr>
<tr>
<td>Resumes</td>
<td>2-44</td>
</tr>
<tr>
<td>Isolation Portion of the Full 2-Port Measurement Calibration Program Resumes</td>
<td>2-45</td>
</tr>
<tr>
<td>Running the Program</td>
<td>2-47</td>
</tr>
<tr>
<td>Example 3: Measurement Data Transfer</td>
<td>2-48</td>
</tr>
<tr>
<td>Trace-Data Formats and Transfers</td>
<td>2-48</td>
</tr>
<tr>
<td>Example 3A: Data Transfer Using Markers</td>
<td>2-49</td>
</tr>
</tbody>
</table>
Initializing the System ............................................. 2-80
Main Program .................................................. 2-81
Running the Program ............................................. 2-82
Example 6: Limit-Line Testing ................................. 2-83
Using List-Frequency Mode .................................. 2-83
Example 6A: Setting Up a List-Frequency Sweep ......... 2-83
Tutorial Program Explanation ................................ 2-84
Initializing the System ......................................... 2-84
Main Program .................................................. 2-84
Main Program Resumes .......................................... 2-85
Running the Program ............................................. 2-86
Example 6B: Selecting a Single Segment from a Table of Segments 2-86
Tutorial Program Explanation ................................ 2-87
Initializing the System ......................................... 2-87
Running the Program ............................................. 2-89
Example 6C: Setting Up Limit Lines ......................... 2-89
Tutorial Program Explanation ................................ 2-90
Initializing the System ......................................... 2-90
Main Program .................................................. 2-90
Main Program Resumes .......................................... 2-92
Running the Program ............................................. 2-92
Example 6D: Performing PASS/FAIL Tests While Tuning . 2-93
Tutorial Program Explanation ................................ 2-94
Initializing the System ......................................... 2-94
Main Program .................................................. 2-94
Running the Program ............................................. 2-95
Example 7: Report Generation ................................. 2-96
Example 7A1: Operation Using Talker/Listener Mode .... 2-96
Tutorial Program Explanation ................................ 2-96
Initializing the System ......................................... 2-97
Main Program .................................................. 2-97
Running the Program ............................................. 2-98
Example 7A2: Controlling Peripherals Using Pass-Control Mode 2-98
Tutorial Program Explanation ................................ 2-99
Initializing the System ......................................... 2-99
Main Program .................................................. 2-99
Running the Program ............................................. 2-101
Example 7A3: Printing with the Serial Port ............... 2-101
Tutorial Program Explanation ................................ 2-102
Initializing the System ......................................... 2-102
Main Program .................................................. 2-102
Running the Program ............................................. 2-103
Example 7B1: Plotting to a File and Transferring the File Data to a Plotter . 2-103
Tutorial Program Explanation ................................ 2-104
Initializing the System ......................................... 2-104
Main Program .................................................. 2-104
Running the Program ............................................. 2-105
Utilizing PC-Graphics Applications Using the Plot File 2-105
LIF2DOS ................................................................ 2-106
Example 7B2: Reading Plot Files From a Disk ............. 2-106
Running the Program ............................................. 2-107
Example 7C: Reading ASCII Disk Files to the Instrument Controller's Disk File (HP 8753D standard only) . 2-108
Tutorial Program Explanation ................................ 2-110
Initializing the System ......................................... 2-110
Figures

2-1. HP-IB Bus Structure ................................................. 2-3
2-2. Analyzer Single Bus Concept ..................................... 2-6
2-3. The Data-Processing Chain ...................................... 2-21
2-4. Status Reporting Structure ....................................... 2-24
2-5. Status Reporting Structure ....................................... 2-62

Tables

2-1. Code Naming Convention .......................................... 2-10
2-2. FORM 4 (ASCII) Data-Transfer Character Definitions ........... 2-15
2-3. Units as a Function of Display Format ............................ 2-17
2-4. HP 8753D/Option 011 Network Analyzer Array-Data Formats .... 2-19
2-5. Status Bit Definitions ............................................. 2-25
HP-IB Programming Reference

This chapter includes information on the following topics:

- HP-IB operation
- HP-IB requirements
- analyzer command syntax
- analyzer operation
- reading analyzer data
- data-processing chain
- error reporting
- measurement process
- programming examples
  - measurement setup
    - setting parameters
    - verifying parameters
  - measurement calibration
    - $S_{11}$ 1-port measurement calibration
    - full 2-port measurement calibration
  - data transfer
    - data transfer using markers
    - ASCII data transfer using FORM 4
    - data transfer using floating-point numbers
    - data transfer using frequency array information
    - internal-binary data transfer using FORM 1
  - measurement-process synchronization
    - using the error queue
    - generating interrupts
    - power meter calibration
  - network analyzer system setups
    - using the learn string
    - reading calibration data
    - saving and restoring the instrument state
  - limit-line testing
    - setting up a list-frequency sweep
    - selecting a single segment from a table of segments
    - setting up limit lines
    - performing PASS/FAIL tests while tuning
  - report generation
    - operation using the talker/listener mode
    - controlling peripherals using pass-control mode
    - printing with the serial port
    - plotting to a file and transferring file data to a plotter
    - utilizing PC-graphics applications using the plot file
    - reading ASCII disk files to the instrument controller disk file
Where to Look for More Information

Additional information covering many of the topics discussed in this chapter is located in the following:

- Chapter 1, “HP-IB Command Reference,” includes both functional and alphabetical lists of all analyzer HP-IB commands.
- *Tutorial Description of the Hewlett-Packard Interface Bus,* presents a description and discussion of all aspects of the HP-IB. A thorough overview of all technical details as a broad tutorial. HP publication, HP part number 5021-1927.
- Chapter 3, “HP BASIC Programming Reference,” includes programming examples in HP BASIC.

HP-IB Operation

The Hewlett-Packard Interface Bus (HP-IB) is Hewlett-Packard’s hardware, software, documentation, and support for IEEE 488.1 and IEC-625 worldwide standards for interfacing instruments. This interface allows you to operate the analyzer and peripherals in two methods:

- by an external system controller
- by the network analyzer in system-controller mode

Device Types

The HP-IB employs a party-line bus structure in which up to 15 devices can be connected on one contiguous bus. The interface consists of 16 signal lines and 8 ground lines within a shielded cable. With this cabling system, many different types of devices including instruments, computers, power meters, plotters, printers, and disk drives can be connected in parallel.

Every HP-IB device must be capable of performing one or more of the following interface functions:

**Talker**

A talker is a device capable of transmitting device-dependent data when addressed to talk. There can be only one active talker at any given time. Examples of this type of device include:

- power meters
- disk drives
- voltmeters
- counters
- tape readers

The network analyzer is a talker when it sends trace data or marker information over the bus.
Listener

A listener is a device capable of receiving device-dependent data over the interface when addressed to listen. There can be as many as 14 listeners connected to the interface at any given time. Examples of this type of device include:

- printers
- power supplies
- signal generators

The network analyzer is a listener when it is controlled over the bus by a system controller.

Controller

A controller is defined as a device capable of:

1. managing the operation of the bus
2. addressing talkers and listeners

There can be only one active controller on the interface at any time. Examples of controllers include desktop computers, minicomputers, workstations, and the network analyzer. In a multiple-controller system, active control can be passed between controllers, but there can only be one system controller connected to the interface. The system controller acts as the master and can regain active control at any time. The analyzer is an active controller when it plots, prints, or stores to an external disk drive in the pass-control mode. The analyzer is also a system controller when it is operating in the system-controller mode.

HP-IB Bus Structure

![HP-IB Bus Structure Diagram]

Figure 2-1. HP-IB Bus Structure
Data Bus

The data bus consists of 8 bi-directional lines that are used to transfer data from one device to another. Programming commands and data transmitted on these lines are typically encoded in ASCII, although binary encoding is often used to speed up the transfer of large arrays. Both ASCII- and binary-data formats are available to the analyzer. In addition, every byte transferred over HP-IB undergoes a handshake to insure valid data.

Handshake Lines

A three-line handshake scheme coordinates the transfer of data between talkers and listeners. To insure data integrity in multiple-listener transfers, this technique forces data transfers to occur at the transfer rate of the slowest device connected to the interface. With most computing controllers and instruments, the handshake is performed automatically, making it transparent to the programmer.

Control Lines

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

IFC (Interface Clear)

This line is used exclusively by the system controller. When this line is true (low), all devices (whether addressed or not) unaddress and revert to an idle state.

ATN (Attention)

The active controller uses this line to define whether the information on the data bus is command-oriented or data-oriented. 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.

SRQ (Service Request)

This line is set true (low) when a device requests service and the active controller services the requesting device. The network analyzer can be enabled to pull the SRQ line for a variety of reasons such as requesting control of the interface, for the purposes of printing, plotting, or accessing a disk.

REN (Remote Enable)

This line is used exclusively by the system controller. When this line is set true (low), the bus is in the remote mode, and devices are addressed by the controller to either listen or talk. When the bus is in remote and a device is addressed, it receives instructions from the system controller via HP-IB rather than from its front panel (pressing [LOCAL] returns the device to front-panel operation). When this line is set false (high), the bus and all of the connected devices return to local operation.

EOI (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.
**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 a synchronous data transfer using a 3-line handshake system.
- **Data Rate:** Maximum of 1 megabyte-per-second over the specified distances with tri-state drivers. Actual data rate depends on the transfer rate of the slowest device connected to the bus.
- **Address Capability:** Primary addresses: 31 talk, 31 listen. A maximum of 1 talker and 14 listeners can be connected to the interface at given time.
- **Multiple-Controller Capability:** In systems with more than one controller (like the analyzer system), only one controller can be active at any given 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.

**HP-IB Operational Capabilities**

On the network analyzer’s rear panel, next to the HP-IB connector, there is a list of HP-IB device subsets as defined by the IEEE 488.2 standard. The analyzer has the following capabilities:

- **SH1** Full-source handshake.
- **AH1** Full-acceptor handshake.
- **T6** Basic talker, answers serial poll, unaddresses if MLA is issued. No talk-only mode.
- **L4** Basic listener, unaddresses if MTA is issued. No listen-only mode.
- **SR1** Complete service request (SRQ) capabilities.
- **RL1** Complete remote/local capability including local lockout.
- **PP0** Does not respond to parallel poll.
- **DC1** Complete device clear.
- **DT1** Responds to a Group Execute Trigger (GET) in the hold-trigger mode.
- **C1,C2,C3** System controller capabilities in system-controller mode.
- **C10** Pass control capabilities in pass-control mode.
- **E2** Tri-state drivers.
- **LE0** No extended listener capabilities.
- **TE0** No extended talker capabilities.
**HP-IB Status Indicators**

When the analyzer is connected to other instruments over the HP-IB, the HP-IB status indicators illuminate to display the current status of the analyzer. The HP-IB status indicators are located in the instrument-state function block on the front panel of the network analyzer.

- **R** = Remote Operation
- **L** = Listen mode
- **T** = Talk mode
- **S** = Service request (SRQ) asserted by the analyzer

**Bus Device Modes**

The analyzer uses a single-bus architecture. The single bus allows both the analyzer and the host controller to have complete access to the peripherals in the system.

Three different controller modes are possible in and HP-IB system:

- system-controller mode
- talker/listener mode
- pass-control mode

---

![Diagram](image)

**Figure 2-2. Analyzer Single Bus Concept**
System-Controller Mode

This mode allows the analyzer to control peripherals directly in a stand-alone environment (without an external controller). This mode can only be selected manually from the analyzer’s front panel. It can only be used if no active computer or instrument controller is connected to the system via HP-IB. If an attempt is made to set the network analyzer to the system-controller mode when another controller is connected to the interface, the following message is displayed on the analyzer’s display screen:

"CAUTION: ANOTHER SYSTEM CONTROLLER ON HP-IB"

The analyzer must be set to the system-controller mode in order to access peripherals from the front panel. In this mode, the analyzer can directly control peripherals (plotters, printers, disk drives, power meters, etc.) and the analyzer may plot, print, store on disk or perform power meter functions.

**Note**

Do not attempt to use this mode for programming. HP recommends using an external instrument controller when programming. See the following section, “Talker/Listener Mode.”

Talker/Listener Mode

This is the mode that is normally used for remote programming of the analyzer. In talker/listener mode, the analyzer and all peripheral devices are controlled from an external instrument controller. The controller can command the analyzer to talk and other devices to listen. The analyzer and peripheral devices cannot talk directly to each other unless the computer sets up a data path between them. This mode allows the analyzer to act as either a talker or a listener, as required by the controlling computer for the particular operation in progress.

Pass-Control Mode

This mode allows the computer to control the analyzer via HP-IB (as with the talker/listener mode), but also allows the analyzer to take control of the interface in order to plot, print, or access a disk. During an analyzer-controlled peripheral operation, the host computer is free to perform other internal tasks (i.e. data or display manipulation) while the analyzer is controlling the bus. After the analyzer-controlled task is completed, the analyzer returns control to the system controller.

**Note**

Performing an instrument preset does not affect the selected bus mode, although the bus mode will return to talker/listener mode if the line power is cycled.

**Note**

“Specifications and Measurement Uncertainties” in the *HP 8753D/Option 011 Network Analyzer User’s Guide* provides information on setting the correct bus mode from the front-panel menu.

Analyzer Bus Modes

As discussed earlier, under HP-IB control, the analyzer can operate in one of three modes: talker/listener, pass-control, or system-controller mode.

In talker/listener mode, the analyzer behaves as a simple device on the bus. While in this mode, the analyzer can make a plot or print using the OUTPLOT; or OUTPRIN; commands. The analyzer will wait until it is addresses to talk by the system controller and then dump the display to a plotter/printer that the system controller has addressed to listen. Use of the commands PLOT; and PRINALL; require control to be passed to another controller.
In pass-control mode, the analyzer can request control from the system controller and take control of the bus if the controller addresses it to take control. This allows the analyzer to take control of printers, plotters, and disk drives on an as-needed basis. The analyzer sets event-status-register bit 1 when it needs control of the interface, and the analyzer will transfer control back to the system controller at the completion of the operation. It will pass control back to its controller address, specified by ADDRCONT.

The analyzer can also operate in the system-controller mode. This mode is only used when there is no remote controller on the bus. In this mode, the analyzer takes control of the bus, and uses it whenever it needs to access a peripheral. While the analyzer is in this mode, no other devices on the bus can attempt to take control. Specifically, the REN, ATN, and IFC lines must remain unasserted, and the data lines must be freed by all but the addressed talker.

**Setting HP-IB Addresses**

In systems interfaced using HP-IB, each instrument on the bus is identified by an HP-IB address. This address code must be different for each instrument on the bus. These addresses are stored in short-term, non-volatile memory and are not affected when you press **Preset** or cycle the power. The analyzer occupies two HP-IB addresses: the instrument itself and the display. The display address is derived from the instrument address by complementing the instrument’s least-significant bit. Hence, if the instrument is at an even address, the display occupies the next higher address. If the instrument is at an odd address, the display occupies the next lower address. A change in the next instrument preset following the address change.

The analyzer addresses are set by pressing **Local** **Set Addresses**. In system-controller mode, the addresses must be set for the plotter, printer, disk drive, and power meter.

The default address for the analyzer is device 16, and the display address is device 17.

| Note | There is also an address for the system controller. This address refers to the controller when the network analyzer is being used in pass-control mode. This is the address that control is passed back to when the analyzer-controlled operation is complete. |

**Response to HP-IB Meta-Messages (IEEE-488 Universal Commands)**

**Abort**

The analyzer responds to the abort message (IFC) by halting all listener, talker, and controller functions.

**Device Clear**

The analyzer responds to the device clear commands (DCL, SDC) by clearing the input and output queues, and clearing any HP-IB errors. The status registers and the error queue are unaffected.

**Local**

The analyzer will go into local mode if: the local command (GTL) is received, the remote line is unasserted, or the front-panel local key is pressed. Changing the analyzer’s HP-IB status from remote to local does not affect any of the front-panel functions or values.
Local Lockout

If the analyzer receives the local-lockout command (LLO) while it is in remote mode, it will disable the entire front panel except for the line power switch. A local-lockout condition can only be cleared by releasing the remote line, although the local command (GTL) will place the instrument temporarily in local mode.

Parallel Poll

The analyzer does not respond to parallel-poll configure (PPC) or parallel-poll unconfigure (PPU) messages.

Pass Control

If the analyzer is in pass-control mode, is addressed to talk, and receives the take-control command (TCT), from the system control it will take active control of the bus. If the analyzer is not requesting control, it will immediately pass control to the system controller's address.

Otherwise, the analyzer will execute the function for which it sought control of the bus and then pass control back to the system controller.

Remote

The analyzer will go into remote mode when the remote line is asserted and the analyzer is addressed to listen. While the analyzer is held in remote mode, all front-panel keys (with the exception of LOCAL) are disabled. Changing the analyzer's HP-IB status from remote to local does not affect any front-panel settings or values.

Serial Poll

The analyzer will respond to a serial poll with its status byte, as defined in the "Status Reporting" section of this chapter. To initiate the serial-poll sequence, address the analyzer to talk and issue a serial-poll enable command (SPE). Upon receiving this command, the analyzer will return its status byte. End the sequence by issuing a serial-poll disable command (SPD). A serial poll does not affect the value of the status byte, and it does not set the instrument to remote mode.

Trigger

In hold mode, the analyzer responds to device trigger by taking a single sweep. If a one-path, 2-port measurement calibration is active, the analyzer will set the waiting-for-Group-Execute-Trigger bits in the status byte. If waiting-for-forward-GET is set, the analyzer will assume the device is connected for forward measurement and take a sweep when GET is received. Similarly, if the waiting-for-reverse-GET bit is set, the analyzer will assume the device is connected for reverse measurement. The analyzer responds only to selected-device trigger (SDT). This means that it will not respond to group execute-trigger (GET) unless it is addressed to listen. The analyzer will not respond to GET if it is not in hold mode.
Analyzer Command Syntax

Code Naming Convention

The analyzer HP-IB commands are derived from their front-panel key titles (where possible), according to the naming convention described in this section.

Simple commands are the first four letters of the function they control, as in POWE, the command name for power. If the function label contains two words, the first three mnemonic letters are the first three letters of the first word, and the fourth mnemonic letter is the first letter of the second word. For example, ELED is derived from electrical delay.

If there are many commands grouped together in a class, as in markers or plotting pen numbers, the command is increased to 8 letters. The first 4 letters are the class label derived using rule 1. The last 4 letters are the function specifier, again derived using rule 1. As an example, class pen numbers are represented by the command PENN, which is used in combination with several functions such as PENNDATA, PENNMEM.

The code naming guidelines, listed in Table 2-1, are used in order to:

- make commands more meaningful and easier to remember
- maintain compatibility with other products

Note: There are times when these guidelines are not followed due to technical considerations.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Key Title</th>
<th>For HP-IB Code Use</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>One Word</td>
<td>Power Start</td>
<td>First Four Letters</td>
<td>POWE</td>
</tr>
<tr>
<td>Two Words</td>
<td>Electrical Delay</td>
<td>First Three Letters of First Word, First Letter of Second Word</td>
<td>ELED</td>
</tr>
<tr>
<td></td>
<td>Search Right</td>
<td></td>
<td>SEAR</td>
</tr>
<tr>
<td>Two Words in a Group</td>
<td>Marker — Center</td>
<td>Four Letters of Both</td>
<td>MARKCENT</td>
</tr>
<tr>
<td></td>
<td>Gate — Span</td>
<td></td>
<td>GATESPAN</td>
</tr>
<tr>
<td>Three Words</td>
<td>Cal Kit N 500</td>
<td>First Three Letters of First Word, First Letter of Second Word, First Four Letters of Third Word</td>
<td>CALKN50</td>
</tr>
<tr>
<td></td>
<td>Pen Num Data</td>
<td></td>
<td>PENNDATA</td>
</tr>
</tbody>
</table>

Some codes require appendages (ON, OFF, 1, 2, etc.). Codes that do not have a front-panel equivalent are HP-IB only commands. They use a similar convention based on the common name of the function. Where possible, analyzer codes are compatible with HP 8510A/B codes.

Note: Front-panel-equivalent codes and HP-IB-only codes are summarized in "Command Reference" in this guide.
Valid Characters

The analyzer accepts ASCII letters, numbers, decimal points, +/-, semicolons (;), quotation marks ("), carriage returns (CR), and linefeeds (LF). Both upper- and lower-case letters are acceptable. Carriage returns, leading zeros, spaces, and unnecessary terminators are ignored, except for those within a command or appendage. An invalid character causes a syntax error. Syntax errors are described in more detail in “Error Messages” of the HP 8753D/Option 011 Network Analyzer User’s Guide.

Units and Terminators

The analyzer outputs data in basic units and assumes these basic units when it receives an input (unless the input is otherwise qualified). The basic units and allowable expressions can be found below. Both upper- and lower-case letters are acceptable.

<table>
<thead>
<tr>
<th>Terminator codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>S Seconds</td>
</tr>
<tr>
<td>MS Milliseconds</td>
</tr>
<tr>
<td>US Microseconds</td>
</tr>
<tr>
<td>NS Nanoseconds</td>
</tr>
<tr>
<td>FS Picoseconds</td>
</tr>
<tr>
<td>FS Femtoseconds</td>
</tr>
</tbody>
</table>

Terminators are used to indicate the end of a command. This allows the analyzer to recover to the next command in the event of a syntax error. The semicolon (;) is the recommended command terminator. The line-feed character (LF) and the HP-IB EOI line can also be used as terminators. Again, the analyzer will ignore the carriage-return character (CR).

Command Formats

The HP-IB commands accepted by the analyzer can be grouped into five input-syntax types. The analyzer does not distinguish between upper- and lower-case letters.

General Structure:

The general syntax structure is: [code][appendage][data][unit][terminator]

The individual sections of the syntax code are explained below.

[code] The root mnemonic, as found in “Command Reference.”

[appendage] A qualifier attached to the root mnemonic. Possible appendages are ON or OFF (toggle a function ON or OFF), or integers, which specify one option out of several. There can be no spaces or symbols between the code and the appendage.

[data] A single operand used by the root mnemonic, usually to set the value of a function. The data can be a number or a character string. Numbers are accepted as integers or decimals, with power of ten specified by E (for example, STAR 0.2E+10; sets the start frequency to 2 GHz). Character strings must be preceded and followed by double quotation marks.

[unit] The units of the operand, if applicable. If no units are specified, the analyzer assumes the basic units as described above. The data is entered into the function when either units or a terminator are received.
[terminator] Indicates the end of the command, enters the data, and switches the
active-entry area OFF. A semicolon (;) is the recommended terminator.

The analyzer also interprets line feed as a terminator. It is a advantageous to
carefully and methodically use semicolons as terminators. This practice insures
correct command interpretation and operation. Additional or extra terminators
are ignored.

Syntax Types
The specific syntax types are:

SYNTAX TYPE 1: [code] [terminator]

These are simple action commands that require no complementary information, such as
AUTO; (autoscales the active channel).

SYNTAX TYPE 2: [code][appendage][terminator]

These are simple action commands requiring limited customization, such as CORR0N; and
CORR0FF; (Switch error correction ON or OFF) or RECA1;, RECA2;, RECA3; (recall register 1,
2, 3). There can be no characters or symbols between the code and the appendage. In the
following cases: CLEAREG[D], RECAREG[D], SAVREG[D], and TITREG[D], [D] must be 2
characters. For example, CLEAREG01; will execute; CLEAREG1; will generate a syntax error.

SYNTAX TYPE 3: [code] [data] [unit] [terminator]

These are data-input commands such as STAR 1.0 GHZ; (set the start frequency to 1 GHz).

SYNTAX TYPE 4: [code] [appendage] [data] [terminator]

These are titling and marker commands that have an appendage, such as TITL “FILTER”
(enter FILTER as the CRT title), TITR1 “STATE1”, TITR2 “EMPTY” (title register 1 STATE1,
title register 2 EMPTY.)

INTERROGATE SYNTAX: [code][?]

To interrogate a front-panel-equivalent function, append a question mark (?) to the root
mnemonic, for example POWE?, AVERO?, or REAL?. To interrogate commands with integer
appendages, place the question mark after the appendage.
Analyzer Operation

Held Commands

The analyzer cannot process HP-IB commands while executing certain key commands known as "held" commands. Once a held command is received, the analyzer will read new commands into the input buffer, but it will not begin the execution of any commands until the completion of the held command. When the 15-character input buffer is full, the analyzer will put hold on the bus until it is able to process the commands in the buffer. In other words, it will not complete the handshake with the talker and hold off further communication until space becomes available in the input buffer. See the section titled "Handshake Lines" earlier in this chapter.

Operation Complete

Note

For a complete list of OPC-compatible commands, see the section titled "HP-IB Capabilities" at the beginning of Chapter 1.

Occasionally, there is a need to know when certain analyzer operations have been completed. There is an operation-complete function (OPC) that allows a synchronization of programs with the execution of certain key commands. This mechanism is activated by issuing OPC; or OPC?; prior to an OPC-compatible command. The status byte or ESR operation-complete bit will then be set after the execution of the OPC-compatible command. For example, issuing OPC; SING; causes the OPC bit to be set when the single sweep is finished. Issuing OPC?; in place of the OPC; causes the analyzer to output a one (1) when the command execution is complete. Addressing the analyzer to talk after issuing OPC?; will not cause an "addressed to talk without selecting output" error, but the analyzer will halt the computer by not transmitting the one (1) until the command has completed. For example, executing OPC?; PRES; and then immediately interrogating the analyzer causes the bus to halt until the instrument preset is complete and the analyzer outputs a one (1).

Note

Commands that call a calibration class are held if there is only one standard in the class, since these commands trigger a measurement.

As another example, consider the timing of sweep completion. Send the command string "SWET 3 S;OPC?;SING;" to the analyzer. This string sets the analyzer sweep time to 3 seconds, and then waits for completion of a single sweep to respond with a one (1). The computer is programmed to read the one (1) response from the analyzer as the completion of the single sweep. The program then waits until the sweep is completed before continuing operation. At this point a valid trace exists and the trace data can be read into the computer.
Reading Analyzer Data

Output Queue
Whenever a output-data command is received, the analyzer puts the data into the output queue (or buffer) where it is held until the system controller outputs the next read command. The queue, however, is only one event long: the next output-data command will overwrite the data already in the queue. Therefore, it is important to read the output queue immediately after every interrogation or data request from the analyzer.

Command Interrogate
All instrument functions can be interrogated to find the current ON/OFF state or value. For instrument state commands, append the question mark character (?) to the command to interrogate the state of the functions. Suppose the operator has changed the power level from the analyzer’s front panel. The computer can ascertain the new power level using the analyzer’s command-interrogate function. If a question mark is appended to the root of a command, the analyzer will output the value of that function. For instance, POWE 7 DB; sets the source power to 7 dB, and POWE?; outputs the current RF source power at the test port. When the analyzer receives POWE?; it prepares to transmit the current RF source power level. This condition illuminates the analyzer front-panel talk light (T). In this case, the analyzer transmits the output power level to the controller.

ON/OFF commands can be also be interrogated. The reply is a one (1) if the function is ON or a zero (0) if it is OFF. For example, if a command controls an active function that is underlined on the analyzer display, interrogating that command yields a one (1) if the command is underlined or a zero (0) if it is not. As another example, there are nine options on the format menu and only one option is underlined at a time. Only the underlined option will return a one when interrogated.

For instance, send the command string DUAC?; to the analyzer. If dual-channel display is switched ON, the analyzer will return a one (1) to the instrument controller.

Similarly, to determine if phase is being measured and displayed, send the command string PHAS?; to the analyzer. In this case, the analyzer will return a one (1) if phase is currently being displayed. Since the command only applies to the active channel, the response to the PHAS?; query depends on which channel is active.
Output Syntax

The following three types of data are transmitted by the analyzer in ASCII format:

- response to interrogation
- certain output commands
- ASCII floating-point (FORM 4) array transfers

Marker-output commands and interrogated commands are output in ASCII format only, meaning that each character and each digit is transmitted as a separate byte, leaving the receiving computer to reconstruct the numbers and strings. Numbers are transmitted as 24-character strings, consisting of:

- DDD.DDDDDDDDDDDDDDDDDDDDE-DD

When multiple numbers are sent, the numbers are separated by commas. See Table 2-2 for character definitions.

Table 2-2. FORM 4 (ASCII) Data-Transfer Character Definitions

<table>
<thead>
<tr>
<th>Character(s)*</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign</td>
<td>&quot;.&quot; for negative, blank for positive.</td>
</tr>
<tr>
<td>3 digits</td>
<td>Digits to the left of the decimal point.</td>
</tr>
<tr>
<td>Decimal point</td>
<td></td>
</tr>
<tr>
<td>15 digits</td>
<td>Digits to the right of the decimal point.</td>
</tr>
<tr>
<td>E</td>
<td>Exponent notation.</td>
</tr>
<tr>
<td>Sign</td>
<td>&quot;.&quot; for negative, &quot;+&quot; for positive.</td>
</tr>
<tr>
<td>Exponent</td>
<td>Two digits for the exponent.</td>
</tr>
</tbody>
</table>

* The items in this column are separated by commas and delimited (terminated) with a line feed character (LF).
Marker data

The network analyzer offers several options for outputting trace-related data. Trace information can be read out of the analyzer in several different formats. Data can be selectively read from the trace using the markers, or the entire trace can be read by the controller. If only specific information is required (such as a single point on the trace or the result of a marker search), the marker output command can be used to read the information.

To read the trace data using the marker, the marker must first be assigned to the desired frequency. This is accomplished using the marker commands. The controller sends a marker command followed by a frequency within the trace-data range. If the actual desired frequency was not sampled, the markers can be set to continuous mode and the desired marker value will be linearly interpolated from the two nearest points. This interpolation can be prevented by putting the markers into discrete mode. Discrete mode allows the marker to only be positioned on a measured trace-data point.

As an alternative, the analyzer can be programmed to choose the stimulus value by using the MARKER SEARCH function. Maximum, minimum, target value, or bandwidths search can be automatically determined with MARKER SEARCH. To continually update the search, switch the marker tracking ON. The trace-maximum search will remain activated until:

- The search is switched OFF.
- The tracking is switched OFF.
- All markers are switched OFF.

Marker data can be output to a controller with a command to the analyzer. This set of commands causes the analyzer to transmit three numbers: marker value 1, marker value 2, and marker stimulus value. In log-magnitude display mode we get the log magnitude at marker 1, zero, and the marker frequency. See Table 2-3 for a complete listing of all the possibilities for values 1 and 2. The three possibilities for the third parameter are:

- frequency
- time (as in time domain, option 010 only)
- CW time
Table 2-3. Units as a Function of Display Format

<table>
<thead>
<tr>
<th>Display Format</th>
<th>Marker Mode</th>
<th>OUTPMARK</th>
<th>OUTPFORM</th>
<th>MARKER READOUT*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>value 1</td>
<td>value 2</td>
<td>value 1</td>
</tr>
<tr>
<td>LOG MAG</td>
<td></td>
<td>dB</td>
<td>†</td>
<td>dB</td>
</tr>
<tr>
<td>PHASE</td>
<td></td>
<td>degrees</td>
<td>†</td>
<td>degrees</td>
</tr>
<tr>
<td>DELAY</td>
<td></td>
<td>seconds</td>
<td>†</td>
<td>seconds</td>
</tr>
<tr>
<td>SMITH CHART</td>
<td>LIN MKR</td>
<td>lin mag</td>
<td>degrees</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>LOG MKR</td>
<td>dB</td>
<td>degrees</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>Re/Im</td>
<td>real</td>
<td>imag</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>R + jX</td>
<td>real ohms</td>
<td>imag ohms</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>G + jB</td>
<td>real</td>
<td>imag</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>Siemens</td>
<td>Siemens</td>
<td>Siemens</td>
<td>Siemens</td>
</tr>
<tr>
<td>POLAR</td>
<td>LIN MKR</td>
<td>lin mag</td>
<td>degrees</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>LOG MKR</td>
<td>dB</td>
<td>degrees</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>Re/Im</td>
<td>real</td>
<td>imag</td>
<td>real</td>
</tr>
<tr>
<td>LIN MAG</td>
<td></td>
<td>lin mag</td>
<td>†</td>
<td>lin mag</td>
</tr>
<tr>
<td>REAL</td>
<td></td>
<td>real</td>
<td>†</td>
<td>real</td>
</tr>
<tr>
<td>SWR</td>
<td></td>
<td>SWR</td>
<td>†</td>
<td>SWR</td>
</tr>
</tbody>
</table>

*The marker readout values are the marker values displayed in the upper right-hand corner of the display. They also correspond to the value and auxiliary value associated with the fixed marker.

†Value 2 is not significant in this format, though it is included in data transfers.
Array-Data Formats

The analyzer can transmit and receive arrays in the analyzer’s internal binary format as well as four different numeric formats. The current format is set with the FORM1, FORM2, FORM3, FORM4, and FORM5 commands. These commands do not affect learn-string transfers, calibration-kit string transfers, or non-array transfers, such as command interrogate, or output marker values. A transmitted array will be output out in the current format, and the analyzer will attempt to read incoming arrays according to the current format. Each data point in an array is a pair of numbers, usually a real/imaginary pair. The number of data points in each array is the same as the number of points in the current sweep.

The five formats are described below:

FORM 1  The analyzer’s internal binary format, 6 bytes-per-data point. The array is preceded by a four-byte header. The first two bytes represent the string "#A", the standard block header. The second two bytes are an integer representing the number of bytes in the block to follow. FORM 1 is best applied when rapid data transfers, not to be modified by the computer nor interpreted by the user, are required.

FORM 2  IEEE 32-bit floating-point format, 8 bytes-per-data point. The data is preceded by the same header as in FORM1. Each number consists of a 1-bit sign, an 8-bit biased exponent, and a 23-bit mantissa. FORM 2 is the format of choice if your computer supports single-precision floating-point numbers.

FORM 3  IEEE 64-bit floating-point format, 16 bytes-per-data point. The data is preceded by the same header as in FORM 1. Each number consists of a 1-bit sign, an 11-bit biased exponent, and a 52-bit mantissa. This format may be used with double-precision floating-point numbers. No additional precision is available in the analyzer data, but FORM 3 may be a convenient form for transferring data to your computer.

FORM 4  ASCII floating-point format. The data is transmitted as ASCII numbers, as described in “Output Syntax”.

There is no header. The analyzer uses FORM 4 to transfer data that is not related to array transfers (i.e. marker responses and instrument settings).

FORM 5  PC-DOS 32-bit floating-point format with 4 bytes-per-number, 8 bytes-per-data point. The data is preceded by the same header as in FORM 1. The byte order is reversed to comply with PC-DOS formats. If you are using a PC-based controller, FORM 5 is the most effective format to use.

The analyzer terminates each transmission by asserting the EOI interface line with the last byte transmitted. Table 2-4 offers a comparative overview of the five array-data formats.
Table 2-4. HP 8753D/Option 011 Network Analyzer Array-Data Formats

<table>
<thead>
<tr>
<th>Format type</th>
<th>Type of Data</th>
<th>Bytes per Data Value</th>
<th>Bytes-per-point 2 data values</th>
<th>201 Bytes-per-trace</th>
<th>Total Bytes add header</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORM 1</td>
<td>Internal Binary</td>
<td>3</td>
<td>6</td>
<td>1206</td>
<td>1210</td>
</tr>
<tr>
<td>FORM 2</td>
<td>IEEE 32-bit</td>
<td>4</td>
<td>8</td>
<td>1608</td>
<td>1612</td>
</tr>
<tr>
<td></td>
<td>Floating-Point</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORM 3</td>
<td>IEEE 64-bit</td>
<td>8</td>
<td>16</td>
<td>3216</td>
<td>3220</td>
</tr>
<tr>
<td></td>
<td>Floating-Point</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORM 4</td>
<td>ASCII Numbers</td>
<td>24</td>
<td>50</td>
<td>10,050</td>
<td>10,050*</td>
</tr>
<tr>
<td>FORM 5</td>
<td>PC-DOS 82-bit</td>
<td>4</td>
<td>8</td>
<td>1608</td>
<td>1612</td>
</tr>
<tr>
<td></td>
<td>Floating-Point</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*No header is used in FORM 4.

Trace-Data Transfers

Transferring trace data from the analyzer using an instrument controller can be divided into three steps:

1. allocating an array to receive and store the data
2. commanding the analyzer to transmit the data
3. accepting the transferred data

Data residing in the analyzer is always stored in pairs for each data point (to accommodate real/imaginary pairs). Hence, the receiving array has to be two elements wide, and as deep as the number of points in the array being transferred. Memory space for the array must be declared before any data can be transferred from the analyzer to the computer.

As mentioned earlier, the analyzer can transmit data over HP-IB in five different formats. The type of format affects what kind of data array is declared (real or integer), because the format determines what type of data is transferred. Examples of data transfers using different formats are discussed below. The first, Example 3A, illustrates an ASCII transfer using FORM 4. For more information on the various data formats, turn to “Array Data Formats” located earlier in this chapter. For information on the various types of data that can be obtained (raw data, error-corrected data, etc.), see “Data Levels,” located later in this chapter.

Note

"Example 7C: Reading ASCII Disk Files to the Instrument Controller’s Disk File,” located later in this chapter, explains how to access disk files from a computer.
**Frequency-Related Arrays**

Frequency-related values are calculated for the analyzer displays. The only data available to the programmer are the start and stop frequencies or center and span frequencies of the selected frequency range.

In a linear frequency range, the frequency values can be easily calculated because the trace data points are equally spaced across the trace. Relating the data from a linear frequency sweep to frequency can be done by interrogating the start frequency, the frequency span, and the number of points in the trace.

Given that information, the frequency of point \( n \) in a linear-frequency sweep is represented by the equation:

\[
F = \text{Start frequency} + (n-1) \times \frac{\text{Span}}{(\text{Points}-1)}
\]

In most cases, this is an easy solution for determining the related frequency value that corresponds with a data point. This technique is illustrated in “Example 3B: Data Transfer Using FORM 4 (ASCII Format).”

When using log sweep or a list-frequency sweep, the points are not evenly spaced over the frequency range of the sweep. In these cases, the frequencies can be read directly out of the instrument with the OUTPLIML command. “Example 3D: Data Transfer Using Frequency Array Information” demonstrates this technique.

Executing OUTPLIML reports the limit-test results by transmitting:

- the stimulus point tested
- a number indicating the limit-test results
- the upper test limit at the stimulus point (if available)
- the lower test limit at the stimulus point (if available)

The numbers used to indicate the limit-test results are:

- a negative one (-1) for no test
- a zero (0) for fail
- a positive one (1) for pass

If there are no limits available, the analyzer transmits zeros.

This data is very useful when testing with limit lines. It provides a method of obtaining accurate frequency-stimulus values to correspond with the trace data. The other limit-related values may be discarded and the stimulus values used with the trace-data points.
Data-Processing Chain

This section describes the manner in which the analyzer processes measurement data. It includes information on data arrays, common output commands, data levels, the learn string, and the calibration kit string.

Data Arrays

Figure 2-3 shows the different kinds of data available within the instrument:

- raw measured data
- error-corrected data
- formatted data
- trace memory
- calibration coefficients

Trace memory can be directly output to a controller with OUTPMEMO; but it cannot be directly transmitted back. If an analyzer with time domain (option 10) contains 1601 points, the formatted data array will have only 401 points.

Figure 2-3. The Data-Processing Chain

All the data-output commands are designed to insure that the data transmitted reflects the current state of the instrument:

- OUTPDATA, OUTPRAW, and OUTPFORM will not transmit data until all formatting functions have completed.
- OUTPLIML, OUTPLIMM, and OUTPLIMF will not transmit data until the limit test has occurred (if activated).
- OUTPMARK will activate a marker if a marker is not already selected. It will also insure that any current marker searches have been completed before transmitting data.
OUTPMSTA insures that the statistics have been calculated for the current trace before transmitting data. If the statistics are not activated, it will activate the statistics long enough to update the current values before deactivating the statistics.

OUTPMWID insures that a bandwidth search has been executed for the current trace before transmitting data. If the bandwidth-search function is not activated, it will activate the bandwidth-search function long enough to update the current values before switching OFF the bandwidth-search functions.

Data Levels

Different levels of data can be read out of the instrument. Refer to the data-processing chain in Figure 2-3. The following list describes the different types of data that are available from the network analyzer.

- **Raw data**
  - The basic measurement data, reflecting the stimulus parameters, IF averaging, and IF bandwidth. If a full 2-port measurement calibration is activated, there are actually four raw arrays kept: one for each raw S-parameter. The data can be output to a controller with the commands OUTPRAW1, OUTPRAW2, OUTPRAW3, OUTPRAW4. Normally, only raw 1 is available, and it holds the current parameter. If a 2-port measurement calibration is active, the four arrays refer to $S_{11}$, $S_{21}$, $S_{12}$, and $S_{22}$ respectively. This data is represented in real/imaginary pairs.

- **Error-corrected data**
  - This is the raw data with error-correction applied. The array represents the currently measured parameter, and is stored in real/imaginary pairs. The error-corrected data can be output to a controller with the OUTPDATA; command. The OUTPMEM0; command reads the trace memory, if available. The trace memory also contains error-corrected data. Note that neither raw nor error-corrected data reflect such post-processing functions as electrical-delay offset, trace math, or time-domain gating.

- **Formatted data**
  - This is the array of data actually being displayed. It reflects all post-processing functions such as electrical delay and time domain. The units of the array output depend on the current display format. See Table 2-3 for the various units defined as a function of display format.

- **Calibration coefficients**
  - The results of a measurement calibration are arrays containing calibration coefficients. These calibration coefficients are then used in the error-correction routines. Each array corresponds to a specific error term in the error model. The HP 8753D/Option 01l Network Analyzer User’s details which error coefficients are used for specific calibration types, as well as the arrays those coefficients can be found in. Not all calibration types use all 12 arrays. The data is stored as real/imaginary pairs.

Generally, formatted data is the most useful of the four data levels, because it is the same information the operator sees on the display. However if post-processing is unnecessary (e.g. possibly in cases involving smoothing), error-corrected data may be more desirable. Error-corrected data also affords the user the opportunity to input the data to the network analyzer and apply post-processing at another time.
Learn String and Calibration-Kit String

The learn string is summary of the instrument state. It includes all the front-panel settings, the limit-test tables, and the list-frequency table for the current instrument state. It does not include calibration data or the information stored in the save/recall registers.

The learn string can be output to a controller with the OUTPLEAS; executable, which commands the analyzer to start transmitting the binary string. The string has a fixed length for a given firmware revision. It can not be more than 3000 bytes in length. The array has the same header as in FORM 1.

The calibration kit is a set of key characteristics of the calibration standards used to increase the calibration accuracy. There are default kits for several different connector types. There is also space for a user-defined calibration kit. The command OUTPCALK outputs the currently active calibration kit as a binary string in FORM 1. As with the learn string, the calibration-kit string has a fixed length for a given firmware revision. It can not be longer than 1000 bytes.
Error Reporting

This section describes the analyzer's error-reporting process. It includes information on status reporting, the status byte, the event-status registers, and the error output.

Status Reporting

The analyzer status reporting structure is depicted in Figure 2-4.

![Diagram of status reporting structure]

Figure 2-4. Status Reporting Structure
### Table 2-5. Status Bit Definitions

#### Status Byte

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Waiting for reverse GET</td>
<td>A one path, 2-port measurement calibration is active, and the instrument has stopped, waiting for the operator to connect the device for reverse measurement.</td>
</tr>
<tr>
<td>1</td>
<td>Waiting for forward GET</td>
<td>A one path, 2-port measurement calibration is active, and the instrument has stopped, waiting for the operator to connect the device for forward measurement.</td>
</tr>
<tr>
<td>2</td>
<td>Check event-status-register B</td>
<td>One of the enabled bits in event status register B has been set.</td>
</tr>
<tr>
<td>3</td>
<td>Check error queue</td>
<td>An error has occurred and the message has been placed in the error queue, but has not been read yet.</td>
</tr>
<tr>
<td>4</td>
<td>Message in output queue</td>
<td>A command has prepared information to be output, but it has not been read yet.</td>
</tr>
<tr>
<td>5</td>
<td>Check event-status register</td>
<td>One of the enabled bits in the event-status register has been set.</td>
</tr>
<tr>
<td>6</td>
<td>Request service</td>
<td>One of the enabled status-byte bits is causing an SRQ.</td>
</tr>
<tr>
<td>7</td>
<td>Preset</td>
<td>An instrument preset has been executed.</td>
</tr>
</tbody>
</table>

#### Event-Status Register

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Operation complete</td>
<td>A command for which OPC has been enabled has completed operation.</td>
</tr>
<tr>
<td>1</td>
<td>Request control</td>
<td>The analyzer has been commanded to perform an operation that requires control of a peripheral, and needs control of HP-IB. Requires pass-control mode.</td>
</tr>
<tr>
<td>2</td>
<td>Query error</td>
<td>The analyzer has been addressed to talk but there is nothing in the output queue to transmit.</td>
</tr>
<tr>
<td>3</td>
<td>Sequence Bit</td>
<td>A sequence has executed the assert SRQ command.</td>
</tr>
<tr>
<td>4</td>
<td>Execution error</td>
<td>A command was received that could not be executed.</td>
</tr>
<tr>
<td>5</td>
<td>Syntax error</td>
<td>The incoming HP-IB commands contained a syntax error. The syntax error is can only be cleared by a device clear or an instrument preset.</td>
</tr>
<tr>
<td>6</td>
<td>User request</td>
<td>The operator has pressed a front-panel key or turned the RPG.</td>
</tr>
<tr>
<td>7</td>
<td>Power on</td>
<td>A power-on sequence has occurred since the last read of the register.</td>
</tr>
</tbody>
</table>

#### Event-Status-Register B

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Single sweep, number of groups, or calibration step complete</td>
<td>A single sweep, group, or calibration step has been completed since the last read of the register.</td>
</tr>
<tr>
<td>1</td>
<td>Service routine waiting or done</td>
<td>An internal service routine has completed operation, or is waiting for an operator response.</td>
</tr>
<tr>
<td>2</td>
<td>Data entry complete</td>
<td>A terminator key has been pressed or a value entered over HP-IB since the last read of the register.</td>
</tr>
<tr>
<td>3</td>
<td>Limit failed, Channel 2</td>
<td>Limit test failed on Channel 2.</td>
</tr>
<tr>
<td>4</td>
<td>Limit failed, Channel 1</td>
<td>Limit test failed on Channel 1.</td>
</tr>
<tr>
<td>5</td>
<td>Search failed, Channel 2</td>
<td>A marker search was executed on Channel 2, but the target value was not found.</td>
</tr>
<tr>
<td>6</td>
<td>Search failed, Channel 1</td>
<td>A marker search was executed on Channel 1, but the target value was not found.</td>
</tr>
<tr>
<td>7</td>
<td>Copy Complete</td>
<td>A copy has been completed since the last read of the register.</td>
</tr>
</tbody>
</table>
The Status Byte

The analyzer has a status-reporting mechanism that reports information about specific analyzer functions and events. The status byte (consisting of summary bits) is the top-level register. Each bit reflects the condition of another register or queue. If a summary bit is set (equals 1), the corresponding register or queue should be read to obtain the status information and clear the condition. Reading the status byte does not affect the state of the summary bits. The summary bits always reflect the condition of the summarized queue or register. The status byte can be read by a serial poll or by using the command OUTPSTAT. When using this command, the sequencing bit can be set by the operator during the execution of a test sequence. OUTPSTAT does not automatically put the instrument in remote mode, thus giving the operator access to the analyzer front-panel functions.

The status byte:
- summarizes the error queue
- summarizes two event-status registers that monitor specific conditions inside the instrument
- contains a bit that is set when the instrument is issuing a service request (SRQ) over HP-IB
- contains a bit that is set when the analyzer has data to transmit over HP-IB

Any bit in the status byte can be selectively enabled to generate a service request (SRQ) when set. Setting a bit in the service-request-enable register with the SREN[nn] executable enables the corresponding bit in the status byte. The units variable nn represents the binary equivalent of the bit in the status byte. For example, SRE24; enables status-byte bits 3 and 4 (since $2^3 + 2^4 = 24$) and disables all the other bits. SRE will not affect the state of the status-register bits.

The status byte also summarizes two queues: the output queue and the error queue. (The error queue is described in the next section.) When the analyzer outputs information, it puts the information in the output queue where it resides until the controller reads it. The output queue is only one event long. Therefore, the next output request will clear the current data. The summary bit is set whenever there is data in the output queue.
The Event-Status Register and Event-Status-Register B

The event-status register and event-status-register B are the other two registers in the status-reporting structure. They are selectively summarized by bits in the status byte via enable registers. The event-status registers consist of latched bits. A latched bit is set at the beginning of a specific trigger condition in the instrument. It can only be cleared by reading the register. The bit will not be reactivated until the condition occurs again. If a bit in one of these two registers is enabled, it is summarized by the summary bit in the status byte. The registers are enabled using the commands ESEnn; and ESBnn; both of which work in the same manner as SREEnn. The units variable unn represents the binary equivalent of the bit in the status byte.

If a bit in one of the event-status registers is enabled, and therefore, summary bit in the status byte is enabled, an SRQ will be generated. The SRQ will not be cleared until one of the five following conditions transpire:

1. The event-status register is read, clearing the latched bit.
2. The summary bit in the status byte is disabled.
3. The event-status-register bit is disabled.
4. The status registers are cleared with the CLES; command.
5. An instrument preset is performed.

Service requests generated when there are error messages or when the instrument is waiting for the Group Execute Trigger (GET) command are cleared by:

- reading the errors
- issuing GET (disabling the bits)
- clearing the status registers

Error Output

When an error condition is detected in the analyzer, an message is generated, displayed on the analyzer's display screen, and placed in the error queue. Error messages consist of an error number followed by an ASCII string no more than 50-characters long. The string contains the same message that appears on the analyzer's display. The error queue holds up to 20 error messages in the order in which they occur. The error messages remain in the error queue until the errors are read by the system controller using the command OUTPERRO. The OUTPERRO command outputs one error message.

Note

The error queue can only be cleared by performing an instrument preset or by cycling the line power. In order to keep the queue up-to-date, it is important to read all of the messages out of the queue each time errors are detected.
Measurement Process

This section explains how to organize instrument commands into a measurement sequence. A typical measurement sequence consists of the following steps:

1. setting up the instrument
2. calibrating the test setup
3. connecting the device under test
4. taking the measurement data
5. post-processing the measurement data
6. transferring the measurement data

Step 1. Setting Up the Instrument

Define the measurement by setting all of the basic measurement parameters. These include:

- the sweep type
- the frequency span
- the sweep time
- the number of points (in the data trace)
- the RF power level
- the type of measurement
- the IF averaging
- the IF bandwidth

You can quickly set up an entire instrument state, using the save/recall registers and the learn string. The learn string is a summary of the instrument state compacted into a string that the computer reads and retransmits to the analyzer. See “Example 5A: Using the Learn String.”

Step 2. Calibrating the Test Setup

After you have defined an instrument state, you should perform a measurement calibration. Although it is not required, a measurement calibration improves the accuracy of your measurement data.

The following list describes several methods to calibrate the analyzer:

- Stop the program and perform a calibration from the analyzer’s front panel.
- Use the computer to guide you through the calibration, as discussed in “Example 2A: S11 1-Port Measurement Calibration” and “Example 2B: Full 2-Port Measurement Calibration.”
- Transfer the calibration data from a previous calibration back into the analyzer, as discussed in “Example 5C: Saving and Restoring the Analyzer Instrument State.”

Step 3. Connecting the Device under Test

After you connect your test device, you can use the computer to speed up any necessary device adjustments such as limit testing, bandwidth searches, and trace statistics.
Step 4. Taking the Measurement Data

Measure the device response and set the analyzer to hold the data. This captures the data on the analyzer display.

By using the single-sweep command (SING), you can insure a valid sweep. When you use this command, the analyzer completes all stimulus changes before starting the sweep, and does not release the HP-IB hold state until it has displayed the formatted trace. Then when the analyzer completes the sweep, the instrument is put into hold mode, freezing the data. Because single sweep is OPC-compatible, it is easy to determine when the sweep has been completed.

The number-of-groups command (NUMGn) triggers multiple sweeps. It is designed to work the same as single-sweep command. NUMGn is useful for making a measurement with an averaging factor n (n can be 1 to 999). Both the single-sweep and number-of-groups commands restart averaging.

Step 5. Post-Processing the Measurement Data

Figure 2-3 shows the process functions used to affect the data after you have made an error-corrected measurement. These process functions have parameters that can be adjusted to manipulate the error-corrected data prior to formatting. They do not affect the analyzer's data gathering. The most useful functions are trace statistics, marker searches, electrical-delay offset, time domain, and gating.

After Performing and activating a full 2-port measurement calibration, any of the four S-parameters may be viewed without taking a new sweep.

Step 6. Transferring the Measurement Data

Read your measurement results. All the data-output commands are designed to insure that the data transmitted reflects the current state of the instrument.

Data transfer is also discussed in Example 3.
Programming Examples

This section contains a description of the operations and the required commands for the following programming examples used to make device measurements with the network analyzer:

- Example 1: Measurement Setup
  - Example 1A: Setting Parameters
  - Example 1B: Verifying Parameters

- Example 2: Measurement Calibration
  - Example 2A: S11 1-Port Measurement Calibration
  - Example 2B: Full 2-Port Measurement Calibration

- Example 3: Measurement Data Transfer
  - Example 3A: Data Transfer Using Markers
  - Example 3B: Data Transfer Using FORM 4 (ASCII Format)
  - Example 3C: Data Transfer Using Floating-Point Numbers
  - Example 3D: Data Transfer Using Frequency-Array Information
  - Example 3E: Data Transfer Using FORM 1 (Internal Binary Format)

- Example 4: Measurement Process Synchronization
  - Example 4A: Using the Error Queue
  - Example 4B: Generating Interrupts
  - Example 4C: Power Meter Calibration

- Example 5: Network Analyzer System Setups
  - Example 5A: Using the Learn String
  - Example 5B: Reading Calibration Data
  - Example 5C: Saving and Restoring the Analyzer Instrument State

- Example 6: Limit-Line Testing
  - Example 6A: Setting up a List-Frequency Sweep
  - Example 6B: Selecting a Single Segment from a Table of Segments
  - Example 6C: Setting Up Limit Lines
  - Example 6D: Performing PASS/FAIL Tests While Tuning

- Example 7: Report Generation
  - Example 7A1: Analyzer Operation Using the Talker/Listener Mode
  - Example 7A2: Controlling Peripherals with the Pass-Control Mode
  - Example 7A3: Printing with the Serial Port
  - Example 7B1: Plotting to a File and Transferring the File Data to a Plotter
  - Example 7B2: Reading Plot Files From a Disk
    - Utilizing PC-Graphics Applications Using the Plot File
Example 7C: Reading ASCII Disk Files to the Instrument Controller’s Disk File (HP 8753D standard only)

Program Information

The following information is provided for every example program included on the “HP 8753D/Option 011 Programming Examples” disk:

■ A program description
■ An outline of the program’s processing sequence
■ A step-by-step instrument-command-level tutorial explanation of the program including:
  □ The command mnemonic and command name for the HP-IB instrument command used in the program.
  □ An explanation of the operations and affects of the HP-IB instrument commands used in the program.

Note The HP BASIC programming code for each of these examples is contained in “HP BASIC Programming Examples.”

Analyzer Features Helpful in Developing Programming Routines

Analyzer-Debug Mode

The analyzer-debug mode aids you in developing programming routines. The analyzer displays the commands being received. If a syntax error occurs, the analyzer displays the last buffer and points to the first character in the command line that it could not understand.

You can enable this mode from the front panel by pressing LOCAL HP-IB DIAG ON. The debug mode remains activated until you preset the analyzer or deactivate the mode. You can also enable this mode over the HP-IB using the DEBU ON; executable and disable the debug mode using the DEBU OFF; executable.

User-Controllable Sweep

There are two important advantages to using the single-sweep mode:

1. The user can initiate the sweep.
2. The user can determine when the sweep has completed.
3. The user can be confident that the trace data has been derived from a valid sweep.

Execute the command string OPC?; SING; to place the analyzer in single-sweep mode and trigger a sweep. Once the sweep is complete, the analyzer returns an ASCII character one (1) to indicate the completion of the sweep.

Note The measurement cycle and the data acquisition cycle must always be synchronized. The analyzer must complete a measurement sweep for the data to be valid.
Example 1: Measurement Setup

Example 1A: Setting Parameters

Note
This program is stored as EXAMP1A on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

You can use the same command sequence for manually setting up measurements or remotely setting up measurements via HP-IB. There is no required order, as long as you set the desired frequency range, number of points, and power level prior to performing a measurement calibration.

This example sets the following parameters:
- reflection log magnitude on channel 1
- reflection phase on channel 2
- dual channel display mode
- frequency range from 100 MHz to 500 MHz

The following is an outline of the program’s processing sequence:
- The system is initialized.
- The analyzer is adjusted to measure return loss on channel 1 and display it in log magnitude.
- The analyzer is adjusted to measure return loss on channel 2 and display the phase.
- The dual-channel display mode is activated.
- The system operator is prompted to enter the frequency range of the measurement.
- The displays are autoscaled.
- The analyzer is released from remote control and the program ends.

Tutorial Program Explanation

The following section explains “Example 1A: Setting Parameters” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference” in this guide.

Initializing the System

IFC (HP-IB Meta-Message: Interface Clear)

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

SDC (HP-IB Meta-Message: Selected Device Clear)

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

"0PC?;PRES;" (Instrument Preset with Operation Complete Query)

Resets the network analyzer’s operating state to the default state (preset). This command is similar to pressing [PRES] on the front panel, though it does not initiate an analyzer self-test. An operation-complete query (OPC?) asks for the analyzer to return an ASCII character one (1) to the system controller when the preset operation is complete.
Main Program
"CHAN1;" (Select Channel 1)
Selects channel 1 as the active channel.

"S11;" (Select Reflection)
Selects a reflection-measurement mode.

"LOGM;" (Select Log-Magnitude Display)
Sets the data display to a log-magnitude format.
This command could be joined together with other commands in a single string of characters and sent to the analyzer in one operation as follows:

"CHAN1;S11;LOGM;"
Notice the use of semicolons as the individual command delimiters and terminators.

"CHAN2;" (Select Active Channel 2)
Selects channel 2 as the active channel.

"S11;" (Select Reflection)
Selects a reflection-measurement mode.

"PHAS;" (Select Phase Display)
Sets the analyzer to display phase data.

"DUA CON;" (Activate Dual-Channel Display)
Displays both channels simultaneously in a dual-display configuration.

ENTER START FREQUENCY (MHz):, F_start (Operator Prompt)
Prompts the operator to enter the start frequency to define the beginning of the network analyzer's frequency span. F_start is the program's input variable for start frequency, input by the operator (in MHz).

ENTER STOP FREQUENCY (MHz):, F_stop (Operator Prompt)
Prompts the operator to enter the stop frequency to define the end of the network analyzer's frequency span. F_stop is the program's input variable for the stop frequency data, input by the operator (in MHz).

"STAR";F_start;"MHz" (Select Start Frequency in MHz)
Inputs the operator-defined start frequency, using MHz as the basic units, to the network analyzer.
"STOP"; F_stop; "MHZ" (Select Stop Frequency in MHz)
Inputs the operator-defined stop frequency, using MHz as the basic units, to the network analyzer. The selection of the analyzer's frequency span is now complete.

"CHAN1;AUTO;" (Select Channel 1 and Autoscale)
Selects channel 1 and autoscales the display to provide the optimum view of the measurement data.

"CHAN2;AUTO;" (Select Channel 2 and Autoscale)
Selects channel 2 and autoscales the display to provide the optimum view of the measurement data.

"OPC?;WAIT;" (Operation Complete Query and Wait for Reply)
The controller waits for the analyzer to return a one (1), signifying that the last command received from the controller has completed. Upon reading the one, the analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.

Running the Program
The analyzer is initialized and the operator is queried for the measurement's start and stop frequencies. The analyzer is set up to display the S11 reflection measurement as a function of log magnitude and phase over the selected frequency range. The displays are autoscaled and the program ends.
Example 1B: Verifying Parameters

Note
This program is stored as EXAMP1B on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

This example shows how to read analyzer settings into your program. The “Reading Analyzer Data” section, located earlier, contains information on the command formats and operations. Appending a question mark (?) to a command that sets an analyzer parameter will return the value of that setting from the analyzer to the controller. Parameters that are set as ON or OFF when queried will return a one (1) if active or a zero (0) if OFF. Parameters are returned in ASCII format, FORM 4. This format varies in length from 1 to 24 characters-per-value. In cases of marker or other multiple responses, these values are separated by commas.

The following is an outline of the program’s processing sequence:

- The system is initialized.
- The number of points in the trace is queried and printed on the controller display.
- The start frequency is queried and printed on the controller display.
- The averaging function state is queried and printed on the controller display.
- The analyzer is released from remote control and the program ends.

Tutorial Program Explanation

The following section explains “Example 1B: Verifying Parameters” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”

Initializing the System

IFC (HP-IB Meta-Message: Interface Clear)

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

SDC (HP-IB Meta-Message: Selected Device Clear)

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

"OPC?;PRES;" (Instrument Preset with Operation Complete Query)

Resets the network analyzer’s operating state to the default state (preset). This command is similar to pressing [Preset] on the front panel, though it does not initiate an analyzer self-test. An operation-complete query (OPC?) asks for the analyzer to return an ASCII character one (1) to the system controller when the preset operation is complete.
Main Program

"POIN?" (Number of Points Query)
Queries the analyzer for the number of points in the measurement trace. The returned value is displayed on the controller's CRT.

"STAR?" (Start Frequency Query)
Queries the start frequency. The returned value is displayed on the controller's CRT.

"AVER?" (Averaging Status Query)
Queries the averaging function to determine whether averaging is currently active. Notice that this is an ON or OFF command. The ON or OFF is replaced by a "?" to return the analyzer's averaging state to the controller.
The returned value is tested as a flag, and "Averaging ON" or "Averaging OFF" is displayed on the controller as a result of that test.

"OCEP?; WAIT;" (Wait for Reply with Operation Complete Query)
The controller waits for the analyzer to return a one (1), signifying that the last command received from the controller has completed. Upon reading the one, the analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.

Running the Program
The analyzer is preset. The preset values are returned for: the number of points, the start frequency, and the state of the averaging function. The analyzer is released from remote control and the program ends.
Example 2: Measurement Calibration

This section shows you how to coordinate a measurement calibration over HP-IB. You can use the following sequence for performing either a manual measurement calibration, or a remote measurement calibration via HP-IB:

1. Select the calibration type.
2. Measure the calibration standards.
3. Declare the calibration done.

The actual sequence depends on the calibration kit and changes slightly for full 2-port measurement calibrations, which are divided into three calibration sub-sequences.

Calibration kits

The calibration kit tells the analyzer what standards to expect at each step of the calibration. The set of standards associated with a given calibration is termed a "class." For example, measuring the short during an S11 1-port measurement calibration is one calibration step. All of the shorts that can be used for this calibration step make up the class, which is called class S11B. For the 7-mm and the 3.5-mm cal kits, class S11B uses only one standard. For type-N cal kits, class S11B contains two standards: male and female shorts.

When doing an S11 1-port measurement calibration use a 7- or 3.5-mm calibration kit. Selecting SHORT automatically measures the short because the class contains only one standard. When doing the same calibration in type-N, selecting SHORT brings up a second menu, allowing the operator to select which standard in the class is to be measured. The sex listed refers to the test port: if the test port is female, then the operator selects the female short option.

Doing an S11 1-port measurement calibration over HP-IB is very similar. When using a 7- or 3.5-mm calibration kit, sending CLASS11B will automatically measure the short. In type-N, sending CLASS11B brings up the menu with the male and female short options. To select a standard, use STANA or STANB. The STAN command is appended with the letters A through G, corresponding to the standards listed under softkeys 1 through 7, softkey 1 being the topmost softkey.

The STAN command is OPC-compatible. A command that calls a class is only OPC-compatible if that class has only one standard in it. If there is more than one standard in a class, the command that calls the class brings up another menu, and there is no need to query it.

Example 2A: S11 1-Port Measurement Calibration

Note

This program is stored as EXAMP2A on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

This example shows you how to coordinate an S11 1-port measurement calibration over HP-IB, using the HP 85032B 500 type-N calibration kit.

1. Set up the desired instrument state.
2. Run the program.
3. Connect the standards as prompted.
4. Press [ENTER] on the controller keyboard to measure the standard.
Information on selecting calibration standards can be found earlier in the “Measurement Process” section.

The following is an outline of the program's processing sequence:

- The system is initialized.
- The appropriate calibration kit is selected.
- The softkey menu is deactivated.
- The $S_{11}$ 1-port measurement calibration sequence is run.
- The $S_{11}$ 1-port measurement calibration data is saved.
- The softkey menu is activated.
- The analyzer is released from remote control and the program ends.

**Tutorial Program Explanation**

The following section explains “Example 2A: $S_{11}$ 1-Port Measurement Calibration” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”

**Initializing the System**

**IFC (HP-IB Meta-Message: Interface Clear)**

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

**Main Program**

"CALKN50;" (Select 500 Type-N Calibration Kit)

Selects the 500 type-N calibration kit.

"MENUGOFF;" (Deactivate Softkey Menu)

Deactivates the softkey menu.

"CALIS111;" (Initiate $S_{11}$ 1-Port Measurement Calibration)

Begins the calibration cycle by selecting the $S_{11}$ 1-port measurement calibration.
"CONNECT OPEN AT PORT 1" (Operator Prompt)
Prompts the operator to connect the first calibration standard, the open, at PORT 1. 
CALL Waitforkey

Subroutine: Waitforkey:

Note: In order to present a streamlined explanation of the example programs, subroutines and their associated commands are only discussed at the first point they are accessed in the program.

"PG;PU;PA390,3700;PD;LB;Lab$;", PRESS ENTER WHEN READY;" (Position and Display Text Commands and Operator Prompt)
A subroutine is accessed to position and display the text prompt on the controller’s display console. It is accessed every time the operator is required to press (ENTER).

" Press ENTER when ready" (Operator Prompt)
Prompts the operator to press (ENTER) on the controller keyboard after connecting the open at PORT 1. The controller reads the (ENTER) key-press and the calibration sequence continues.

"PG;" (Page)
Removes the last prompt message ("Press ENTER when ready") from the analyzer’s display. 
This is the last subroutine command and the subroutine ends.

Main Program Resumes
"CLASS11A;" (Select Open Reflection Class)
Measures the open. There is more than one standard in this loads class, so you must identify the specific standard within that class. The female open is the second softkey selection from the top in the menu.

"OCP?;STANB;" (Select Standard B with Operation Complete Query)
The command STANB selects a lowband load to be used as the standard. The analyzer returns an ASCII one (1) when the measurement is complete.

"CONNECT SHORT AT PORT 1" (Operator Prompt)
Prompts the operator to connect the next calibration standard, the short, at PORT 1. 
CALL Waitforkey

"CLASS11B;" (Select Short Reflection Class)
Measures the short. There is more than one standard in this loads class, so you must identify the specific standard within that class. The female short is the second softkey selection from the top in the menu (choice “B”).

"OCP?;STANB;" (Select Standard B with Operation Complete Query)
The command STANB selects a lowband load to be used as the standard. The analyzer returns an ASCII one (1) when the measurement is complete.

"CONNECT LOAD AT PORT 1" (Operator Prompt)
Prompts the operator to connect the 50Ω load to PORT 1. The program operation continues when you connect the load and press ENTER on the controller keyboard.
CALL Waitforkey

"OPC?;CLASS11C;" (Select Load Reflection with Operation Complete Query)
Measures the load. There is only one standard choice in this class, so the CLASS command is OPC-compatible. Using the OPC? command causes the program to wait until the standard is measured before continuing. This is very important, because the prompt to connect the next standard must appear after the first standard is measured. An ASCII one (1) is returned once the analyzer has measured the load standard.

"PG;" (Page)
Removes the last prompt message ("CONNECT LOAD AT PORT 1") from the controller's display.

"COMPUTING CALIBRATION COEFFICIENTS" (Program Message)
Indicates that the calibration constants are being calculated by the analyzer and put into the calibration arrays for use with the measurement.

"DONE;" (Calibration Complete)
Completes the calibration process.

"OPC?;SAV1;" (Save 1-Port Measurement Calibration with Operation Complete Query)
Saves the instrument state and resulting calibration arrays as a 1-port measurement calibration. The OPC? command creates a wait-for-completion condition until the analyzer has finished calculating and storing the calibration coefficients.

"S11 1-Port measurement cal COMPLETED. CONNECT TEST DEVICE." (Prompt)
Indicates that the calibration is complete.

"MENUON;" (Activate Softkey Menu)
Reactivates the analyzer's softkey menu.

"OPC?;WAIT;" (Wait for Reply with Operation Complete Query)
The controller waits for the analyzer to return a one (1), signifying that the last command received from the controller has completed. Upon reading the one (1), the analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.
Running the Program

This program assumes the following test port characteristics:

- 50 ohm
- type-N connectors
- female

The prompts appear just above the message line on the analyzer’s and the controller’s display. Pressing **ENTER** on the controller continues the program and measures the standard. The program displays a message when the measurement calibration is complete.

Example 2B: Full 2-Port Measurement Calibration

Note

This program is stored as EXAMP2B on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

The following example shows how to perform a full 2-port measurement calibration using the HP 85032B calibration kit. The main difference between this example and Example 2A is that the full 2-port measurement calibration process allows removal of both the forward- and reverse-error terms, so that all four S-parameters of the device under test can be measured. PORT 1 is a female test port and PORT 2 is a male test port.

The following is an outline of the program’s processing sequence:

- The system is initialized.
- The appropriate calibration kit is selected.
- The softkey menu is deactivated.
- The 2-port calibration sequence is run.
- The operator is prompted to choose or skip the isolation calibration.
- The softkey menu is activated.
- The analyzer is released from remote control and the program ends.

Tutorial Program Explanation

The following section explains “Example 2B: Full 2-Port Measurement Calibration” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “HP-IB Command Reference.”

Initializing the System

IFC (HP-IB Meta-Message: Interface Clear)

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

SDC (HP-IB Meta-Message: Selected Device Clear)

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.
Reflection Portion of the Full 2-Port Measurement Calibration Program
"CALKN50;" (Select 50Ω Type-N Calibration Kit)
Selects the 50Ω type-N calibration kit.

"MENUG0F;" (Deactivate Softkey Menu)
Deactivates the softkey menu.

"CALIFUL2;" (Initiate Full 2-Port Measurement Calibration)
Begins the full 2-port measurement calibration cycle.

"REFL;" (Select Reflection Calibration)
Selects the reflection calibration subsequence first.

"CONNECT OPEN AT PORT 1" (Operator Prompt)
Prompts the operator to connect the first calibration standard, the open, at PORT 1.

CALL Waitforkey

Subroutine: Waitforkey:

Note
In order to present a streamlined explanation of the example programs, subroutines and their associated commands are only discussed at the first point they are accessed in the program.

"PG;PU;PA390,3700;PD;LB;Lab$;", PRESS ENTER WHEN READY;" (Position and Display Text Commands and Operator Prompt)
A subroutine is accessed to position and display the text prompt on the controller’s display console. It is accessed every time the operator is required to press [ENTER].

" Press ENTER when ready" (Operator Prompt)
Prompts the operator to press [ENTER] on the controller keyboard after connecting the open at PORT 1. The controller reads the [ENTER] key-press and the calibration sequence continues.

"PG;" (Page)
Removes the last prompt message ("Press ENTER when ready") from the analyzer’s display.
This is the last subroutine command and the subroutine ends.

Reflection Portion of the Full 2-Port Measurement Calibration Program Resumes
"CLASS11A;" (Select Open Reflection Class)
Measures the open. There is more than one standard in this loads class, so you must identify the specific standard within that class.

"0PC?;STANB;" (Select Standard B with Operation Complete Query)
The command STANB selects a female open to be used as the standard. The analyzer returns an ASCII one (1) when the measurement is complete. The female open is the second softkey selection from the top in the menu.

"CONNECT SHORT AT PORT 1" (Operator Prompt)
Prompts the operator to connect the next calibration standard, the short, at PORT 1.
CALL Waitforkey

"CLASS11B;" (Select Short Reflection Class)
Measures the short. There is more than one standard in this loads class, so you must identify the specific standard within that class. The female short is the second softkey selection from the top in the menu (choice “B”).

"OPC?;STANB;" (Select Standard B with Operation Complete Query)
The command STANB selects a female short to be used as the standard. The analyzer returns an ASCII one (1) when the measurement is complete.

"CONNECT LOAD AT PORT 1" (Operator Prompt)
Prompts the operator to connect the 50Ω load to PORT 1. The program operation continues when you connect the load and press **ENTER** on the controller keyboard.
CALL Waitforkey:

"OPC?;CLASS11C;" (Select Load Reflection with Operation Complete Query)
Measures the load. There is only one standard choice in this class, so the CLASS command is OPC-compatible. Using the OPC? command causes the program to wait until the standard is measured before continuing. This is very important, because the prompt to connect the next standard must appear after the first standard is measured. An ASCII one (1) is returned once the analyzer has measured the load standard.

"CONNECT OPEN AT PORT 2" (Operator Prompt)
Prompts the operator to connect the next calibration standard, the open, at PORT 2.

"CLASS22A;" (Select Open Reflection Class)
Measures the open. There is more than one standard in this loads class, so you must identify the specific standard within that class. The male open is the first softkey selection from the top in the menu.

"OPC?;STANA;" (Select Standard A with Operation Complete Query)
The command STANA selects a lowband load to be used as the standard. The analyzer returns an ASCII one (1) when the measurement is complete.

"CONNECT SHORT AT PORT 2" (Operator Prompt)
Prompts the operator to connect the next calibration standard, the short, at PORT 2.
"CLASS22B;" (Select Short Reflection Class)
Measures the short. There is more than one standard in this class, so you must identify the specific standard within that class. The male short is the first softkey selection from the top in the menu (choice “A”).

"OPC?;STANA" (Select Standard A with Operation Complete Query)
The command STANA selects a lowband load to be used as the standard. The analyzer returns an ASCII one (1) when the measurement is complete.

"CONNECT LOAD AT PORT 2" (Operator Prompt)
Prompts the operator to connect the 50Ω load to PORT 2.

"OPC?;CLASS11C;" (Select Load Reflection with Operation Complete Query)
Measures the load. There is only one standard choice in this class, so the CLASS command is OPC-compatible. Using the OPC? command causes the program to wait until the standard is measured before continuing. This is very important, because the prompt to connect the next standard must appear after the first standard is measured. An ASCII one (1) is returned once the analyzer has measured the load standard.

"COMPUTING REFLECTION CALIBRATION COEFFICIENTS" (Program Comment)
Indicates that the reflection calibration coefficients are being calculated by the analyzer.

"REFD;" (Reflection Calibration Done)
Completes the reflection cycle and the ends the reflection calibration routine.
The analyzer now computes the reflection calibration coefficients for the calibration arrays.

Transmission Portion of the Full 2-Port Measurement Calibration Program Resumes
"TRAN;" (Select Transmission Calibration)
Begins the transmission calibration cycle.

"CONNECT THRU [PORT 1 TO PORT 2];" (Operator Prompt)
Prompts the operator to make a thru connection between PORT 1 and PORT 2.

"MEASURING FORWARD TRANSMISSION" (Program Comment)
Displays a message on the controller console showing that the forward transmission is being measured.

"OPC?;FWDT;" (Select FWD Transmission with Operation Complete Query)
Measures the forward transmission. A one (1) is received from the analyzer after the measurement is complete.

"OPC?;FWDM;" (Select FWD Trans Load Match with Operation Complete Query)
Measures the forward load match. Again, a one (1) is received when the analyzer completes the measurement.

"MEASURING REVERSE TRANSMISSION" (Program Comment)
The prompt is displayed on the controller console informing the operator that the reverse transmission measurements are taking place.

"OPC?;REV;" (Select Reverse Transmission with Operation Complete Query)
Measures the reverse transmission. A one (1) is received from the analyzer after the measurement is complete.

"OPC?;REVM;" (Select REV Trans Load Match with Operation Complete Query)
Measures the reverse load match. Again, a one (1) is received when the analyzer completes the measurement.

"TRAD;" (Transmission Calibration Done)
Completes the transmission cycle and the ends the transmission-calibration routine.

Isolation Portion of the Full 2-Port Measurement Calibration Program Resumes

"SKIP ISOLATION CAL? Y OR N." (Operator Prompt)
Prompts the operator to choose whether to perform the isolation-calibration step of the full 2-port measurement calibration cycle. The response is read into a string and tested.

- If the operator responds by entering "Y", the isolation cycle is skipped with the omit-isolation command executable "OMI". The program then branches to the isolation-done command at the end of the isolation tests.
- If the operator responds with any other character to the prompt, the isolation tests are performed.

"ISOLATE TEST PORTS" (Operator Prompt)
Prompts the operator to connect loads to the test ports in preparation for the isolation measurement.

"ISOL;" (Select Isolation Calibration)
Selects the reverse-isolation calibration.

"AVERFACT10;" (Select Averaging Factor {10})
Selects averaging for ten sweeps.

"AVER0ON;" (Activate Averaging)
Activates the averaging function.

"MEASURING REVERSE ISOLATION" (Program Comment)
Displays "MEASURING REVERSE ISOLATION" on the controller’s console.
"OPC?:REV;" (Select Reverse Isolation with Operation Complete Query)
Performs a reverse-isolation calibration. Ten sweeps are taken and measured. The noise level is averaged to obtain the best test results. A one (1) is returned from the analyzer to indicate the completion of the operation.

"MEASURING FORWARD ISOLATION" (Program Comment)
Displays "MEASURING FORWARD ISOLATION" on the controller’s console.

"OPC?:FWDI;" (Select Forward Isolation with Operation Complete Query)
Measures the forward isolation. A one (1) is returned from the analyzer to indicate the completion of the operation.

"ISOD;AVER0FF;" (Isolation Calibration Done and Deactivate Averaging)
This completes the isolation measurements and the isolation calibration cycle ends.

"PG;" (Page)
Removes the last prompt message ("MEASURING FORWARD ISOLATION") from the controller’s display.

"COMPUTING CALIBRATION COEFFICIENTS" (Program Comment)
Displays "COMPUTING CALIBRATION COEFFICIENTS" on the controller’s console.

"DONE;" (Completed)
Completes the calibration process.

"OPC?:SAV2;" (Save Full 2-Port Measurement Calibration with Operation Complete Query)
Saves the instrument state and resulting calibration arrays as a 2-port measurement calibration. The OPC? command creates a wait-for-completion condition until the analyzer responds that it has finished calculating and storing the calibration coefficients in its internal registers.

"DONE WITH FULL 2-PORT CAL. CONNECT TEST DEVICE." (Operator Prompt)
Displays the final message on the controller console to alert the operator that the full 2-port measurement calibration process is complete.

"MENUON;" (Activate Softkey Menu)
Reactivates the analyzer’s softkey menu.

"OPC?:WAIT;" (Wait for Reply with Operation Complete Query)
The controller waits for the analyzer to return a one (1), signifying that the last command received from the controller has completed. Upon reading the one (1), the analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.
Running the Program

The program assumes the following situation:

- test ports are type-N
- PORT 1 is a female test port
- PORT 2 is a male test port
- HP 85032B 500 type-N calibration kit is used

The prompts appear just above the message line on the analyzer's display as well as the computer's console. Pressing [ENTER] on the computer keyboard continues the program and measures the standard. You have the option of omitting the isolation portion of the measurement calibration. If you perform the isolation portion of the calibration, averaging is automatically employed to insure a valid calibration. The program will display a message when the measurement calibration is complete.
Example 3: Measurement Data Transfer

There are two methods that can be used to read trace information from the analyzer:

- selectively, using the trace markers
- completely, using the trace-data array

If only specific information (such as a single point on the trace or the result of a marker search) is required, the marker output command can be used to read the information. If all of the trace data is required, see Examples 3B through 3E for examples of the various formats available.

Trace-Data Formats and Transfers

Refer to Table 2-4. This table shows the number of bytes required to transfer a 201-point trace in the different formats. As you will see in the first example FORM 4, ASCII data is the easiest to transfer, but the most time consuming due to the number of bytes in the trace. If you are using a PC-based controller, a more suitable format would be FORM 5. To use any trace data format other than FORM 4 (ASCII data) requires some care in transferring the data to the computer. Data types must be matched to read the bytes from the analyzer directly in to the variable array. The computer must be told to stop formatting the incoming data and treat it as a binary-data transfer. All of the other data formats also have a four byte header to deal with. The first two bytes are the ASCII characters "#A" that indicate that a fixed length block transfer follows, and the next two bytes form an integer containing the number of bytes in the block to follow. The header must be read in to separate the header from the rest of the block data to be mapped into an array. "Array-Data Formats," located earlier in this chapter, discusses the different types of formats and their compositions.

Data may also be transferred from several different locations in the trace-processing chain. These examples will illustrate formatted-data transfers, but other locations in the trace-data processing chains may be accessed. See Figure 2-3.

In this section, an example of each of the data formats will be shown for comparison. A general rule of thumb is to use FORM 1 (internal binary format) for traces that are not being utilized for data content. Learn strings, state transfers, and calibration data that are being transferred to a file and back are good examples. See Example 3D.

Arrays which will be interpreted or processed within your program should be in FORM 2, 3 or 5, whichever is appropriate for your computer. Example 3C shows how to transfer a trace in these formats.

In Examples 3B and 3C, the frequency counterpart of each data point in the array is also determined. Many applications generate a frequency and magnitude, or a phase array for the test results. Such data may be required for other data processing applications (such as comparing data from other measurements).

In Example 3B, the frequency data is constructed from the frequency span information. Alternatively, it is possible to read the frequencies directly out of the instrument with the OUTPLIML command. OUTPLIML reports the limit-test results by transmitting the stimulus point tested, a number indicating the limit-test results, and then the upper and lower limits at that stimulus point (if available). The number indicating the limit results is a -1 for no test, 0 for fail, and 1 for pass. If there are no limits available, the analyzer transmits zeros. For this example, we delete the limit test information and keep the stimulus information.

In Example 3C, the limit-test array is read into the controller and used to provide the values directly from the analyzer memory. Reading from the limit-test array is convenient, although it outputs the results in ASCII format (FORM 4), which may be slow. If there is no other way to obtain the frequency data, this transfer time may be acceptable. Frequency information becomes more difficult to determine when not using the linear sweep mode. Log-frequency
sweeps and list-frequency sweeps have quite different values for each data point. For these special cases, the additional time spent reading out the limit test results is an acceptable solution for obtaining the valid frequency information for each data point in the trace.

Example 3A: Data Transfer Using Markers

Note
This program is stored as EXAMP3A on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

Markers are the simplest form of trace-data transfer. You can position a marker in the following locations on the trace:

- a frequency location
- an actual data point location
- a trace data value

The marker data is always returned in FORM 4, ASCII format. Each number is sent as a 24-character string, each character being a digit, sign, or decimal point. In the case of markers, three numbers are sent. The display format determines the values of the marker responses. See Table 2-4 for additional information.

The following is an outline of the program’s processing sequence:

- The system is initialized.
- The selected frequency span is swept once.
- The marker is activated and placed on the maximum trace value.
- The three marker values (value 1, value 2, and stimulus) are output to the controller and displayed.
- The instrument is returned to local control and the program ends.

Tutorial Program Explanation

The following section explains “Example 3A: Data Transfer Using Markers” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”

Initializing the System

IFC (HP-IB Meta-Message: Interface Clear)

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

SDC (HP-IB Meta-Message: Selected Device Clear)

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.
"OPC?;PRES;" (Instrument Preset with Operation Complete Query)

Resets the network analyzer's operating state to the default state (preset). This command is similar to pressing (Preset) on the front panel, though it does not initiate an analyzer self-test. An operation-complete query (OPC?) asks for the analyzer to return an ASCII character one (1) to the system controller when the preset operation is complete.

**Main Program**

"OPC?;SING;" (Select Single-Sweep Mode with Operation Complete Query)

Places the analyzer in single-sweep mode and activates a sweep. The analyzer returns a one (1) at the completion of the sweep.

"MARK1;" (Select Active Marker {1})

Marker 1 is selected and activated.

"SEAMAX;" (Search Maximum)

The marker is placed at the maximum data value on the trace.

"OUTPMARK;" (Output Marker)

Outputs the active marker values in 3 numbers. The first two numbers are the marker values, and the last is the stimulus value. The data values are read from the marker function and output to the controller.

The analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.

**Running the Program**

Execute the program. After performing an instrument preset, the analyzer switches into the log-magnitude mode and measures the S11 reflection values of the device under test.

The three values returned to the controller are:

1. reflection, in dB
2. a non-significant value
3. the stimulus frequency at the maximum point

A non-significant value means that the analyzer returned a value that is meaningless in this data format.

Table 2-3, located earlier in this chapter, provides an easy reference for the types of data returned with the various data-format operational modes.
Example 3B: Data Transfer Using FORM 4 (ASCII Transfer)

Note
This program is stored as EXAMP3B on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

This example shows you how to transfer a trace array from the analyzer using FORM 4, an ASCII data transfer.

Table 2-3 shows the relationship of the two values-per-point that are transferred to the analyzer. When FORM 4 is used, each number is sent as a 24-character string (each character represented by a digit, sign, or decimal point). Since there are two numbers-per-point, plus a comma and line-feed, a 201-point transfer in FORM 4 takes 10,050 bytes. This form is useful only when input-data formatting is difficult with the instrument controller. Refer to Table 2-4 for a comparison with the other formats.

Another example is included with the ASCII data transfer. A fairly common requirement is to create frequency-amplitude data pairs from the trace data. No frequency information is included with the trace-data transfer. Relating the data from a linear frequency sweep to frequency can be done by interrogating the analyzer start frequency, the frequency span, and the number of points. Given that information, the frequency of point \( n \) in a linear frequency sweep is defined by the equation:

\[
F = \text{Start frequency} + (n-1) \times \frac{\text{Span}}{(\text{Points} - 1)}
\]

This example illustrates the technique of generating corresponding frequency data. This is a straight-forward solution for linear uniform sweeps. For other sweep types, frequency data is more difficult to construct and may be best read from the analyzer directly from the limit-test array. See “Example 3D: Data Transfer Using Frequency Array Information” for an example of this technique.

The following is an outline of the program’s processing sequence:

- The system is initialized.
- The trace-data array is allocated.
- The trace length is set to 11.
- The selected frequency span is swept once.
- The FORM 4, ASCII format is set.
- The formatted trace is read from the analyzer.
- The frequency increments between the points are calculated.
- The marker is activated and placed at 30 kHz.
- The instrument is returned to local control and the program ends.

Tutorial Program Explanation

The following section explains “Example 3B: Data Transfer Using FORM 4 (ASCII Transfer)” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”
Initializing the System

IFC (HP-IB Meta-Message: Interface Clear)
Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

SDC (HP-IB Meta-Message: Selected Device Clear)
Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

"OPC?;PRES;" (Instrument Preset with Operation Complete Query)
Resets the network analyzer's operating state to the default state (preset). This command is similar to pressing [Preset] on the front panel, though it does not initiate an analyzer self-test. An operation-complete query (OPC?) asks for the analyzer to return an ASCII character one (1) to the system controller when the preset operation is complete.

Main Program

"Poin 11;" (Select Number of Points {11})
Sets the number of points in the trace to 11.

"OPC?;SING;" (Select Single-Sweep Mode with Operation Complete Query)
Places the analyzer in single-sweep mode and activates a sweep. The analyzer returns a one (1) at the completion of the sweep.

"FORM4;" (Select Array-Data Format 4)
Sets array-data format 4, ASCII floating-point format.

"OUTPFORM;" (Output Formatted Trace-Data Array from Active Channel)
Outputs the formatted trace data from the active channel in current display units. The analyzer's trace data is output as an ordered pair separated by a comma and terminated with a line-feed character for each trace-data point. The trace-data array is then read by the controller.

"POIN?;" (Number of Points Query)
Queries the analyzer for the number of points in the measurement trace. The returned value is displayed on the controller's CRT.

"STAR?;" (Start Frequency Query)
Queries the start frequency. The returned value is displayed on the controller's CRT.

"SPAN?;" (Frequency Span Query)
Queries the frequency span. The returned value is displayed on the controller's CRT.
F_inc=Span/(Num_points-1) (Fixed-Frequency Increment Equation)

The program calculates the frequency increments between data points. The values are then displayed on the controller's display console. The data is then formatted for display on the controller's console. The program loops through the data points and calculates the frequency values for each point on the 11-point trace, as defined by "P01N 11;". The display-formatted frequency-value data is then shown on the controller's display console. The program then exits the loop and continues to the next command line.

**Note**
In this format, the second value in the array is not significant. See Table 2-4.

"MARKDISC;" (Select Discrete-Marker Mode)

Sets the analyzer to display discrete-data points with the marker. In discrete-marker mode, the trace data is not interpolated between points. This allows the operator to easily identify and view the datapoints in the trace.

"MARK1 3E+4;" (Select Active Marker 1 and position at 30 kHz)

Positions marker 1 at 30 kHz (the left-hand edge of the display).

"OPC?;WAIT;" (Wait for Reply with Operation Complete Query)

The controller waits for the analyzer to return a one (1), signifying that the last command received from the controller has completed. Upon reading the one (1), the analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB.

"Position the marker with the knob and compare values" (Operator Prompt)

The operator is prompted to position the marker with the RPG knob along the points in the trace and the program ends.

**Running the Program**

Run the program and watch the controller console. The analyzer will perform an instrument preset. The program will then print out the data values received from the analyzer. The marker is activated and placed at the left-hand edge of the analyzer display. Position the marker with the knob and compare the values read with the active marker with the results printed on the controller console. The data points should agree exactly. Keep in mind that no matter how many digits are displayed, the analyzer is specified to measure:

- magnitude to a resolution of 0.001 dB
- phase to a resolution of 0.01 degrees
- group delay to a resolution of 0.01 ps

Changing the display format will change the data sent with the OUTPFORM transfer. See Table 2-4 for a list of the specific data that is provided with each format. The data from OUTPFORM reflects all the post processing such as:

- time domain
- gating
- electrical delay
- trace math
- smoothing
Note
Note that if time domain (option 010 only) is activated, operation is limited to 201 points in the lowpass mode.

Example 3C: Data Transfer Using Floating-Point Numbers

Note
This program is stored as EXAMP3C on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

This example program illustrates data transfer using FORM 3 in which data is transmitted in the floating-point formats. FORM 2 is nearly identical except for the IEEE 32-bit format of 4 bytes-per-value. FORM 5 reverses the order of the bytes to conform with the PC conventions for defining a real number.

The block-data formats have a four-byte header. The first two bytes are the ASCII characters #A that indicate that a fixed-length block transfer follows, and the next two bytes form an integer containing the number of bytes in the block to follow. The header must be read in so that data order is maintained.

This transfer is more than twice as fast as a FORM 4 transfer. With the FORM 4 transfer, 10,050 bytes are sent (201 points × 2 values-per-point × 24 bytes-per-value). Using FORM 2 to transfer the data, only 1612 bytes are sent (201 points × 2 values-per-point × 4 bytes-per-value). See Table 2-4 “Array-Data Formats,” to compare the formats.

The following is an outline of the program’s processing sequence:

- The system is initialized.
- The integer variables are defined to contain the header information.
- The number of points in the trace is set to 11.
- The selected frequency span is swept once.
- Data-transfer format 3 is set.
- The headers are read from the trace.
- The array size is calculated and allocated.
- The trace data is read in and printed on the controller display.
- The marker is activated and placed at 30 kHz.
- The instrument is returned to local control and the program ends.

Tutorial Program Explanation

The following section explains “Example 3C: Data Transfer Using Floating-Point Numbers” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”
Initializing the System

IPC (HP-IB Meta-Message: Interface Clear)
Resets the HP-IB interface by toggling the state of the IPC line and taking control of the bus, halting all operations for all connected instruments.

SDC (HP-IB Meta-Message: Selected Device Clear)
Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

"OPC?;PRES;" (Instrument Preset with Operation Complete Query)
Resets the network analyzer’s operating state to the default state (preset). This command is similar to pressing [PRES] on the front panel, though it does not initiate an analyzer self-test. An operation-complete query (OPC?) asks for the analyzer to return an ASCII character one (1) to the system controller when the preset operation is complete.

Main Program

First, two integers are defined to hold the header information. Since an integer takes two bytes, the header and length variables will form the entire four-byte header.

The program variable Numpoints is defined as 11. The value “11” will be used in the next command line to set the number of points in the trace.

The program command line "P01N"; Numpoints; ";" selects the number of points in the trace using the value (11) for the variable Numpoints. This operation is virtually identical to executing the "P01N 11;" command used in Example 3B. Both operations set the number of points in the trace to 11.

"OPC?;SING;" (Select Single-Sweep Mode with Operation Complete Query)
Places the analyzer in single-sweep mode and activates a sweep. The analyzer returns a one (1) at the completion of the sweep.

"FORM3;" (Select Array-Data Format 3)
Selects array-data format 3. Data will now be output to the controller, formatted as 64-bit floating point numbers, with a four byte header.

"OUTPFORM;" (Output Formatted Data Array from Active Channel)
Outputs the formatted trace data from the active channel in current display units.

The header for the trace is read into the header- and length-integer variables. A data array is allocated to the size of the length statement. Using the length variable allows dynamic trace-array allocation. This can be very useful when using varying-length traces in your application.

The trace data is then read into the controller. The array size as well as the number of bytes contained in the array are displayed on the controller’s display console.

The data points are then displayed on the controller’s display.
"MARKDISC;" (Select Discrete-Marker Mode)
Sets the analyzer to display discrete-data points with the marker. In discrete-marker mode, the
trace data is not interpolated between points. This allows the operator to easily identify and
view the datapoints in the trace.

"MARK1 3E+4;" (Select Active Marker 1 and position at 30 kHz)
Activates marker 1 and positions it at 30 KHz (the left-hand edge of the display).

"OPC?;WAIT;" (Wait for Reply with Operation Complete Query)
The controller waits for the analyzer to return a one (1), signifying that the last command
received from the controller has completed. Upon reading the one (1), the analyzer is
addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on
the HP-IB.

"Position the marker with the knob and compare values" (Operator Prompt)
The operator is prompted to position the marker with the RPG knob along the points in the
trace and the program ends.

Running the Program
Run the program. The computer displays the number of elements and bytes associated with the
transfer of the trace, as well as the first 10 data points. Position the marker and examine the
data values. Compare the received values with the analyzer's marker values.

Example 3D: Data Transfer Using Frequency Array Information

Note
This program is stored as EXAMP3D on the “HP 8753D/Option 011
Programming Examples” disk received with the network analyzer.

This example explains how to use the limit-test array to read the corresponding frequency
values for the completed trace array into the controller. The analyzer is set to sweep from 10
MHz to 200 MHz in log-frequency mode with the number of points in the trace set to 11. This
makes it very difficult to compute the frequency-point spacing in the trace. The points are
equally spaced across the trace, but not equally spaced in relation to frequency (because the
frequency span is displayed in a logarithmic scale, as opposed to a linear scale). The limit-test
data array may be read from the analyzer to provide the frequency values for each data point.
Four values are read for each data point on the analyzer. The test results and limit values
are not used in this example. Only the frequency values are used. This technique is the only
method of obtaining the non-linear frequency data from the analyzer display. The test data
and frequencies are printed on the controller display and the marker is enabled to allow the
operator to examine the actual locations on the analyzer display.
The following is an outline of the program's processing sequence:
- The system is initialized.
- The integer variables for the header information are defined.
- The number of points in the trace is set to 11.
- The frequency span (10 MHz to 200 MHz) is selected.
- The log-frequency sweep is selected.
- The data-transfer format 3 is set.
- The headers are read from the trace.
- The array size is calculated and allocated.
- The trace data is read in.
- The limit-test array is calculated and allocated.
- The limit-line test array is read in.
- The table header is printed.
- The program cycles through the trace values.
- The trace data and frequency are printed.
- The discrete-marker mode is activated.
- The marker is activated and placed at 10 MHz.
- The instrument is returned to local control and the program ends.

**Tutorial Program Explanation**

The following section explains "Example 3D: Data Transfer Using Frequency Array Information" at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer's remote commands, see "Command Reference."

**Initializing the System**

**IFC (HP-IB Meta-Message: Interface Clear)**

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

"OPC?;PRES;" (Instrument Preset with Operation Complete Query)

Resets the network analyzer's operating state to the default state (preset). This command is similar to pressing (Preset) on the front panel, though it does not initiate an analyzer self-test. An operation-complete query (OPC?) asks for the analyzer to return an ASCII character one (1) to the system controller when the preset operation is complete.
**Main Program**

First, two integers are defined to hold the header information. Since an integer takes two bytes, the header and length variables will form the entire four-byte header.

The program variable Numpoints is defined as 11. The value "11" will be used in the next command line to set the number of points in the trace.

The program command line "POIN";Numpoints;";" selects the number of points in the trace using the value (11) for the variable Numpoints. This operation is virtually identical to executing a "POIN 11;" command used in Example 3B. Both operations set the number of points in the trace to 11.

"POIN 11;" (Edit Segment Points {11})

Sets the number of points in the trace to 11.

"STAR 10.E+6;" (Select Start Frequency as 10 MHz)

A start frequency of 10 MHz is selected.

"STOP 200.E+6;" (Select Stop Frequency as 200 MHz)

A stop frequency of 200 MHz is selected. This frequency range allows 2 cycles of log frequency scaling on the display.

"LOGFREQ;" (Select Log Frequency Sweep)

Selects a log frequency sweep.

"OPC?;SING;" (Select Single-Sweep Mode with Operation Complete Query)

Places the analyzer in single-sweep mode and activates a sweep. The analyzer returns a one (1) at the completion of the sweep.

"FORM3;" (Select Array-Data Format 3)

Selects array-data format 3. Data will now be output to the controller formatted as a 64-bit floating point number, with a four byte header.

"OUTP3;" (Output Formatted Data Array from Active Channel)

Outputs the formatted trace data from the active channel in current display units.

The header for the trace is read into the header- and length-integer variables. A data array is allocated to the size of the length statement. Using the length variable allows dynamic trace array allocation, which is very useful when using varying-length traces in your application.

The trace data is then input to the controller.

Next, a limit-line-results array is allocated in the controller.
"OUTPLML;" (Output Limit-Line Test Results)
Outputs the limit test results for each stimulus point. The results consist of four numbers. The first number represents the stimulus value tested. The second represents the test result: -1 for no test, 0 for fail, 1 for pass. The third number is the upper limit value, and the fourth is the lower limit value.

One shortcoming of this technique is that the array data is transferred in FORM 4, ASCII format. Although all the values are transferred, this example is only concerned with the frequency values.

As the program continues, a table heading with columns titled “Freq (MHz)” and “Mag (dB)” is printed on the controller’s display. The frequency and data-point values are cycled through and the resulting values are printed beneath their respective column headings on the controller display.

"MARKDISC;" (Select Discrete-Marker Mode)
Sets the analyzer to display discrete-data points with the marker. In discrete-marker mode, the trace data is not interpolated between points. This allows the operator to easily identify and view the datapoints in the trace.

"MARK1 10.E+6;" (Select Active Marker 1 and position at 10 MHz)
Activates marker 1 and positions it at 10 MHz (at the left side of the analyzer display).

"OPC?;WAIT;" (Wait for Reply with Operation Complete Query)
The controller waits for the analyzer to return a one (1), signifying that the last command received from the controller has completed. Upon reading the one (1), the analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB.

"Position the marker with the knob and compare values" (Operator Prompt)
The operator is prompted to position the marker with the RPG knob along the points in the trace and the program ends.

Running the Program
Run the program. Observe the controller display. The corresponding frequency values are shown with the trace-data values. Position the marker and observe the relationship between the frequency values and the point spacing on the trace. Compare the trace-data values on the analyzer with those received by the controller.
Example 3E: Data Transfer Using FORM 1 (Internal Binary Format)

Note
This program is stored as EXAMP3E on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

FORM 1 is used for rapid I/O transfer of analyzer data. It contains the least number of bytes-per-trace and does not require re-formatting in the analyzer. This format is more difficult to convert into a numeric array in the controller. Analyzer-state information, such as learn strings and calibration arrays, may be easily transferred in this format because data conversion is not required. Recalling an instrument state that has been stored in a file and transferring instrument-state information to the analyzer are excellent applications of a FORM 1 data transfer.

The following is an outline of the program’s processing sequence:

- The system is initialized.
- The integer variables for the header information are defined.
- The string variable for the header is defined.
- The selected frequency span is swept once.
- The internal-binary format is selected.
- The error-corrected data is output from the analyzer.
- The two data-header characters and the two length bytes are read in.
- The string buffer is allocated for data.
- The trace data is read into the string buffer.
- The analyzer is restored to continuous-sweep mode and queried for command completion.
- The instrument is returned to local control and the program ends.

Tutorial Program Explanation

The following section explains “Example 3E: Data Transfer Using FORM 1 (Internal Binary Format)” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”

Initializing the System

IFC (HP-IB Meta-Message: Interface Clear)

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

SDC (HP-IB Meta-Message: Selected Device Clear)

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the Instrument settings are reset to a power-on condition.
"OPC?;PRES;" (Instrument Preset with Operation Complete Query)
 Resets the network analyzer’s operating state to the default state (preset). This command is similar to pressing \texttt{PRES} on the front panel, though it does not initiate an analyzer self-test. An operation-complete query (OPC?) asks for the analyzer to return an ASCII character one (1) to the system controller when the preset operation is complete.

**Main Program**
 First, two integers are defined to hold the header information. Since an integer takes two bytes, the header and length variables will form the entire four-byte header.

"OPC?;SING;" (Select Single-Sweep Mode with Operation Complete Query)
 Places the analyzer in single-sweep mode and activates a sweep. The analyzer returns a one (1) at the completion of the sweep.

"FORM1;" (Select Array-Data Format 1)
 Selects array-data format 1. Data will now be output to the controller formatted in the analyzer internal format, with header.

"OUTPDATA;" (Output Error-Corrected Data)
 Outputs the error-corrected data from the active channel in real/imaginary pairs.
 In this case, the data is not converted to trace offsets and conversions. The trace data is read as error-corrected only. See Figure 2-3 to review the trace data processing chain. The first two bytes of the header are read as a string and placed in the variable header. The next two bytes are read into an integer variable which is the length of the data block to follow. The header value and the number of bytes are displayed on the controller console. The block is allocated and read in as a long string. Once the transfer is complete, the length of the data string received is compared to the header value.

"CONT;" (Select Continuous-Sweep Mode)
 The analyzer is restored to continuous-sweep mode.

"OPC?;WAIT;" (Wait for Reply with Operation Complete Query)
 The controller waits for the analyzer to return a one (1), signifying that the last command received from the controller has completed. Upon reading the one (1), the analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.

**Running the Program**
 Run the program. The analyzer is initialized. The header and the number of bytes in the block transfer are printed on the controller display. Once the transfer is complete, the number of bytes in the data string is printed. Compare the two numbers to be sure that the transfer was completed.
Example 4: Measurement Process Synchronization

Figure 2-5. Status Reporting Structure

Status Reporting

The analyzer has a status reporting mechanism, illustrated in Figure 2-5, that provides information about specific analyzer functions and events. The status byte is an 8-bit register with each bit summarizing the state of one aspect of the instrument. For example, the error queue summary bit will always be set if there are any errors in the queue. The value of the status byte can be read with the HP-IB serial poll operation. This command does not automatically put the instrument in remote mode, which gives you access to the analyzer front-panel functions while reading the status byte. The status byte can also be read by sending the command OUTSTAT. Reading the status byte does not affect its value.

The status byte summarizes the error queue, as mentioned before. It also summarizes two event-status registers that monitor specific conditions inside the instrument. The status byte also has a bit (6) that is set when the instrument is issuing a service request over HP-IB, and
a bit (0) that is set in the event-status register when the analyzer has data to send out over HP-IB. See “The Event-Status Registers,” located earlier in this chapter, for a discussion of the event-status registers.

Example 4A: Using the Error Queue

Note
This program is stored as EXAMP4A on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

The error queue holds up to 20 instrument errors and warnings in the order that they occurred. Each time the analyzer detects an error condition, the analyzer displays a message on the CRT, and puts the error in the error queue. If there are any errors in the queue, bit 3 of the status byte will be set. The errors can be read from the queue with the OUTPERRO command. OUTPERRO causes the analyzer to transmit the error number and message of the oldest error in the queue.

The following is an outline of the program’s processing sequence:

- The system is initialized.
- The error-message string is allocated.
- The analyzer is released from remote control.
- The program begins an endless loop to read the error queue.
- The status byte is read with a serial poll.
- The program tests to see if an error is present in the queue.
- The error-queue bit is set.
- The program requests the contents of the error queue.
- The error number and string are read.
- The error messages are printed until there are no more errors in the queue.
- The instrument is returned to local control.
- The controller emits a beep to attract the attention of the operator and resumes searching for errors.

Tutorial Program Explanation

The following section explains “Example 4A: Using the Error Queue” at the instrument-command level. Individual commands used in the program and their effects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”

Initializing the System

IFC (HP-IB Meta-Message: Interface Clear)

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

SDC (HP-IB Meta-Message: Selected Device Clear)

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments
may be affected differently by this command. In many cases, the instrument settings are reset
to a power-on condition.

"OPC?;PRES;" (Instrument Preset with Operation Complete Query)

Resets the network analyzer’s operating state to the default state (preset). This command is
similar to pressing [PRES] on the front panel, though it does not initiate an analyzer self-test.
An operation-complete query (OPC?) asks for the analyzer to return an ASCII character one (1)
to the system controller when the preset operation is complete.

Main Program

An error-message string variable is defined to contain the error messages from the analyzer.
Next, the analyzer is returned to local control. The program then reads the error queue in an
endless loop waiting to detect an error message.

The status byte for the analyzer is read by the serial-poll command until it finds that the
error-queue summary bit is set. Serial poll is an HP-IB function dedicated specifically to reading
the status byte of an instrument quickly, and does not cause the analyzer to go into remote
operation. If the bit is not set, the program will loop until the error bit gets set.

Once the bit is set (the error queue contains an error), the analyzer is instructed to output the
error number and the error message.

"OUTPERRO;" (Output Error Queue Contents)

Outputs the oldest error in the error queue. First, the error number is transmitted, then the
error message, both in ASCII format. This data string returns the analyzer to remote mode and
the controller then reads the error message(s) from the analyzer. The analyzer returns the error
number (2) and the error message(s) that appear(s) on the analyzer display. The error queue
is read until the error number is zero, which states that there are no more errors in the error
queue.

These values are read into the controller and displayed on the controller console.

The analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To
Local (GTL) on the HP-IB. The front-panel functions are now accessible and the controller gives
an audible signal that there is a problem.

Afterwards, the program returns to looping and reading the status byte to look for another
error message.

Running the Program

Run the program. The analyzer goes through the preset cycle. Nothing will happen at first.
The program is waiting for an error condition to activate the error queue. To cause an error,
press a blank softkey. The message CAUTION: INVALID KEY will appear on the analyzer’s
display. The computer will beep and print two error messages. The first line will be the
invalid key error message, and the second line will be the NO ERRORS message. To clear the
error queue, you can either loop until the NO ERRORS message is received, or until the bit in
the status register is cleared. In this case, we wait until the status bit in the status register is
clear. Note that while the program is running, the analyzer remains in the local mode and the
front-panel keys may be accessed.

The error queue will hold up to 20 errors until all the errors are read out or the instrument is
preset. It is important to clear the error queue whenever errors are detected. Otherwise, old
errors may be mistakenly associated with the current instrument state.
Press `SYSTEM` and then the unlabeled key several times quickly and watch the display. The number of errors observed should correspond to the number of times you pressed the key.

As another example, press `CAL`. Then the `CALIBRATE MENU` key. Select the `RESPONSE` calibration. Press `DONE: RESPONSE` without performing any calibrations. Note the error message on the analyzer and on the controller display. Push the `THRU` key and then `DONE: RESPONSE`. We are not concerned with the validity of the calibration, just setting a simple calibration on the analyzer. Note that COR is displayed in the upper left-hand section of the graticule. Now press `START` and `[]`. This will generate an error because the start frequency has been changed, invalidating the calibration. This error is reported on the controller display as well. A complete list of error messages and their descriptions can be found in “Error Messages” of the HP 8753D/Option 011 Network Analyzer User’s Guide.

The program is in an infinite loop waiting for errors to occur. End the program by pressing `RESET` or `BREAK` on the controller keyboard.

**Note** Not all messages displayed by the analyzer are put in the error queue: operator prompts and cautions are not included.

---

**Example 4B: Generating Interrupts**

**Note** This program is stored as EXAMP4B on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

Interrupts can be generated using the status-reporting mechanism. The status-byte bits can be enabled to generate a service request (SRQ) when set. In turn, the instrument controller can be set up to generate an interrupt on the SRQ and respond to the condition which caused the SRQ.

To generate an SRQ, a bit in the status byte is enabled using the `SREn;` command. A one (1) in a bit position enables that bit in the status byte. Hence, executing `SRE 8;` enables an SRQ on bit 3, the check-error queue, since the decimal value 8 equals 00001000 in binary representation. Whenever an error is put into the error queue and bit 3 is set, the SRQ line is asserted, illuminating the (S) indicator in the HP-IB status block on the front panel of the analyzer. The only way to clear the SRQ is to disable bit 3, re-enable bit 3, or read out all the errors from the queue.

A bit in the event-status register can be enabled so that it is summarized by bit 5 of the status byte. If any enabled bit in the event-status register is set, bit 5 of the status byte will also be set. For example, executing `ESE 66;` enables bits 1 and 6 of the event-status register, since in binary, the decimal number 66 equals 01000100. Hence, whenever active control is requested or a front-panel key is pressed, bit 5 of the status byte will be set. Similarly, executing `ESnIBn;` enables bits in event-status-register B so that they will be summarized by bit 2 in the status byte.

To generate an SRQ from an event-status register, enable the desired event-status-register bit. Then enable the status byte to generate an SRQ. For instance, executing `ESE 32;SRE 32;` enables the syntax-error bit. When the syntax-error bit is set, the summary bit in the status byte will be set. This will, in turn, enable an SRQ on bit 5 of the status byte, the summary bit for the event-status register.
The following is an outline of the program's processing sequence:

- The system is initialized.
- The status registers are cleared.
- The event-status register bit 5 is enabled.
- The status-register bit 5 is enabled.
- The interrupt pointer is enabled and points to a subroutine.
- A bad command is sent to the analyzer to generate errors.
- The controller reads a serial-poll byte from HP-IB in the event of an interrupt.
- The program tests for an SRQ.
- If the SRQ is not generated by the analyzer, the subroutine stops and displays SRQ FROM OTHER DEVICE.
- If the SRQ was generated by the analyzer, the program reads the status byte and event-status register.
- If bit 5 in the event-status register is set, program prints: SYNTAX ERROR FROM ANALYZER.
- If bit 5 in the event-status register is NOT set, program prints: SYNTAX ERROR BIT NOT SET.
- The SRQ interrupt is re-enabled on the bus.
- At the finish, the interrupt is deactivated.
- The analyzer is released from remote control and the program ends.

**Tutorial Program Explanation**

The following section explains "Example 4B: Generating Interrupts" at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer's remote commands, see "Command Reference."

**Initializing the System**

**IFC (HP-IB Meta-Message: Interface Clear)**

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

"OPC?;PRES;" **(Instrument Preset with Operation Complete Query)**

Resets the network analyzer's operating state to the default state (preset). This command is similar to pressing [Preset] on the front panel, though it does not initiate an analyzer self-test. An operation-complete query (OPC?) asks for the analyzer to return an ASCII character one (1) to the system controller when the preset operation is complete.
Main Program

"CLES;" (Clear Status)
Cleans the status register, the event-status registers, and the enable registers.

"ESE 32;" (Event Status Enable {Bit 5 of the Event-Status Register})
Enables bit 5 of the event-status register.

"SRE 32;" (Service Request Enable {Bit 5 of the Status Byte})
Service request enable. Enables bit 5 of the status byte so that an SRQ is generated when a syntax error is detected. Upon the detection of SRQ, the subroutine is executed.
The degree of testing depends upon the number of operations enabled to generate an SRQ. The check-error queue is always set as a consequence of the error occurring. On a complex system with other devices and multiple bits enabled, the SRQ-handler routine can become quite complex in order to test all possible conditions.
The SRQ can be cleared by reading the event-status register and clearing the latched bit, or by clearing the enable registers with the CLES; command. The syntax-error message on the analyzer display can only be cleared by the HP-IB Device Clear (DCL) message or Selected Device Clear (SDC) message. Device Clear is not commonly used because it clears every device on the bus. Selected Device Clear can be used to reset the input and output queue and the registers, but will also clear all the interrupt definitions.
The controller issues a serial poll to read and clear any pending service requests. The program then sets an interrupt on the HP-IB bit 2 (SRQ).

"Waiting for bad syntax" (Program Comment)
Displays "Waiting for bad syntax" on the controller’s CRT while the program pauses for 2 seconds.

"STIP 2GHZ;" (Incorrect Analyzer Stop Frequency Command)
This incorrect command generates a syntax error. The interrupt causes the program to branch to the interrupt routine Srq_det.

Subroutine: Srq_det:
If there is more than one instrument on the bus capable of generating an SRQ, it is necessary to use serial poll to determine which device has issued the SRQ. Each serial poll returns a status byte from the instrument. Bit 6 is set in the status byte of the instrument that caused the request. A branch to the SRQ-detection routine disables the interrupt assignment. Therefore, the return from Srq_det must re-enable the controller interrupt so it can detect another SRQ interrupt.
If the analyzer’s status-byte bit 6 is set (the status-byte-SRQ bit enabled) the following comment is displayed on the controller’s console:

"SRQ received from analyzer" (Program Comment)
Acknowledges on the controller display console that an SRQ was received from the analyzer if bit 6 is set in the status byte.
"SRQ from other device" *(Program Comment)*

Acknowledges on the controller display console that an SRQ was not generated by the analyzer if bit 6 is not set in the status byte. If this is the case, the program halts the reading of the analyzer's status byte and continues looking for the SRQ on the bus.

"ESR?;" *(Event-Status Register Query)*

The controller uses this command to query the event-status register.
The controller then reads the event-status register with a serial poll.

"Event Status Register caused SRQ" *(Program Comment)*

If bit 5 of the status byte is set, this message is displayed on the controller's CRT.

"Some other bit caused SRQ" *(Program Comment)*

If bit 5 of the status byte is not set, this message is displayed on the controller's CRT.

"ESE?;" *(Event-status register Query)*

Queries the event-status register. The controller then reads the status-register byte with another serial poll.

"Syntax error from analyzer" *(Program Comment)*

If bit 5 of the event-status register byte is set, this message is displayed on the controller's CRT.

"Syntax error bit not set" *(Program Comment)*

If bit 5 of the event-status register byte is not set, this message is displayed on the controller's CRT.

To continue operation with SRQ detection enabled, the operator must re-enable the interrupt and then return to normal programming operation. At this time, the subroutine ends.

The analyzer will display CAUTION: SYNTAX ERROR and the incorrect command, pointing to the first character it did not understand.

The program displays FINISHED, switches OFF the interrupt, addresses the analyzer and returns it to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.

**Running the Program**

Run the program. The computer will preset the analyzer, then pause for a second or two. After pausing, the program sends an invalid command string "STIP 2GHz;" to cause a syntax error. This command is intended to be "STUP 2 GHz;". The computer will display a series of messages from the SRQ-handler routine. The analyzer will display CAUTION: SYNTAX ERROR and the incorrect command, pointing to the first character it did not understand.

The SRQ can be cleared by reading the event-status register and clearing the latched bit, or by clearing the enable registers with the CLES; command. The syntax-error message on the analyzer display can only be cleared by the HP-IB Device Clear (DCL) message or Selected Device Clear (SDC) message. Device Clear is not commonly used because it clears every device on the bus. Selected Device Clear can be used to reset the input and output queue and the registers of a specific instrument on the bus. This will also clear all the interrupt definitions.
Example 4C: Power Meter Calibration

Note
This program is stored as EXAMP4C on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

For increased accuracy of the analyzer’s PORT 1-output power, a power meter calibration is available. This measurement-accuracy enhancement technique is described in “Error Messages” of the HP 8753D/Option 011 Network Analyzer User’s Guide. The example described will perform the sample and sweep calibration under HP-IB remote control.

The power meter is usually connected to PORT 1 for the forward measurements. Its address must be set correctly and it must be connected to the HP-IB. The power meter address can be set by pressing: LOCAL SET ADDRESSES ADDRESS P MTR/HP-IB and using the and keys or the numeric key pad to complete the process. The appropriate command must be selected for the model number of power meter being used. Press POWER MTR until the model being used is displayed between the brackets. The correction factors for the power sensor are entered into the analyzer. All of these steps are explained in “Error Messages” of the HP 8753D/Option 011 Network Analyzer User’s Guide.

The number of readings-per-point must also be selected before starting. The number of points directly affects the measurement time of the calibration sequence. The power meter must interact with the analyzer for each of the selected points and read the number of values specified for each trace point. Typically, two readings-per-point is considered appropriate. More than two readings-per-point could lead to unacceptable processing time.

To control a power meter calibration via HP-IB, the analyzer must be set to pass-control mode. The analyzer must position the local oscillator to a point in the sweep and read the power present at the power meter sensor. For this operation to take place, the system controller must set up the measurement and then pass control to the analyzer to read each data point in the sweep. After reading the data point from the power meter, the analyzer passes control back to the system controller. The analyzer then sets up to measure the next point and again requests control from the system controller. This process continues until the analyzer signals that the entire sweep has been measured point-by-point.

The following is an outline of the program’s processing sequence:

- The system is initialized.
- The number of points in the trace is set.
- The number of readings-per-point is set.
- The frequency span is set.
- The reference channel is measured.
- The power meter-calibration array is allocated.
- The power meter model is chosen.
- The status registers are cleared.
- The request-control summary bit is enabled.
- The pass-control mode is enabled.
- A calibration sweep is taken to begin the sequence.
- The status byte is read until control is requested.
- The computer passes control to the analyzer.
- The display is cleared and the analyzer is set to talker/listener mode.
- The HP-IB interface status is read until control is returned.
- The program loops until all the points have been measured.
- The power meter calibration is enabled.
- The calibration data is output to the controller in FORM 4, ASCII format.
- The power meter-calibration factors are read into the controller.
- The analyzer is released from remote control and the program ends.

**Tutorial Program Explanation**

The following section explains “Example 4C: Power Meter Calibration” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”

**Initializing the System**

**IFC (HP-IB Meta-Message: Interface Clear)**

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

"0PC?;PRES;" (Instrument Preset with Operation Complete Query)

Resets the network analyzer’s operating state to the default state (preset). This command is similar to pressing (PRES) on the front panel, though it does not initiate an analyzer self-test. An operation-complete query (OPC?) asks for the analyzer to return an ASCII character one (1) to the system controller when the preset operation is complete.
**Main Program**

The controller must set up the measurement parameters and then pass control to the analyzer so the analyzer can read the power at each measurement point.

For this operation to take place, the controller must set up the measurement and then pass control to the analyzer to read in each point in the sweep. After the analyzer reads the power, the analyzer passes control back to the controller. The analyzer sets up the next measurement parameters and requests HP-IB control from the controller. This process continues until the analyzer signals that all the data points in the entire sweep have been measured.

If control is passed before requested, the analyzer will just ignore it and return control to the controller. If control is passed while the analyzer is in control, the pass is ignored. The program accomplishes this pass control synchronization by using the event-status registers to signal the state of the analyzer. The program monitors the analyzer’s status byte by performing a serial poll and reading the status byte.

The program sets the value of the variables by defining:

- the serial poll as an integer
- the number of points in the trace as 11
- the number of reading-per-point as 2
- the start frequency as 10 MHz
- the stop frequency as 50 MHz

The controller then uses these variables to set the analyzer’s instrument state.

"MEASR;" (Select and Measure Input R)

Once the instrument state has been set, the program selects and measures Input R.

At this point in the program, the power meter calibration array is created.

"DATI;" (Store Data in Channel Memory)

Stores the trace data in the active-channel memory array.

"DISPDATM;" (Display Data and Memory)

Displays the combinations of active data and stored trace memory on the active channel.

"AUTO;" (Autoscale Selected Channel)

Autoscales the active channel.

"POWMOFF;DEBUON;" (Deactivate Power Meter Menu/Activate Debug Mode)

Selects the HP 437B/438A power meter and activates the debug-mode display. This debug display shows the control transfers.

"CLES;" (Clear Status)

Clears the status register, the event-status registers, and the enable registers.
"ESE2;" (Event-Status Enable (Control-Summary Bit 2))
Enables the request-control bit when the analyzer is ready to perform a power meter measurement over the HP-IB. This is part of the pass-control operation synchronization. This bit is enabled in the event-status register as bit 1, decimal 2.

When this bit is set, the event-status-register bit is set in the analyzer-status register as bit 5, decimal 32.

Bit 5, decimal 32 will indicate that a request for pass control is active.

"SRE 32;" (Service Request Enable (Bit 5 of the Status Byte))
Service request enable. Enables bit 5 of the status byte so that an SRQ is generated when the pass-control bit is detected.

"Beginning Power Meter Cal" (Program Comment)
Displays the message "Beginning Power Meter Cal" on the controller’s Display console.

"USEPASC;" (Enable Pass-Control Mode)
Enables pass control mode. The HP-IB control must be in pass-control mode for the analyzer to control the power meter calibration.

"TAKCS;" (Take Calibration Sweep)
Initiates a power meter calibration sweep at the number of points for the number of readings plus one, as defined at the beginning of the program. Repeat when the controller detects a service request by issuing a serial poll.

"Waiting for request" (Program Message)
The message "Waiting for request" is displayed on the controller’s CRT. The controller continues to read the status byte with a serial poll until bit 6 is enabled on the status byte.

An interrupt is then set on the HP-IB status byte bit (SRQ).

"ESR?;" (Event-Status-Register Query)
Queries the analyzer’s event-status register to clear the serial poll. The value returned is read by the controller and discarded.

"Passing Control" (Program Message)
This message is displayed on the controller’s CRT as control is passed to the analyzer. The controller’s HP-IB interface register is read and tested until control is returned from the analyzer.

"Waiting for request" (Program Message)
This message is displayed as the system control continues to read the event-status register while waiting to take control back from the analyzer. Reading the event-status register does not interact with the analyzer’s operation. When bit 6 in the event-status-register byte is set, control is returned to the system controller.
"Finished with Power meter Cal" (Program Message)

This message is displayed on the controller's console at the end of the power meter calibration process. The display is then cleared.

"TALKLIST;" (Select Talker/Listener Mode)

Enables the talker/listener mode in the analyzer.

"CLE:\" (Clear Status)

Clears the status register, the event-status registers, and the enable registers.

"PWMCONES;" (Activate Power Meter Calibration Sweep for One Sweep)

Activates a power meter calibration sweep after taking one calibration sweep. At this time, the correction factors are applied to the data.

"FORM4;" (Select Array-Data Format 4)

Sets format 4, ASCII floating-point format to be used in reading the power meter calibration correction factors.

"OUTPPMCAL1;" (Output Channel 1 Power Meter Calibration Array)

Outputs the channel 1 power meter calibration array. The power meter calibration factors are then read by the controller. The power meter calibration factors are cycled through and printed on the controller's display.

The analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.

Running the Program

The analyzer is preset and the power meter-calibration routine begins. The analyzer displays the message "WAITING FOR HP-IB CONTROL" when it is requesting control. The system controller display prints "Passing Control" when control is passed to the analyzer. The controller displays "Waiting for request" while the analyzer has control and is reading the power meter.

The interaction of the messages and the movement of the cursor allow observation of the calibration process. Once the calibration is complete, the analyzer displays "POWER METER CAL IS COMPLETE" and the system controller displays "Finished with Power meter Cal".

The power meter-calibration mode (with one sweep of correction data) is enabled and the calibration is switched ON. At the completion of the program, talker/listener mode is restored, the event-status registers are cleared (to halt the status-byte interaction), the power meter correction factors are displayed, the sweep is placed in continuous-sweep mode, the analyzer is released from HP-IB control, and the program ends.
Example 5: Network Analyzer System Setups
This section provides several different examples of performing analyzer system setups.

Storing and Recalling Instrument States

Note  The most efficient option for storing and recalling analyzer states is using the analyzer’s internal registers to save the CAL data. Recalling these registers is the fastest solution to restoring analyzer setups. See “Printing, Plotting, or Saving Measurement Results” in the HP 8753D/Option 011 Network Analyzer User’s Guide for detailed information on the analyzer’s internal storage registers.

In the event that all the registers have been used, the internal disk drive is not used, or if internal memory limitations exist, then these external solutions become viable.

The purpose of this example is to demonstrate several programming options for storing and recalling entire instrument states over HP-IB. The examples describe two different processes for storing and recalling instrument states. The first example accomplishes the task using the learn string. The second example involves reading both the learn string and the calibration arrays out of the analyzer and storing them to disk or storing them in the system controller itself.

Using the learn string is a very rapid way of saving the instrument state, but using direct disk access has the advantage of automatically storing calibrations, cal kits, and data along with the instrument state.

A complete analyzer setup requires sending the learn string and a calibration array to set the analyzer parameters. The CAL array may also be placed in the analyzer, just as if a calibration was performed. By sending both sets of data, the analyzer may be quickly setup for a measurement.

Several different measurements may be required in the course of testing a device. An efficient way of performing multiple measurements is to send both the calibration array and the learn string, and then perform the measurements.

Example 5A: Using the Learn String

Note  This program is stored as EXAMP5A on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

The learn string is a very fast and easy way to read an instrument state. The learn string includes all front-panel settings, the limit table for each channel, and the list-frequency table. It can be read out of the analyzer with the command OUTPLEAS, and input to the analyzer with the command INPULEAS. This array is always transmitted in FORM 1, the internal format for the analyzer. It cannot be longer than 3000 bytes. The example for a FORM 1 transfer could also have been used. However, Example 5A is the simplest solution for reading the learn string from the analyzer.

The following is an outline of the program’s processing sequence:

- The system is initialized.
- The string storage is allocated.
- The learn string is requested.

2-74  HP-IB Programming Reference
The string is read without any processing.

The analyzer is released from remote control.

The instrument state is changed by the operator.

The learn string is sent back to the analyzer.

The analyzer is released from remote control and the program ends.

**Tutorial Program Explanation**

The following section explains "Example 5A: Using the Learn String" at the instrument-command level. Individual commands used in the program and their effects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”

**Initializing the System**

**IFC (HP-IB Meta-Message: Interface Clear)**

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

**Main Program**

First, a string is defined for contents to receive the instrument state learn-string data.

"OUTPLEAS;" (Output Learn String)

Outputs the learn string. The learn string is in binary data and must not be interpreted by the input process. The EOI terminator is sent with the last byte in the string and is used to terminate the entry. The conventional terminators (carriage return or line feed) may appear in the string and must be treated as data.

The controller releases the analyzer from HP-IB control.

"Change state and press ENTER" (Operator Prompt)

Prompts the operator to change the analyzer settings after the analyzer has received the learn string. The analyzer is placed in local mode to allow you to change the analyzer settings. If you press [PRES], the analyzer restores the settings to the default state. Pressing [ENTER] on the controller keyboard continues the program.

"INPULEAS;" (Input Learn String to Analyzer)

Prepares the analyzer to receive the learn string and then sends the learn string to the analyzer. Notice that no semicolon is used as a terminator. The command is terminated by the end of the learn string. The previous state of the analyzer is then restored with the learn string.
"OPC?;WAIT;" (Wait for Reply with Operation Complete Query)

The controller waits for the analyzer to return a one (1), signifying that the last command received from the controller has completed. Upon reading the one (1),

The analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.

Running the Program

Run the program. When the program stops, change the instrument state and press ENTER on the controller. The analyzer will be returned to its original state by sending the learn string to the analyzer.

Example 5B: Reading Calibration Data

This program is stored as EXAMP5B on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

This example demonstrates:

- how to read measurement calibration data out of the network analyzer
- how to read it back into the analyzer
- how to determine which calibration is active

The data used to perform measurement-error correction is stored inside the analyzer in one (or more) of twelve calibration-coefficient arrays. Each array is a specific error coefficient, and is stored and transmitted as an error-corrected data array. Each point is a real/imaginary pair, and the number of points in the array is the same as the number of points in the trace. The five array-data formats also apply to the transfer of calibration-coefficient arrays. "Printing, Plotting, or Saving Measurement Results" in the HP 8753D/Option 011 Network Analyzer User's Guide contains information on the storage locations for calibration coefficients and different calibration types.

A computer can read out the error coefficients using the commands OUTPCALC01, OUTPCALC02, ... through OUTPCALC12. Each calibration type uses only as many arrays as required, beginning with array 1. Hence, it is necessary to know the type of calibration about to be read out: attempting to read an array not being used in the current calibration causes the "REQUESTED DATA NOT CURRENTLY AVAILABLE" warning.

A computer can also store calibration coefficients in the analyzer. To do this, declare the type of calibration data about to be stored in the analyzer just as if you were about to perform that calibration. Then, instead of calling up different classes, transfer the calibration coefficients using the INPUCALCmn commands. The variables mn are a data pair appended to the command representing a calibration number from 01 through 12. When all the coefficients are stored in the analyzer, activate the calibration by issuing the mnemonic SAVC:, and trigger a sweep on the analyzer.

This example reads the calibration coefficients into a very large array, from which they can be examined, modified, stored, or put back into the instrument. If the data is to be directly stored on to disk, it is usually more efficient to use FORM 1 (analyzer's internal-binary format), and to store each coefficient array as it is read in.

The following is an outline of the program’s processing sequence:

- The system is initialized.
- The calibration types and number of arrays are defined.
The integer variables for reading the headers are defined.

The calibration type and number of arrays are read by the controller.

The output is formatted in FORM 3.

The number of points in the trace is read.

The memory is allocated for the calibration arrays.

Each calibration array is requested from the analyzer.

The elements from each calibration array are read in.

The next calibration array is requested until all the arrays have been read.

The calibration type is sent to the analyzer.

Each calibration array is sent.

The calibration is activated.

The analyzer is released from remote control and the program ends.

**Tutorial Program Explanation**

The following section explains "Example 5B: Reading Calibration Data" at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer's remote commands, see "Command Reference."

**Initializing the System**

**IFC (HP-IB Meta-Message: Interface Clear)**

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

**Main Program**

Data defined for determining calibration types and numbers of arrays.

The analyzer responds with a one (1) if one of the following calibrations are active:

- "CALRESP", 1 which is a response calibration with one array
- "CALIRAI", 2 which is a response and isolation calibration with 2 arrays
- "CALIS111", 3 which is an input one-port measurement calibration with 3 arrays
- "CALIS221", 3 which is an output one-port measurement calibration with 3 arrays
- "CALIFUL2", 12 which is a full 2-port measurement calibration with 12 arrays

Caltype"?;" (Calibration Type Query) Each calibration type followed by a "?" is sent to the analyzer as a query. a one (1) is returned for the active calibration type.
If there is an active calibration, the program uses the number of arrays associated with the calibrations to query the active calibration present in the analyzer. If no calibration is found, the program ends. Once the calibration type is determined, the type and number of arrays is shown on the console display.

"FORM3;" (Select Array-Data Format 3)

Selects array-data format 3. Data will now be output to the controller formatted as a 64-bit floating point, with header.

"POIN?;" (Number of Points Query)

Queries the analyzer for the number of points in the measurement trace.

An array to store the calibration arrays is created.

Each of the calibration arrays are read from the controller with the command "OUTPCALCnn;", where nn is the calibration number 01 through 12. A leading zero (0) is required to generate a two-digit command.

The selected error coefficient array is output from the active channel. Each array is the same as a data array for use by the program to read in the values. The "I" variable must be in a fixed, two-digit form with a leading zero (if needed).

The required number of arrays for the active calibration are read from the analyzer and stored into the calibration array.

A 4-byte header, containing "#A" and two bytes which identify the length of the array is read by the program before putting the calibration-array data into the multi-dimensional array.

Each pair is read from the analyzer and the array is indexed for the next element in the array.

After all the calibration arrays have been read from the analyzer, the operator is prompted:

"PRESS RETURN TO RE-TRANSMIT CALIBRATION" (Operator Prompt)

| Note | If your system controller has an [ENTER] rather than a [RETURN], this function should be performed by pressing [ENTER]. |

Prompts the operator to send the calibration arrays of coefficients back to the analyzer. When [ENTER] is pressed the calibration coefficients are sent back to the analyzer.

"FORM3;" (Select Array-Data Format 3)

Selects array-data format 3. Data will now be output to the controller formatted as a 64-bit floating point, with header. This is the same format with which the data originally read.

Caltype;";" (Output Calibration Type)

Output the calibration type determined at the beginning of the program to the analyzer. Each calibration array is separately transmitted to the analyzer and "TRANSMITTING ARRAY :nn" (where nn represents the array number) is displayed on the controller’s screen.

The header and array length are sent to the analyzer along with each calibration array of real/imaginary pairs.

The program then moves to the next calibration array and performs the transmission functions again until all of the calibration arrays have been transmitted back to the analyzer.
"SAVC;" (Calibration-Array Transmission Complete)
Completes the transfer of error correction coefficients back into the instrument.
Activates the calibration after all the arrays have been indexed and sent to the analyzer.

"CONT;" (Select Continuous-Sweep Mode)
Sets the sweep mode to continuous.

"GPC?; WAIT;" (Wait for Reply with Operation Complete Query)
The controller waits for the analyzer to return a one (1), signifying that the last command
received from the controller has completed. Upon reading the one (1), the analyzer is
addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on
the HP-IB and the program ends.

Running the Program
Before executing the program, perform a calibration.

The program is able to detect which type of calibration is active. With that information, it
predicts how many arrays to read out. When all the arrays have been sent to the computer, the
program prompts the operator. The operator then switches OFF the calibration or performs
a completely different calibration on the analyzer and continues the program. The computer
reloads the old calibration. The operator should not preset the analyzer because the instrument
settings must be the same as those that were present when the calibration was taken.

Note
The re-transmitted calibration is associated with the current instrument state:
the instrument has no way of knowing the original state associated with the
 calibration data. For this reason, it is recommended that the learn string be
 used to store the instrument state whenever calibration data is stored. The
 next example demonstrates how to reload the analyzer state with both the
 learn string and the calibration arrays.

Example 5C: Saving and Restoring the Analyzer Instrument State

Note
This program is stored as EXAMP5C on the “HP 8753D/Option 011
Programming Examples” disk received with the network analyzer.

Note
The instrument state may also be stored in the analyzer's internal registers.
This is the fastest and most efficient method for toggling between instrument
states. This example is for use when the analyzer's internal memory is full, or
when there are other internal-memory limitations.

This example demonstrates using both the learn string and the calibration arrays to completely
re-program the analyzer state. If you were performing two entirely different measurements
on a device and wanted to quickly change between instrument states and perform the
measurements, this example program is a potential solution.

The example will request the learn string and a calibration array from the analyzer and store
them in a disk file on the system controller. Once the storage is complete, the operator will be
prompted to change the state of the analyzer and then re-load the state that was previously
stored in the disk file. Once the file is created on the disk, the state information can be
retrieved from the controller and restored on the analyzer.
Note

The disk file can only be created once. Errors will occur if the operator repeatedly tries to re-create the file.

For this example, only a thru response calibration will be performed and transferred. This means only one calibration array will be read from the analyzer and written to the disk file with the instrument state. To work with more elaborate calibrations, additional arrays will need to be defined and transferred to the disk file. This is not difficult, but requires some additional programming steps which were omitted in the interest of presenting a simple example.

The following is an outline of the program’s processing sequence:

- The integers for reading the headers are defined.
- The system is initialized.
- An array is created to hold the learn string.
- The learn string is requested by the controller.
- The number of points in the trace is read.
- The controller allocates an array for the calibration data.
- The calibration data is read into the controller.
- The controller creates and assigns a data file for the calibration array and the learn string.
- The learn string and calibration array are stored in the disk file.
- The operator presses [ENTER] on the controller to read the calibration data back into the analyzer.
- The learn string is read from the disk file and output to the analyzer.
- The calibration array is read in from the disk file and stored in the analyzer.
- The analyzer is returned to continuous-sweep mode.
- The analyzer is released from remote control and the program ends.

Tutorial Program Explanation

The following section explains “Example 5C: Saving and Restoring the Analyzer Instrument State” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”

The instrument state is going to be transferred in the form of a learn string and a calibration array. In the first section of the program, a string array is dimensioned and the learn string is read from the analyzer.

Initializing the System

IFC (HP-IB Meta-Message: Interface Clear)

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

SDC (HP-IB Meta-Message: Selected Device Clear)

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments
may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

**Main Program**

"DPC?;SING;" (Select Single-Sweep Mode with Operation Complete Query)
Places the analyzer in single-sweep mode and activates a sweep. The analyzer returns a one (1) at the completion of the sweep.

"OUTPLEAS;" (Output Learn String)
Outputs the learn string. A 3000-byte string is defined to receive the learn-string data. The learn string is in binary data and must not be interpreted by the input process. The EOI terminator is sent with the last byte in the string. EOI is used to terminate the entry. The conventional terminators (carriage return or line feed) may appear in the string and must be treated as data.

"POIN?;" (Number of Points Query)
Queries the analyzer for the number of points in the measurement trace.
The analyzer outputs the trace length to determine the size of the calibration array. An array was dimensioned to this length with two entries-per-point (the real and imaginary calibration values).

"FORM3;" (Select Array-Data Format 3)
Selects array-data format 3. Data will now be output to the controller formatted in IEEE 64-bit floating point format, with header.

"OUTPCALC01;" (Output Selected Calibration Array)
Outputs the selected error coefficient array from the active channel. Each array is the same as a data array.

Used here, this command queries the calibration array to read the first and only array. The array header and length bytes are read in as integers and the array read as floating-point numbers. Once the array and the learn string are stored in arrays, the next step is to transfer them to a disk file.

A file is created large enough to hold 3000 bytes of learn string and the number of points times 16 bytes-per-point. In this case, \[3000 \times (201 \times 16)\] or 6,216 bytes. The learn string, header and length integers and the calibration array are written to the file and the file closed.

"Cal data received. Press RETURN to send it back." (Operator Prompt)
Prompts the operator to press ENTER on the controller keyboard to reload the calibration arrays back into the analyzer.

A new learn-string variable is dimensioned to 3000 bytes. The data file is opened and the learn string is read along with the header and length integers. The length variable is necessary to dimension the array for the calibration array. The length integer divided by 16 (two 8-byte values-per-data pair) dimensions the array. The calibration data can now be read from the file into the array.

Pressing ENTER on the controller keyboard continues the program.
"INPULEAS" *(Input Learn String)*
Prepares the analyzer to receive the learn string and then sends the learn string to the analyzer.

"INPCALCO1;" *(Input Selected-Calibration Array)*
Sends the calibration data to define the location for the calibration array. The header and length integers are sent first, followed by the calibration array.

"OPC?;SAVC;" *(Cal-Array Transmission Done with Operation Complete Query)*
Completes the transfer of error-correction coefficients back into the instrument.
Activates the calibration after all the arrays have been indexed and sent to the analyzer. The analyzer returns a one (1) at the completion of the array transmission process.

"CONT;" *(Select Continuous-Sweep Mode)*
Sets the sweep mode to continuous.

"OPC?;WAIT;" *(Wait for Reply with Operation Complete Query)*
The controller waits for the analyzer to return a one (1), signifying that the last command received from the controller has completed. Upon reading the one (1), the analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.

**Running the Program**
Setup the analyzer and perform a through calibration.

Run the program. The program prompts the operator to change the state of the analyzer and then press **ENTER** to continue. At this point, the analyzer state is stored on the disk file in the controller. Pressing **ENTER** will begin the transfer from the disk file to internal arrays within the controller and then on to the analyzer. Once completed: the original state will be restored, the analyzer will be sweeping, the analyzer will be calibrated, and CUR will be displayed on the analyzer’s CRT.
Example 6: Limit-Line Testing

Using List-Frequency Mode

The analyzer normally takes data points spaced at regular intervals across the overall frequency range of the measurement. For example, for a 2 GHz frequency span using 201 points, data will be taken at intervals of 10 MHz. The list-frequency mode allows the operator to select the specific points, or frequency spacing between points, at which measurements are to be made. This mode of operation allows flexibility in setting up tests that Insure device performance in an efficient manner. By only sampling specific points, measurement time is reduced. List-frequency sweeps are also discussed in "Application and Operation Concepts" of the HP 8753D/Option 011 Network Analyzer User's Guide. These programs emulate operation from the analyzer's front panel when using list sweeps.

The following two examples illustrate the use of the analyzer's list-frequency mode to perform arbitrary frequency testing. Example 6A allows the operator to construct a table of list-frequency segments which is then loaded into the analyzer's list-frequency table. There are a maximum of 30 segments available. Each segment stipulates a start and stop frequency, and the number of data points to be taken over that frequency range. Example 6B lets the operator select a specific segment to "zoom-in." A single instrument can be programmed to measure several different devices, each with its own frequency range, using a single calibration. When a specific device is connected, the operator selects the appropriate segment for that device. Note that list-frequency segments can be overlapped, but the total number of points in all the segments must not exceed 1632.

Example 6A: Setting Up a List-Frequency Sweep

| Note | This program is stored as EXAMP6A on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer. |

The purpose of this example is to show how to create a list-frequency table and transmit it to the analyzer.

The command sequence for entering a list-frequency table imitates the key sequence followed when entering a table from the front panel; there is a command for every key-press.

Editing a segment is also similar same as the front-panel key sequence, except that the analyzer automatically reorders each edited segment in order of increasing start frequency.

The list-frequency table is also carried as part of the learn string. While the table cannot be modified as part of the learn string, it can be stored and recalled with very little effort by storing and recalling the learn string. See the "Data-Processing Chain" section in Chapter 2 for additional information details on using learn strings.

This example takes advantage of the computer's ability to simplify:

- creating a list-frequency table
- editing a list-frequency table

The table is entered and completely edited before being transmitted to the analyzer. To simplify the programming task, options such as entering center frequency, frequency span, or step size are not included.

The list-frequency information may be acquired using the limit-test results array. The actual stimulus points are available as the first element in the array.
The following is an outline of the program’s processing sequence:

- The system is initialized.
- The existing list frequencies are edited and cleared.
- The number of segments to be defined is read in.
- An array for the list segments is defined.
- The parameters for each segment are requested.
- If the operator wants to edit, the segment parameters are re-entered.
- The new list is sent to the analyzer.
- The analyzer is released from remote control and the program ends.

**Tutorial Program Explanation**

The following section explains “Example 6A: Setting Up a List-Frequency Sweep” at the instrument-command level. Individual commands used in the program and their effects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”

**Initializing the System**

**IFC (HP-IB Meta-Message: Interface Clear)**

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

"OPC?;PRES;" **(Instrument Preset with Operation Complete Query)**

Resets the network analyzer’s operating state to the default state (preset). This command is similar to pressing [Preset] on the front panel, though it does not initiate an analyzer self-test. An operation-complete query (OPC?) asks for the analyzer to return an ASCII character one (1) to the system controller when the preset operation is complete.

**Main Program**

"EDITLIST;" **(Activate List-Frequency-Table Edit Mode)**

Enters the frequency-list edit mode. This function is the same as pressing the softkey under the list-frequency mode. This begins the interaction with the analyzer and displays the list-frequency table on the controller screen.

"CLEL;" **(Clear List)**

Clears any existing tables.
Caution  Any existing segments are erased from the analyzer's internal registers. You may wish to comment-out this command line (in this case, the executable CLEL;) and simply enter new segments.

For this example, a new list is generated by creating a table of values.

A table header that shows the entered segments is printed on the display. The operator is prompted for each entry in the segment.

If the operator chooses to edit a segment, the entire modified segment must be re-entered.

   **Subroutine: Loadpoin:**

   The subroutine Loadpoin: reads in each segment value for:

   - start frequency
   - stop frequency
   - number of points in the segment

   During the subroutine, the operator is prompted for the number of segments to define and a table is created to hold the defined segments.

   After editing, each new segment is displayed as a table on the controller's display screen and the subroutine ends.

Once the list is complete the operator may choose to edit.

"DO YOU WANT TO EDIT? Y OR N" (Operator Prompt)

Choosing Y at the prompt activates the program subroutine Loadpoin:.

**Main Program Resumes**

"SADD;" (Add Segment)

Adds a segment by sending it to the analyzer. The segments are sent to the analyzer by adding segments to the list. The segment is defined by sending the table created.

"STAR";Table(I,1);"MHz"; (Select Start Frequency)

Sends the start frequency using the numbers contained in the variable Table(I,1) (the first value in the list-frequency table) as the data value.

"STOP";Table(I,2);"MHz"; (Select Stop Frequency)

Sends the stop frequency using the numbers contained in the variable Table(I,2) (the second value in the list-frequency table) as the data value.

"POIN",Table(I,3),";"; (Select Number of Points)

Sends the number of points in the trace using the numbers contained in the variable Table(I,3) (the third value in the list-frequency table) as the data value.

"SDON;" (Segment Done)

Closes a segment after editing.
Indicates that the segment is done. The segments do not need to be in numerical order. The analyzer will reorder and display them.

"EDITDONE;" (Edit Done)

Done editing list frequency table. Ends the segment editing.

"LISTFREQ;" (Activate List-Frequency Mode)

Enables the list-frequency mode.

"OPC?;WAIT;" (Wait for Reply with Operation Complete Query)

The controller waits for the analyzer to return a one (1), signifying that the last command received from the controller has completed. Upon reading the one (1), the analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.

Running the Program

Caution 
This example program will delete any existing limit lines before entering the new limits. If this is not desired, omit the line(s) that clear the existing limits (in this case, the executable CLEL;). This program begins by presetting the analyzer. The program will have to add the necessary command lines to set the analyzer to the specific operating conditions required for testing. The example program will show the limit lines defined, but the limits will always fail without additional analyzer setup.

The program displays the frequency-list table as it is entered. During editing, the displayed table is updated as each line is edited. The table is not re-ordered. At the completion of editing, the table is entered into the analyzer, and list-frequency mode is switched ON. During editing, pressing [ENTER] leaves an entry at the old value.

If the analyzer display is within the range of the segments entered, then the number of points-per-segment may be observed on the analyzer’s display.

Activate a marker and select the discrete-marker mode to observe the point spacing. Use an exaggerated scale with just a few points to find the list-frequency spacing between points.

Example 6B: Selecting a Single Segment from a Table of Segments

Note 
This program is stored as EXAMP6B on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

This example program demonstrates how to define a single segment as the operating-frequency range of the analyzer from a table of segments stored in the controller. The program assumes that a list-frequency table has already been entered into the analyzer, either manually, or using the program in Example 6A, “Setting Up a List-Frequency Sweep.”

The program first loads the list-frequency table into the computer by reading the start and stop frequencies of each segment and the number of points for each segment. The segments’ parameters are then displayed on the computer screen, and the operator can choose which segment is to be used by the analyzer. Note that only one segment can be chosen at a time.
The following is an outline of the program's processing sequence:

- The system is initialized.
- The list-frequency segment is edited.
- The largest segment number possible is set.
- The highest segment number in use is requested.
- The number of actual segments in use is read in.
- A list-frequency table is defined and the segments are read in to the controller from the analyzer.
- The operator selects one of the segments of the sweep.
- The controller "zooms-in" and sweeps the defined segment.
- The operator ends the program by entering segment number (0).
- The analyzer returns to sweeping all the segments in the table.
- The activation loop is ended and the program ends.

**Tutorial Program Explanation**

The following section explains "Example 6B: Selecting a Single Segment from a Table of Segments" at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer's remote commands, see "Command Reference."

**Initializing the System**

**IFC (HP-IB Meta-Message: Interface Clear)**

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

The program begins by opening the list-frequency segments, determining the number of segments that have been defined, and printing the data as a table on the controller display.

"EDITLIST;" **(Activate List-Frequency-Table-Edit Mode)**

Activates the list-frequency-table edit mode. This function is the same as pressing the softkey under the list-frequency mode. This begins the interaction with the analyzer and displays the list-frequency table on the controller screen.

"SEDI30;" **(Select Segment Editing Mode for Segment {30})**

Sets the analyzer to edit the largest segment number possible (30). The largest segment number, as defined in the active function area of the analyzer, is displayed.
"SEDI?" (Select Segment Query)

Sends the number of the highest segment defined.

A table is allocated in the controller to hold the number of segments defined in the analyzer. The segments are looped through to read the values into the segment table to display to the operator.

A subroutine is then activated by the program.

Subroutine Readlist:

This subroutine manages the controller functions related to editing the segment list.

"EDITLIST;" (Activate List-Frequency-Table Edit Mode)

Activates the list-frequency-table edit mode. This function is the same as pressing the softkey that activates the list-frequency mode. This command begins the interaction with the analyzer and displays the list-frequency table on the controller screen.

"SEDI",I,";" (Select Segment to Edit)

Reads the segment list into the controller by selecting a segment to edit. Each of the segment parameters are read by sending the parameter command and then reading the active area of the display.

For example:

"STAR;" (Send Start Frequency for Display as a Variable)

Puts start frequency value in active-display area.

"OUTPACTI;" (Output Active-Function Area)

Read value from active area. Place segment start frequency in table

"STOP;" (Send Stop Frequency for Display as a Variable)

Puts stop frequency value in active-display area.

"OUTPACTI;" (Output Active-Function Area)

Read value from active area. Place segment start frequency in table

"POIN;" (Send Number of Points for Display as a Variable)

Puts number of points value in active-display area.

"SELECT SEGMENT NUMBER: (0 TO EXIT)" (Operator Prompt)

Type in the number of the segment, and the analyzer will then “zoom-in” on that segment by activating it in the analyzer.

"EDITDONE;" (Done Editing)

Terminates the segment selection and activates the selected segment. The selected segment is now the new frequency range of the analyzer. Prompt and read another segment number.
"SSEG;" (Send Selected Segment)
Sends the segment number. Once (0) has been selected, the subroutine ends by restoring by restoring the sweep to all segments.

"ASEG;" (Set All Segment Sweep)
Returns to sweep and display the entire list frequency and the controller displays "PROGRAM ENDED"

"CPC?;WAIT;" (Wait for Reply with Operation Complete Query)
The controller waits for the analyzer to return a one (1), signifying that the last command received from the controller has completed. Upon reading the one (1), the analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.

Running the Program
The program will read the parameters for each list-frequency segment from the analyzer, and build a table containing all the segments. The parameters of each segment will be printed on the computer screen.

After all the segments are displayed, the program will prompt the operator for a specific segment to be used. Type in the number of the segment, and the analyzer will then "zoom-in" on that segment. The program will continue looping, allowing continuous selection of different segments. To exit the loop, type 0. This will restore all the segments (with the command ASEG), allowing the analyzer to sweep all of the segments, and the program will terminate.

Example 6C: Setting Up Limit Lines

Note
This program is stored as EXAMP6C on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

The purpose of this example is to show how to create a limit-test table and transmit it to the analyzer.

The command sequence for entering a limit-test table imitates the key sequence followed when entering a table from the analyzer’s front panel: there is a command for every key-press. Editing a limit is also the same as the key sequence, but remember that the analyzer automatically re-orders the table in order of increasing start frequency.

The limit-test table is also carried as part of the learn string. While it cannot be modified as part of the learn string, the learn string can be stored and recalled with very little effort. See section, "Data-Processing Chain" in Chapter 2 for details on using learn strings.

This example takes advantage of the computer’s capabilities to simplify creating and editing the table. The table is entered and completely edited before being transmitted to the analyzer. To simplify the programming task, options such as entering offsets are not included.

This example automates the front-panel operation of entering a limit-test table. Front-panel operation and limits are discussed in the “Application and Operation Concepts” in the HP 8753D/Option 011 Network Analyzer User’s Guide.

The following is an outline of the program’s processing sequence:
- The system is initialized.
- The limit lines are edited and cleared.
- The number of limits is requested.
- The limit table is created.
- The string array of limit types is created.
- The operator is prompted to enter the new limit values.
- The new limit table is sent back to the analyzer.
- The limit line is activated.
- The limit test is activated.
- The analyzer is returned to local control and the program ends.

**Tutorial Program Explanation**

The following section explains “Example 6C: Setting Up Limit Lines” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”

**Initializing the System**

**IFC (HP-IB Meta-Message: Interface Clear)**

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

"0PC?;PRES;" (Instrument Reset with Operation Complete Query)

Resets the network analyzer operating state to the default state (preset). This command is similar to pressing the front-panel (Preset), but it does not initiate an analyzer self-test. The analyzer sends an ASCII character one (1) to the instrument controller when the preset operation is complete.

**Main Program**

You must setup the analyzer to specific ranges to display and test the limits entered.

"EDITLIM;" (Edit Limit Lines)

Opens the limit table for editing.

"CLEL;" (Clear Limits)

Any existing limits are deleted.
"NUMBER OF LIMITS?" (Operator Prompt)

Requests the number of limits to be entered. A table is created in the controller to hold the following entries:

- stimulus value (frequency)
- upper limit value
- lower limit value
- limit type

A string array is created for the limits.

The controller display is cleared and the table header is printed.

Subroutine: Loadlimit:

This subroutine manages the controller functions related to editing the limit table.

Cycle through the limits desired and enter the limit values.

The stimulus entry prompt: "STIMULUS VALUE? (MHz)" is displayed on the controller console. The operator then enters a value based on MHz as the units.

The upper limit entry prompt: "UPPER LIMIT VALUE? (DB)" is displayed on the controller console. The operator then enters a value based on DB as the units.

The lower limit entry prompt: "LOWER LIMIT VALUE? (DB)" is displayed on the controller console. The operator then enters a value based on DB as the units.

"ENTRY TYPE? (FL=FLAT, SL=SLOPE, SP=SINGPOINT)" (Operator Prompt)

Prompts the operator to enter the type representing the desired limit. The choices are FL=FLAT, SL=SLOPE, SP=SINGPOINT.

"DO YOU WANT TO EDIT? Y OR N"

Prompts you to edit the limit table or to send the existing table to the analyzer. If you do make any changes to the entries, you must actually re-enter the entire limit.

"EDITLIM;"

Opens the limit table so you can begin to edit the table. If the limit table was not cleared as in this example, the new limits will just be added to the old ones.

"ENTRY NUMBER?" (Operator Prompt)

Prompts the operator to enter the number representing the desired limit. When the operator enters a value, the program activates a subroutine.

"SADD;LIMS"; stimulus entry,"MHz;" (Add Segment)

Begins the sequence that adds the segments to the analyzer. The stimulus value that was entered is included with the limit stimulus command.

"LIMU";(value);"DB;" (Select Upper Limit Value)

Sends the upper limit value from the table to the analyzer.
"LIML"; (value); "DB"; (Select Limit Type)
Sends the upper limit value from the table to the analyzer
The type of limit point is sent to the analyzer by constructing a command based upon the
string values entered in the table:

- "FL" sends "LIMTFL;" for a flat limit type.
- "SL" sends "LIMTSL;" for a sloped limit type.
- "SP" sends "LIMTSP;" for a point limit type.

"SDON;" (Segment Done)
Ends the complete segment definition. The segments are each sent in the same process.
Then the subroutine ends.

Main Program Resumes

"EDITDONE;" (Done Editing)
Closes the edit list frequency or limit table.

"LIMILINEON;" (Activate Limit Lines)
Activates the limit lines.

"LIMITESTON;" (Activate Limit Tests)
Activates the limit testing.

"OFC; WAIT;" (Wait for Reply with Operation Complete Query)
The controller waits for the analyzer to return a one (1), signifying that the last command
received from the controller has completed. Upon reading the one (1), the analyzer is
addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on
the HP-IB and the program ends.

Running the Program

Caution
This example program will delete any existing limit lines before entering
the new limits. If this is not desired, omit the line(s) that clear the existing
limits (in this case, the executable "CLEL:" contained in LINE 190). This
program begins by presetting the analyzer. The programmer will have to add
the necessary command lines to set the analyzer to the operating conditions
required for testing. The example program will show the limit lines defined,
but the limits will always fail without additional analyzer setup.

The program displays the limit table as it is entered. During editing, the displayed table is
updated as each line is edited. The table is not reordered. At the completion of editing, the
table is entered into the analyzer, and limit-testing mode switched ON. The analyzer will
rearrange the table in ascending order starting with the lowest start frequency entry.
Example 6D: Performing PASS/FAIL Tests While Tuning

Note
This program is stored as EXAMP6D on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

The purpose of this example is to demonstrate the use of the limit/search-fail bits in event-status-register B, to determine whether a device passes the specified limits. Limits can be entered manually, or using the Example 5A.

The limit/search-fail bits are set and latched when limit testing or a marker search fails. There are four bits, one for each channel for both limit testing and marker search. See Figure 2-5 and Table 2-3 for additional information. Their purpose is to allow the computer to determine whether the test/search executed was successful. They are used in the following sequence:

1. Clear event-status-register B.
2. Trigger the limit test or marker search.
3. Check the appropriate fail bit.

When using limit testing, the best way to trigger the limit test is to trigger a single sweep. By the time the single sweep command "SING:" finishes, limit testing will have occurred.

Note
If the device is tuned during the sweep, it may be tuned into and then out of limit, causing a limit test to qualify as "passed" when the device is not in fact within the specified limits.

When using marker searches (max, min, target, and widths), outputting marker or bandwidth values automatically triggers any related searches. Therefore, all that is required is to check the fail bit after reading the data.

In this example, several consecutive sweeps must qualify as "passing" in order to insure that the limit-test pass was not extraneous due to the device settling or operator tuning during the sweep. Upon running the program, the number of "passed" sweeps for qualification is entered. For very slow sweeps, a small number of sweeps such as two are appropriate. For relatively fast sweeps, where the device requires time to settle after tuning, as many sweeps as six or more sweeps may be more appropriate.

A limit-test table can be entered over HP-IB. The sequence is very similar to that used in entering a list-frequency table, as shown in Example 5D. The manual (front-panel entry) sequence is closely followed.

The following is an outline of the program's processing sequence:

- The system is initialized.
- The pass counter is initialized on entry.
- The analyzer takes a sweep.
- The event-status register B byte is output and the channel-1 limit is tested.
- If the device fails the first sweep, the operator is prompted to insure it is tuned correctly and the device is measured again.
- Once the device passes, the operator is prompted not to touch the device as testing continues.
- If the device passes the required number of sweeps, the operator is prompted that the device has passed and to connect the next device for testing.
- The program initializes the pass counter and begins to measure the new device.

**Tutorial Program Explanation**

The following section explains "Example 6D: Performing PASS/FAIL Tests While Tuning" at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer's remote commands, see "Command Reference."

**Initializing the System**

**IFC (HP-IB Meta-Message: Interface Clear)**

 Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

 Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

**Main Program**

You must leave the limit lines and analyzer setup intact to perform this type of testing.

"Number of consecutive passed sweeps for qualification?" (**Operator Prompt**)

Prompts the operator to choose the number of sweeps to determine qualification; that is, how many sweeps must pass before the device is considered to have passed the limit test. The program's pass counter is initialized when the entry is made.

"DPC?;SING;" (**Select Single-Sweep Mode with Operation Complete Query**)

Sets a single sweep and then queries the analyzer to insure that the sweep is complete before executing any further instructions. Once the sweep is complete (the analyzer has returned a one (1) to the controller), the limit test will be performed upon completion of the sweep cycle.

"ESB?;" (**Event-Status-Register B Query**)

Queries event-status-register B and returns the contents of the status byte to the program. The fall bit 4 (or numerical value 16), is read from the status byte to determine if the limit test failed.

- If the test passed, the "passed counter" is incremented and compared to the number of passes required to qualify the test as "passed." The controller creates a high pitch beep to warn you not to adjust the device under test and displays the message "LEAVE DEVICE ALONE".
- If the fall bit for channel one is set, the number of sweeps in the "passed counter" is reset to zero. The controller creates a low pitch beep to indicate that the device is failing the test and displays the message "TUNE DEVICE AS NECESSARY". The device under test may require adjustment by the operator.

If not enough sweeps have passed, the measurement cycle is repeated. The program issues "DPC?;SING;" to repeat the measurement process.
"DEVICE PASSED!"

Indicates that enough sweeps have passed to qualify the device. The controller creates a series of tones and displays a message that prompts you to change the device under test.

"PRESS RETURN FOR NEXT DEVICE" (Operator Prompt)

Prompts the operator to begin the testing cycle for the new device by pressing [ENTER].

Running the Program

| Note | This program assumes a response calibration (thru calibration) or a full 2-port measurement calibration has been performed prior to running the program. |

Set up a limit-test table on channel 1 for a specific device either manually, or using the program in Example 5A.

Run the program, and enter the number of passed sweeps desired for qualification. After entering the qualification number, connect the device. When a sweep passes, the computer beeps. When enough consecutive sweeps qualify the device as "passing," the computer emits a dual-tone beep to attract the attention of the operator, and then prompts for a new device.

To test the program's pass/fail accuracy, try causing the DUT to fail by loosening the cables connecting the DUT to the analyzer and running the program again.
Example 7: Report Generation

The analyzer has three operating modes with respect to HP-IB. These modes can be changed by accessing softkeys in the LOCAL menu. System-controller mode is used when no computer is present. This mode allows the analyzer to control the system. The other two modes allow a remote system controller to coordinate certain actions: in talker/listener mode the remote system controller can control the analyzer, as well as coordinate plotting and printing; and in pass-control mode the remote system controller can pass active control to the analyzer so that the analyzer can plot, print, control a power meter, or load/store to disk. The amount of peripheral interaction is the main difference between talker/listener and pass-control mode. See Chapter 2 for more details.

Example 7A1: Operation Using Talker/Listener Mode

Note
This program is stored as EXAMP7A1 on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

The commands OUTPPLOT and OUTPRIN allow talker/listener mode plotting and printing via a one way data path from the analyzer to the plotter or printer. The computer sets up the path by addressing the analyzer to talk, the plotter to listen, and then releases control of the analyzer in order to transfer the data. The analyzer will then make the plot or print. When it is finished, it asserts the End or Identify (EOI) control line on HP-IB. The controller detects the presence of EOI and re-asserts control of the HP-IB. This example program makes a plot using the talker/listener mode.

Note
One of the attributes of the OUTPPLOT command is that the plot can include the current softkey menu. The plotting of the softkeys is enabled with the PSOFTON; command and disabled with the PSOFTOFF; command.

The following is an outline of the program’s processing sequence:

- The system is initialized.
- The selected frequency span is swept once.
- The plot command is sent to the analyzer.
- The analyzer is set to talker mode and the plotter is set to listener mode.
- The plot is spooled to the plotter.
- The analyzer is set to listener mode when the controller detects an EOI from the analyzer.
- The controller puts the analyzer back in continuous-sweep mode.
- The analyzer is returned to local control and the program ends.

Tutorial Program Explanation

The following section explains “Example 7A1: Analyzer Operation Using Talker/Listener Mode” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”
Initializing the System

IFC (HP-IB Meta-Message: Interface Clear)
Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

SDC (HP-IB Meta-Message: Selected Device Clear)
Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

Main Program

"OUTPLOT;" (Output Plot Data in ASCII Format)
Commands the analyzer to plot. (For printing, substitute the "OUTPRIN;" executable.)

7;UNL LISTEN 5 TALK 16 DATA" (Create Data Path)
A data path is established from the analyzer to the plotter. Before the plotter operation can take place, the talker/listener assignments must be made by the instrument controller.

- The plotter is assigned as a listener at address 5, the default address for a plotter. If a printer is used, the printer is assigned as a listener at address 1, the default address for a printer.
- The analyzer is addressed to talk as device 16; that is, transmit the contents of its output queue.

After the addressing is complete, the instrument controller releases the ATN line and the bus changes from command to data mode. The analyzer becomes a talker, the plotter a listener, and the plot begins.

Note
The "OUTPLOT;" command allows you to include the current softkey menu on the hardcopy. The plotting of the softkeys is enabled with the command "PS0FTON;" and disabled with "PS0FTOFF;".

"Plotting and waiting for EOI" (Program Message)
Prompts the operator to wait for the analyzer to assert the EOI line, indicating the end of transmission. The HP-IB interface status registers for the interface installed on the instrument controller are read and tested for the presence of the active EOI line.

"End of plot" (Program Message)
Indicates the computer has detected the active EOI line on the interface and the system controller returns the analyzers device mode to listener. The analyzer will go on asserting EOI until some other activity on the bus causes it to clear the line.

"CONT;" (Select Continuous-Sweep Mode)
Places the analyzer in continuous-sweep mode.
"OPC?;WAIT;" (Wait for Reply with Operation Complete Query)

The controller waits for the analyzer to return a one (1), signifying that the last command received from the controller has completed. Upon reading the one (1), the analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.

Running the Program

If a problem arises with the plotter, such as no pen or paper, the analyzer cannot detect the situation because it only has a one-way path of communication. Therefore, the analyzer attempts to continue plotting until you intervene and abort the plot by pressing the LOCAL key. This key aborts the plot, causes the warning message "CAUTION: PLOT ABORTED" asserts EOI, and frees the computer.

Because of possible peripheral malfunctions, it is generally advisable to use pass-control mode, which allows two way communication between the peripherals and the analyzer.

Example 7A2: Controlling Peripherals Using Pass-Control Mode

If the analyzer is in pass-control mode and it receives a command telling it to plot, print, control a power meter, or store/load to disk, it sets bit 1 in the event-status register to indicate that it requires control of the bus. If the computer then uses the HP-IB pass-control command to pass control to the analyzer, the analyzer will take control of the bus and access the peripheral. When the analyzer no longer requires control, it will pass control back to the computer. For a discussion on the pass-control mode, see Chapter 2.

In this example, the pass-control mode is used to allow the network analyzer to dump a screen display to a printer.

Pass-control mode allows the analyzer to control the printer while sending the screen display to be printed. Once the printer-dump operation is complete, the analyzer passes control back to the controller and the controller continues programming the analyzer. The analyzer requests control from the instrument controller and the controller allows the analyzer to take control of the HP-IB and dump the plot. The instrument controller must not interact with the HP-IB while this remote analyzer control is taking place.

| Note | The analyzer assumes that the address of the computer is correctly stored in its HP-IB addresses menu under (LOCAL ADDRESS) CONTROLLER. If this address is incorrect, control will not return to the computer. Similarly, if control is passed to the analyzer while it is in talker/listener mode, control will not return to the computer. |

Control should not be passed to the analyzer before it has set event-status-register bit 1 making it Request Active Control. If the analyzer receives control before the bit is set, control is passed immediately back to the controller.

When the analyzer becomes the active system controller, it is free to address devices to talk and listen as required. The only functions denied the analyzer are the ability to assert the interface clear line (IFC), and the remote line (REN). These are reserved for the master system controller. As the active system controller, the analyzer can send and receive messages from printers, plotters, and disk drives.

The following is an outline of the program’s processing sequence:

- The system is initialized.
- The status registers are cleared.
- Bit 1 of ESR request control is set.
- The ESR interrupt for SRQ is enabled.
- The pass-control mode is enabled.
- The data is dumped to the printer.
- The program loops until the SRQ bit is set.
- The status byte is read with a serial poll.
- The program tests for bit 6, SRQ.
- If SRQ is detected, the program tests for pass control (bit 5 of the status byte).
- If the analyzer requests control, the system controller gives the analyzer control of the bus.
- The program loops and waits for the analyzer to complete the print dump.
- The controller reads the HP-IB interface status register.
- If bit 6 is active in the controller, control has returned from the analyzer to the controller.
- The status-byte assignments are cleared.
- The analyzer returns to continuous-sweep mode.
- The analyzer is returned to local control and the program ends.

**Tutorial Program Explanation**

The following section explains "Example 7A2: Controlling Peripherals in Pass-Control Mode" at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer's remote commands, see "Command Reference."

**Initializing the System**

**IFC (HP-IB Meta-Message: Interface Clear)**

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

**Main Program**

"OPC?; SING;" *(Select Single-Sweep Mode with Operation Complete Query)*

Sets a single sweep and then queries the analyzer to insure that the sweep is complete. The analyzer is now in the state used to dump its display to the printer.

"CLES;" *(Clear Status)*

Clears the status register, the event-status registers, and the enable registers.
"ESE2;" (Event-Status Enable {Control-Summary Bit 1})

Enables the request-control bit when the analyzer is ready to perform a the print function over the HP-IB. This is part of the pass-control operation synchronization. The control-summary bit is enabled in the event-status register as bit 1, decimal 2.

When this bit is set, the event-status-register bit is set in the analyzer-status register as bit 5, decimal 32.

Bit 5, decimal 32 will indicate that a request control is active.

"SRE 32;" (Service-Request Enable {Bit 5 of the Status Byte})

Service request enable. Enables bit 5 of the analyzer’s status byte so that an SRQ is generated.

"PRINALL;" (Print Display)

Copies the display, in raster graphics mode, to a printer.

"USEPASC;" (Enable Pass-Control Mode)

The analyzer sets bit 2 in the event-status register to request pass control from the controller. The controller recognizes the presence of the bit 2 by reading the status byte from the analyzer. In this example, a serial poll is performed by the analyzer to read the status byte and to test for bit 5 which indicates that an event has occurred in the event-status register.

The pass control operation has now been established. When the analyzer requests control, bit 1 in the event status register will be set. This bit will cause bit 5 in the status byte to be set and generate an SQR. The analyzer’s status byte is read by performing a serial poll until SRQ is detected. Once SRQ occurs, bit 5 is tested. If bit 5 is set the analyzer is requesting control.

If the controller finds a status byte bit other than status byte bit 5 has been enabled, the controller displays the message: “SRQ but not request pass control” and the program halts.

The system controller will pass control by addressing the analyzer to talk and sending the TCT message (take control). The controller relinquishes control to the analyzer when the ATN line is released. The analyzer sends the printer dump to the printer.

The analyzer will very briefly flash the message WAITING FOR CONTROL before actually receiving control and generating the printer output.

"Printing from analyzer and waiting for control" (Program Message)

Indicates that the system is waiting for the printing to complete. Once the printer dump is complete, the analyzer passes control back to the address stored as the controller address under the LOCAL SET ADDRESSES menu. The computer recognizes that the transfer has taken place by monitoring the active controller status bit in the HP-IB interface registers. The computer then exits the wait loop.

"Control returned from analyzer" (Program Message)

Indicates that the control has been passed from the analyzer to the controller. The program can now continue to interact with the analyzer. The computer releases remote control of the analyzer.

"TALKLIST;" (Select Talker/Listener Mode)
Enables the talker/listener mode in the analyzer.

"CLES;" (Clear Status)
Clears the status register, the event-status registers, and the enable registers.

"CONT;" (Select Continuous-Sweep Mode)
Places the analyzer in continuous-sweep mode.

"DFC?;WAIT;" (Wait for Reply with Operation Complete Query)
The controller waits for the analyzer to return a one (1), signifying that the last command
received from the controller has completed. Upon reading the one (1), the analyzer is
addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on
the HP-IB and the program ends.

Running the Program
The analyzer will briefly flash the message WAITING FOR CONTROL, before actually receiving
control and generating the printer output. The computer will display the Printing from
analyzer and waiting for control message.

When the printer output is complete, the analyzer passes control back to the address stored
as the controller address under the LOCAL SET ADDRESSES menu. The computer will detect
the return of active control and exit the wait loop. The controller will display the message
Control returned from analyzer and then release the analyzer from remote control.

Note Because the program waits for the analyzer's request for control, it can be used
to respond to front-panel requests as well. Remove the "PRINALL;" command
from the program and run the program. Nothing will happen until the operator
requests a print, plot, or disk access from the front panel of the analyzer. For
example, press LOCAL COPY and PRINT MONOCHROME.

Example 7A3: Printing with the Serial Port

Note This program is stored as EXAMP7A3 on the "HP 8753D/Option 011
Programming Examples" disk received with the network analyzer.

This program will select the serial port and program the analyzer to copy its display to a
printer. There are a number of commands associated with the serial and parallel ports which
allow the programmer to configure the output modes, for example the baud rate and the
handshake type used by the port and the printer. In this example, the serial port is configured
by the program. The interface may also be configured from the analyzer’s front-panel keys by
pressing LOCAL SET ADDRESSES PRINTER PORT. This menu allows manual selection of the
serial-interface parameters.

Since the HP-IB port is not being used for the copy operation, programming of the analyzer
and measurement operations may continue once the copy operation has been initiated. An
internal spooler in the analyzer's memory provides buffering of the printer operation. In the
example which follows, the status byte of the analyzer is checked to determine when the print
operation is complete.

- The analyzer is initialized.
- A single sweep is taken and the analyzer is placed in hold mode.
- The status registers are cleared.
- The copy-complete bit is set and enabled.
- The printer operation and communication modes are set.
- The print command is sent.
- The analyzer is released from remote control and placed in continuous-sweep mode.
- The analyzer is polled until the status bit representing copy complete is detected.
- The analyzer is released from remote control and the program ends.

**Tutorial Program Explanation**

The following section explains “Example 7A3: Printing with the Serial Port” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”

**Initializing the System**

**IFC (HP-IB Meta-Message: Interface Clear)**

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

**Main Program**

"CLES;" (Clear Status)

Clears the status register, the event-status registers, and the enable registers.

"ESNB128;" (Enable Copy Complete)

Enables the copy complete bit in event-status-register B.

"SRE4;" (Enable Event-Status-Register B)

Enables the copy complete bit in event-status-register B.

"PRNTAUTF OFF;" (Deactivate Printer Autofeed Function)

Deactivates the printer’s auto feed function.

"PRNTYPTJ;" (Select Print Device {ThinkJet})

An HP ThinkJet printer is selected as the print device.

"PRNPRRTSERI;" (Select Output Port {Serial})

The display data is routed to the serial port.
"PRNTBAUD 9600;" (Select Baud Rate)
The ThinkJet’s baud rate is set to 9600 bits-per-second.

"PRNHNDSHK XON;" (Select Handshake {Xon-Xoff})
The Xon-Xoff printer handshake mode is selected

"PRINALL;" (Print All)
Prints all list values or operating and marker parameters in ASCII.
The entire screen display is output to the printer.

"PRINTING" (Program Message)
Acknowledges that the display information has been output to the printer.

"CONT;" (Select Continuous-Sweep Mode)
Places the analyzer in continuous-sweep mode.
The controller then performs a serial poll to test for bit 2 (data-entry-complete bit) enabled in event-status-register B. The serial poll is repeated until the bit is set at the end of the printing process.
The controller then displays the message: "DONE" on its display console.
The analyzer is then addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.

Running the Program
Run the program. The analyzer is initialized, set to single-sweep mode, and a sweep is taken.
The program sets the system up to print the analyzer’s display to an HP ThinkJet printer connected to the interface. At this time, the analyzer can continue making measurements as the ThinkJet prints the display. When the analyzer display has finished printing, the controller displays the message: "DONE", the analyzer is released from HP-IB control, and the program ends.

Example 7B1: Plotting to a File and Transferring the File Data to a Plotter

| Note | This program is stored as EXAMP7B1 on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer. |

Another report-generation technique is to transfer the plotter string to a disk file, and retrieve and plot the disk file at another time. Test time is increased when a plot occurs during the process. It may be more convenient to plot the data at another site or time. One solution to this problem is to capture the plot data using the controller and store it to a disk file. This disk file may then be read from the controller and the contents transferred to a plotter. This next example shows a method of accomplishing this task.
The analyzer is initialized without presetting the analyzer. The data that is in place on the analyzer is not disturbed by the program operation. A large string is dimensioned to hold the plotter commands as they are received from the analyzer. The length of this string depends upon the complexity of the analyzer’s display. The analyzer is placed in the single-sweep mode.
and OPC?;SING; is used to make sure that operation is complete before plotting. The plotting begins with the OUTPLOT; command.

The string transfer is ended by the controller detecting the EOI line which the analyzer pulls at the end of the transfer. The string transfer terminates and the plot data is now stored in a string in the analyzer.

These strings contain ASCII characters which represent the plotter commands in HP-GL (Hewlett-Packard Graphics Language). A disk file is created and the string is written into the file containing the display-plot commands.

Once the strings are transferred to the disk file, the file pointer is rewound and the data read out into a string for plotting. The string is sent to the plotter which uses the commands to generate a plot.

The following is an outline of the program's processing sequence:

- The system is initialized.
- The string for plotter commands is defined.
- The frequency span is swept once.
- The plotter output is requested and read into the plot string.
- A plot file is created in the controller.
- The plot string is stored into the disk file.
- The plot string is read from the disk file and sent to the plotter.
- The analyzer returns to continuous-sweep mode.
- The analyzer is returned to local control and the program ends.

**Tutorial Program Explanation**

The following section explains “Example 7B: Plotting to a File and Transferring the File Data to a Plotter” at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer’s remote commands, see “Command Reference.”

**Initializing the System**

**IFC (HP-IB Meta-Message: Interface Clear)**

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

**Main Program**

A large string is dimensioned to hold the plotter commands as they are received from the analyzer. The length of this string depends upon the complexity of the display on the analyzer.

"OPC?;SING;" (Select Single-Sweep Mode with Operation Complete Query)
Sets a single sweep and then queries the analyzer to insure that the sweep is complete. The analyzer is now in the state used to dump its display to the printer.

"OUTPLOT;" (Output Plot Data in ASCII Format)
Requests the analyzer to plot using the talker/listener mode. Outputs the plot string in ASCII format to the HP-IB port.

The plotter output is read into the string buffer as one large string. Once the plotter commands have been received, the prompt is displayed.

"Plotter output complete. Press RETURN to store on disk." (Prompt)
A data file is created and opened to receive the plotter commands. Once the file transfer takes place, another prompt is displayed.

"Plot to file is complete. Press RETURN to plot." (Prompt)
When the operator presses [ENTER], the file pointer is reset, the plot string is read from the file and sent to the plotter, and the controller displays the message: “Plot is complete. End of program.”

"CONT;" (Select Continuous-Sweep Mode)
The analyzer is returned to continuous-sweep mode.

"OPC?;WAIT;" (Wait for Reply with Operation Complete Query)
The controller waits for the analyzer to return a one (1), signifying that the last command received from the controller has completed. Upon reading the one (1), the analyzer is addressed and returned to local control with the HP-IB Meta-Message: Go To Local (GTL) on the HP-IB and the program ends.

Running the Program
The program begins by initializing the analyzer and placing it into single-sweep mode. The plotter commands are captured into strings in the controller. The controller display prompts Plotter output complete. Press RETURN to store on disk. Pressing [ENTER] causes the data to be stored to disk. Once this task is complete, the program prompts once more, Plot to file is complete. Press Return to plot. After pressing [ENTER] again, the string output is sent to the plotter and the plot begins. Once the plot is complete, the program prompts Plot is complete. End of program. and the analyzer begins sweeping and returns to local control.

Utilizing PC-Graphics Applications Using the Plot File
You can use this Example 7B1 to generate a plot that can be read into a PC and used in several different graphics generation programs. HP-GL is a commonly recognized graphic format and may be used to transfer information to PC application programs such as CorelDRAW!®, Lotus Freelance® and other graphics packages. By importing the graphics data into these application packages, you can generate reports in many work-processors.

You can then use graphic-data files to generate the following:
- test results documentation
- data sheets from testing results
- archival information for a digital-storage medium
If you would like to create a disk file for graphics processing, modify the previous program to only store the plotter commands to the disk file. The PC will now have a DOS-format file (.hpg) file that can be imported and examined by the graphics package.

**LIF2DOS**

The plot file generated from the previous program (Example 7B) may be utilized on a PC with the graphics utility programs previously mentioned. A disk format translation is required for the PC to read the LIF format file. The translation to PC-DOS format files may be achieved with the LIF2DOS.EXE utility. This utility is included on the PC-DOS format disk of programming examples, “HP 8753D/Option 011 Programming Examples” (HP part number 08753-10028).

Create a directory on the PC hard disk volume and copy the LIF2DOS.EXE file to the volume. Run the utility without any arguments to obtain directions associated with the program.

To transform a LIF-formatted file to DOS format, simply type LIF2DOS followed by a space and the drive/directory path and file name of the file you would like to transform. For example, if your file was named “PLOT”, and was located on a disk in the computer’s b: drive, you would type:

**LIF2DOS B:PLOT**

**Example 7B2: Reading Plot Files From a Disk**

| Note | This program is stored as EXAMP7B2 on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer. |

This example program reads and plots files which have been stored on a LIF formatted disk by the HP 8753D/Option 011. The plots may be sent to either a plotter with auto-feed capability, such as the HP 7550B, or an HP-GL/2 compatible printer, such as a LaserJet 4 Series (monochrome) printer or a DeskJet 1200C (color) printer.

| Note | Sending plots to disk is discussed in the chapter titled “Printing, Plotting, and Saving Measurement Results” in the HP 8753D/Option 011 Network Analyzer User’s Guide. |
| | This section provides detailed information on file naming conventions and instructions on printing multiple plots-per-page. |
| | This program requires HP BASIC 6.0 or greater which provides the use of wild cards with the catalog command. |
| | The peripheral HP-IB addresses and the hard copy device selection are hard coded and may need to be changed for different systems configurations. Turn to “Example 7B2” in the next chapter and refer to lines 1130 through 1240 in the example program. |

This program example provides the form feeds to separate the plots. If the HP 8753D/Option 011 has been configured to store multiple plots-per-page, this program will generate those plots. A file naming convention has been devised to allow the program perform several printer-setup functions. These include: initializing the printer for HP-GL/2 at the beginning of a page, configuring the printer to plot multiple plots to the same page, if desired, and then sending a page eject (form feed) to the hardcopy device at the completion of the printing process.

The plot file name is made up of four parts. The first three are generated automatically by the HP 8753D/Option 011 whenever a plot is requested:
1. the prefix “PLOT”

2. a two-digit sequence number in the range of 00 to 31

3. a two-letter output-format code to indicate the plot quadrant position:
   - LU (Left Upper)
   - LL (Left Lower)
   - RU (Right Upper)
   - RL (Right Lower)
   - FP (Full Page)

For example, the first full page plot to a disk would be named “PLOT00FP”. The second plot, to be located in the lower-right hand corner of the page would be named “PLOT01RL,” and so on.

The fourth part is an optional character. It is used to indicate that the file is part of a multiple-file plot on the same graticule. See “Printing Multiple Measurements Per Page” in chapter 3 of the HP 8753D/Option 011 Network Analyzer User’s Guide.

For detailed information on plotting to disk and outputting the plot files to a printer/plotter see “Printing Plotting, and Saving Measurements Results” in the HP 8753D/Option 011 Network Analyzer User’s Guide.

The following is an outline of the program’s processing sequence:

- Hardcopy device control strings are created.
- The hardcopy output device is defined.
- The disc storing the file names is cataloged.
- Files that match the name specifier in the file name array (Flnm$) are counted.
- Flnm$ is dimensioned for the number of files that match the name specifier.
- The file name array is sorted.
- A form feed is sent to the hardcopy device.
- Each of the files in the file name array is processed and sent to the output device.
  - The current file name root string is defined.
  - If the hardcopy device is a printer, then an HP-GL initialization string is output.
  - Each file which matches the file name root string to the hardcopy device is output.
  - After a form feed is sent to the hardcopy device, the printing/plotting process begins.

**Running the Program**

This program allows you to plot or print as many as four HP 8753D/Option 011 display dumps (stored on a LIF formatted disk) on one side of a single sheet of paper. Refer to the instructions detailed in the chapter “Printing, Plotting, and Saving Measurement Results,” in the HP 8753D/Option 011 Network Analyzer User’s Guide.
Example 7C: Reading ASCII Disk Files to the Instrument Controller’s Disk File (HP 8753D standard only)

Note: This program is stored as EXAMP7C on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

Another approach to accessing the analyzer’s test results is to store the data onto a disk file from the analyzer. This operation generates an ASCII file of the analyzer data in a CITIFILE format. A typical file is generated by the next example and is shown below:

```
CITIFILE A.01.00
#NA VERSION HP8753D.05.34

NAME DATA
VAR FREQ MAG 11
DATA S[1,1] RI
SEG_LIST_BEGIN
SEG 100000000 200000000 11
SEG_LIST_END
BEGIN
8.30566E-1,-1.36749E-1
8.27392E-1,-1.43676E-1
8.26080E-1,-1.52069E-1
8.25653E-1,-1.60003E-1
8.26385E-1,-1.68029E-1
8.26507E-1,-1.77154E-1
8.26263E-1,-1.87316E-1
8.26721E-1,-1.97265E-1
8.2724E-1,-2.07611E-1
8.28552E-1,-2.19940E-1
8.29620E-1,-2.31109E-1
END
```

This data file is stored by the analyzer under remote control or manually from the front panel. See “Printing, Plotting, or Saving Measurement Results” of the HP 8753D/Option 011 Network Analyzer User’s Guide for more details on manual operation. This program performs the same steps that are required to manually store a file from front panel.

Note: If a PC is being used as the system controller, the disk format will have to be transformed from LIF to PC-DOS format.

To accomplish this task, a file-transformation utility, “LIF2DOS.EXE” has been provided on the “HP 8753D/Option 011 Programming Examples” disk (HP part number 08753-10025).

Once transformed, the PC will now have a DOS-format file to read and interpret.

A more extensive utility called LIFUTIL allows file transformation in both directions and is available as a commercial product. Call the nearest HP sales and service office for ordering information.
This program stores a file in the same manner as an operator would store a file on the analyzer's internal disk drive from the front panel.

This example explains the process of storing the data from the analyzer to a file on the internal disk drive. There is also a program to read the data from the file into a data array for further processing or reformattting to another file type. The internal drive will store in the same format that is present on the disk. A new disk may be formatted in either LIF or DOS. For the example, the assumption has been made that the format transformation has already taken place, and there is a file that can be read record by record, from which data can be retrieved.

The goal of this example is to recover an array of stimulus frequency along with the trace-data values. CITIFILES contain the real and imaginary values of each data point. Some further transformation will be required to obtain magnitude values, for example.

The disk file contents for this example are shown above. This file contains more information than will be used in this example. The file is accessed and the records read from the file and printed on the controller display to observe the actual file contents. The file pointer is reset and the records are then read and interpreted for their data contents.

The first six records are skipped for this example. The seventh record contains the stimulus-frequency values and the number of points in the trace. These values are read from the record. The frequency increment, or point spacing, is calculated and used later the frequency-data calculations for a point. Two more records are skipped and the next is the first record representing data values. The data values are read in a loop until the values for the number of points have been recovered from the file. The data values are tabulated and printed out on the controller display.

The following is an outline of the program's processing sequence:

- An I/O path is assigned to the analyzer.
- The system is initialized.
- A string is dimensioned to hold a file record.
- The analyzer operating state is set.
- The internal drive is selected for storage (only ASCII data is stored).
- A file name is entered and the data stored into it.
- The operator is prompted to move the disk to the controller disk drive.
- The disk file is read and the contents displayed.
- The file pointer is rewound.
- The file contents are converted to trace data.
- The frequency and complex-data pair is displayed for each point.
- The analyzer is restored to continuous-sweep mode.
- The analyzer is returned to local control and the program ends.
Tutorial Program Explanation

The following section explains "Example 7C: Reading ASCII Disk Files to the Instrument Controller's Disk File" at the instrument-command level. Individual commands used in the program and their affects on the system are explained. For a list of all of the network analyzer's remote commands, see "Command Reference."

Initializing the System

**IFC (HP-IB Meta-Message: Interface Clear)**

Resets the HP-IB interface by toggling the state of the IFC line and taking control of the bus, halting all operations for all connected instruments.

**SDC (HP-IB Meta-Message: Selected Device Clear)**

Resets the HP-IB interface within the addressed device. When addressed to the analyzer, this command will reset the bus driver chip, but not the analyzer state. Other HP-IB instruments may be affected differently by this command. In many cases, the instrument settings are reset to a power-on condition.

"OPC?; PRET;" *(Instrument Preset with Operation Complete Query)*

Resets the network analyzer operating state to the default state (preset). This command is similar to pressing the front-panel (PRET), but it does not initiate an analyzer self-test. The analyzer sends an ASCII character one (1) to the instrument controller when the preset operation is complete.

Main Program

"STAR100MHZ;" *(Select Start Frequency at 100 MHz)*

Selects 100 MHz as the analyzer's start frequency.

"STOP200MHZ;" *(Select Stop Frequency at 200 MHz)*

Selects 200 MHz as the analyzer's stop frequency.

"POIN 11;" *(Edit Segment Points {11})*

Sets the number of points in the trace to 11.

This reduced span and number of points are chosen to keep the example simple and reduce the effort to decode the data file. In actual use, the task could be more involved with list frequency sweeps or log sweeps.

"OPC?; SING;" *(Select Single-Sweep Mode with Operation Complete Query)*

Sets a single sweep and then queries the analyzer to insure that the sweep is complete before executing any further instructions. Once the sweep is complete (the analyzer has returned a one (1) to the controller), the limit test will be performed upon completion of the sweep cycle.

"INTD;" *(Select Storage Internal Disk)*

Selects the analyzer's internal disk as the data storage destination.
"EXTHFORMON;" (Activate Store-Formatted-Data-Only Mode)
Activates the analyzer’s store-formatted-data-only mode.

"EXTHDAT00N;" (Activate Store-Data-Only Mode)
Sets the store-data-only mode. This mode allows you to store only the data, not the accompanying files of instrument state and calibration data. This mode will not allow the analyzer to recall the data without the analyzer setups.

"Enter data file name (5 chars)" (Operator Prompt)
Prompts the operator to enter a 5-character name for the file. After entering, the program changes the file name characters to upper case.

"TITF1" (Title Save-Register 1)
Save-register 1 is titled and encoded with double-quotations marks using the file name input by the operator.

"SAVUASCII;" (Select ASCII/CITIFILE Save Format)
Saves the file as ASCII/CITIFILE data and completes the command sequence for file storage.

"STOR1;" (Store to Disk)
Stores the ASCII data to the disk file.

"Place disk in controller disk drive, then press Return" (Operator Prompt)

| Note | If you are using a PC as the instrument controller, the disk format will have to be transformed from LIF to PC-DOS format. Once transformed, the PC now will have a DOS-formatted file to read and interpret. See the discussion in the previous example. |

Remove the disk from the analyzer’s disk file and place it into the controller disk drive.
The contents of this disk file are shown at the beginning of this example. The file records are read and then printed on the controller display. The file pointer is reset and the records are interpreted for their data contents.
The first 6 records are skipped for this example. The seventh record contains the stimulus frequency values and the number of points in the trace. These values are read from the record. The frequency increment, or point spacing, is calculated and used later to calculate the frequency data for a point. Two more records are skipped. The next record is the first of the data values. The data values are read in a loop until the number of points have been recovered from the file. The data values are tabulated and printed out on the controller display.
Index

1
1-port measurement calibration example program, 2-37

2
2-port measurement calibration program example, 2-41

A
abort message (IFC), 2-8
address capability, 2-5
addresses for HP-IB, 2-8
AH1 (full-acceptor handshake), 2-5
analyzer array-data formats, 2-18
analyzer bus mode, 2-7
analyzer command syntax, 2-10
analyzer control of peripherals, 2-7
analyzer data reading, 2-14
analyzer-debug mode, 2-31
analyzer features helpful in developing programs, 2-31
analyzer operation, 2-13
analyzer single bus concept, 2-6
analyzer status reporting structure, 2-24
analyzer system setups example program, 2-74
appendage in syntax, 2-11
array-data formats, 2-18
arrays of data, 2-21
arrays related to frequency, 2-20
ASCII disk files read to the instrument controller’s disk file example program, 2-108
ASCII transfer example program, 2-51
ATN (attention) control line, 2-4
attention (ATN) control line, 2-4

B
basic talker (T6), 2-5
bi-directional lines, 2-4
bus device modes, 2-6
bus structure, 2-2, 2-3

C
C10 (pass control capabilities), 2-5
C1, C2, C3 (system controller capabilities), 2-5
calibrating the test setup, 2-28
calibration coefficients, 2-21
calibration data reading example program, 2-76
calibration example program, 2-37
calibration kits, 2-37
calibration kit string and learn string, 2-23
calibration (power meter) example program, 2-69
chain for data processing, 2-21
characters that are valid, 2-11
clear device, 2-8
code naming conventions, 2-10
code syntax structure, 2-11
? command, 2-14
command formats, 2-11
command interrogate, 2-14
command syntax, 2-10
command syntax structure, 2-11
complete operation, 2-13
complete service request capabilities (SR1), 2-5
computer controllers, 2-3
connecting the device under test, 2-28
controlled sweep, 2-31
controller interface function, 2-3
control lines, 2-4
controlling peripherals using pass-control mode example program, 2-98
conventions for code naming, 2-10
correction of errors example program, 2-37

D
data-array formats, 2-18
data arrays, 2-21
data bus, 2-4
data for markers, 2-16
data formats and transfers, 2-48
data levels, 2-22
data-processing chain, 2-21
data rate, 2-5
data reading, 2-14
data taking, 2-29
data transfer, 2-4, 2-29
data-transfer character definitions, 2-15
data transfer example program, 2-48
data transfer for traces, 2-19
data transfer using floating-point numbers, 2-54
data transfer using form 11 example program, 2-60
data transfer using form 4 example program, 2-51
data transfer using frequency array information example program, 2-56
DC1 (complete device clear), 2-5
debug mode, 2-31
decibels terminator code, 2-11
definitions of status bit, 2-24
developing program features, 2-31
device clear, 2-8
device clear (DC1), 2-5
device connection, 2-28
device trigger, 2-9
device types for HP-IB, 2-2
disabling the front panel, 2-9
disk files read to the controller's disk file example program, 2-108
display format units, 2-16
does not respond to parallel poll (PPO), 2-5
D1 (responds to a group execute trigger), 2-5
E
E2 (tri-state drivers), 2-5
does not acknowledge (EOR) control line, 2-4
EOI (end or identify) control line, 2-4
error-corrected data, 2-21
error-correction example program, 2-37, 2-41
error output, 2-27
error queue use example program, 2-63
error reporting, 2-24
event-status register, 2-24, 2-27
example
1. measurement setup, 2-32
controlling peripherals using pass-control mode, 2-98
data transfer using floating-point numbers, 2-54
data transfer using form 1, 2-60
data transfer using form 4 (ASCII transfer), 2-51
data transfer using frequency array information, 2-56
data transfer using markers, 2-49
full 2-port measurement calibration, 2-41

generating interrupts, 2-65
limit-line testing, 2-83
measurement calibration, 2-37
measurement data transfer, 2-48
measurement process synchronization, 2-62
network analyzer system setups, 2-74
operation using talker/listener mode, 2-96
performing PASS/FAIL tests while tuning, 2-93
plotting plot files stored on disk, 2-106
plotting to a file and transferring the file data to a plotter, 2-103
power meter calibration, 2-69
printing plot files stored on disk, 2-106
printing with the serial port, 2-101
reading ASCII disk files to the instrument controller's disk file, 2-108
reading calibration data, 2-76
report generation, 2-96
S<sub>11</sub> 1-port measurement calibration, 2-37
saving and restoring the analyzer instrument state, 2-79
selecting a single segment from a table of segments, 2-86
setting parameters, 2-32
setting up limit lines, 2-89
storing and recalling instrument states, 2-74
system initialization, 2-32
using the error queue, 2-63
using the learn string, 2-74
verifying parameters, 2-35
examples of programs, 2-30
extended listener capabilities (LEO), 2-5
F
features helpful in developing programming routines, 2-31
femtoseconds terminator code, 2-11
floating-point numbers example program, 2-54
form 1 format, 2-18
form 2 format, 2-18
form 3 format, 2-18
form 4 data-transfer character definitions, 2-15
form 4 format, 2-18
form 5 format, 2-18
format display units, 2-16
formats and transfers of trace-data, 2-48
formats for array-data, 2-18
formats for commands, 2-11
formatted data, 2-21
frequency list mode example program, 2-83
frequency-related arrays, 2-20
full 2-port measurement calibration program
  example, 2-41
full-acceptor handshake (AH1), 2-5
full-source handshake (SH1), 2-5

G
general structure of syntax, 2-11
generating interrupts example program, 2-65
generation of report example program, 2-96
gigahertz terminator code, 2-11
group execute trigger response (DT1), 2-5
guidelines for code naming, 2-10

H
halting all modes and functions, 2-8
handshake lines, 2-4
held commands, 2-13
helpful features for developing programs,
  2-31
hertz terminator code, 2-11
HP-IB
  address capability, 2-5
  addresses, 2-8
  bus structure, 2-2, 2-3
  command formats, 2-11
  data rate, 2-5
  device types, 2-2
  message transfer scheme, 2-5
  meta-messages, 2-8
  multiple-controller capability, 2-5
  operation, 2-2
  operational capabilities, 2-5
  programming reference, 2-1
  requirements, 2-5
  status indicators, 2-6

I
IEEE-488 universal commands, 2-8
IEEE standard codes, formats, protocols
  information, 2-2
IEEE standard digital interface information,
  2-2
IFC (abort message), 2-8
IFC (interface clear) control line, 2-4
information on programs, 2-31
initializing the system, 2-32
instrument setup, 2-28
instrument state saving and restoring example
  program, 2-79
instrument state summary, 2-23
interface addresses, 2-8
interface clear (IFC) control line, 2-4
interface functions
  controller, 2-3

K
kilohertz terminator code, 2-11
kits of calibration standards, 2-37

L
LEO (no extended listener capabilities), 2-5
learn string and calibration kit string, 2-23
learn string use example program, 2-74
levels of data, 2-22
limit line setup example program, 2-80
limit-line testing example program, 2-83
limit-test array used to read values example
  program, 2-56
lines for control, 2-4
lines for handshaking, 2-4
listener interface function, 2-3
listen mode (L), 2-6
list-frequency mode example program, 2-83
L (listen mode), 2-6
local command (GTL), 2-8
local lockout command (LLO), 2-9

M
marker data, 2-16
markers used in data transfer example
  program, 2-49
measurement calibration example program,
  2-37, 2-41
measurement data post-processing, 2-29
measurement data taking, 2-29
measurement data transfer example program,
  2-48
measurement process, 2-28
measurement process synchronization
  example program, 2-62
measurement setup, 2-32
megahertz terminator code, 2-11
message transfer scheme, 2-5
meta-messages, 2-8
methods of HP-IB operation, 2-2
microseconds terminator code, 2-11
milliseconds terminator code, 2-11
modes
  analyzer bus, 2-7
diag, 2-31
pass-control, 2-7
system-controller, 2-6
talker/listener, 2-7
modes for bus device, 2-6
multiple-controller capability, 2-5

N
naming conventions, 2-10
nanoseconds terminator code, 2-11
network analyzer system setups example program, 2-74
no extended talker capabilities (TEO), 2-5
number of HP-IB devices allowed, 2-2
number of listeners allowed, 2-3

O
OPC-compatible commands, 2-13
operational capabilities for HP-IB, 2-5
operation complete, 2-13
operation of analyzer, 2-13
operation of HP-IB, 2-2
operation using talker/listener mode example program, 2-96
output-data command, 2-14
output of errors, 2-27
output queue, 2-14
output syntax, 2-15
outputting trace-related data, 2-16

P
parallel poll configure, 2-9
parallel poll non response (PPO), 2-5
parameter setting example program, 2-32
parameter verification example program, 2-35
pass control capabilities (C10), 2-5
pass-control mode, 2-7
pass control mode, 2-8
pass-control mode use in peripheral control example program, 2-98
PASS/FAIL tests example program, 2-93
PC-graphics applications example program, 2-105
performing PASS/FAIL tests while tuning example program, 2-93
peripheral addresses, 2-8
peripheral control using pass-control example program, 2-98
picoseconds terminator code, 2-11
plot file and PC-graphics example program, 2-105
plotting plot files stored on disk example program, 2-106
plotting a file and transferring the file data to a plotter example program, 2-103
post-processing the measurement data, 2-29
power meter calibration example program, 2-69
PPO (does not respond to parallel poll, 2-5
printing plot files stored on disk example program, 2-106
printing with the serial port example program, 2-101
processing after taking measurement data, 2-29
processing data chain, 2-21
process of measuring, 2-28
process synchronization example program, 2-62
program debugging, 2-31
program development features, 2-31
program example
1. measurement setup, 2-32
controlling peripherals using pass-control mode, 2-98
data transfer using floating-point numbers, 2-54
data transfer using form 1, 2-60
data transfer using form 4 (ASCII transfer), 2-51
data transfer using frequency array information, 2-56
data transfer using markers, 2-49
full 2-port measurement calibration, 2-41
generating interrupts, 2-65
limit-line testing, 2-83
measurement calibration, 2-37
measurement data transfer, 2-48
measurement process synchronization, 2-62
network analyzer system setups, 2-74
operation using talker/listener mode, 2-96
performing PASS/FAIL tests while tuning, 2-93
plotting plot files stored on disk, 2-106
plotting to a file and transferring the file data to a plotter, 2-103
power meter calibration, 2-69
printing plot files stored on disk, 2-106
printing with the serial port, 2-101
reading ASCII disk files to the instrument controller's disk file, 2-108
reading calibration data, 2-76
report generation, 2-96
S1, 1-port measurement calibration, 2-37
saving and restoring the analyzer instrument state, 2-79
selecting a single segment from a table of segments, 2-86
setting parameters, 2-32
setting up limit lines, 2-89
storing and recalling instrument states, 2-74
system initialization, 2-32
using the error queue, 2-63
using the learn string, 2-74
verifying parameters, 2-35
program information, 2-31
programming examples, 2-30

Q
queue for output, 2-14

R
raw measured data, 2-21
reading analyzer data, 2-14
reading calibration data example program, 2-76
recalling and storing example program, 2-74
reference of HP-IB programming, 2-1
remote enable (REN) control line, 2-4
remote/local capability (RL1), 2-5
remote mode, 2-9
remote operation (R), 2-6
REN (remote enable) control line, 2-4
report generation example generation, 2-96
reporting of errors, 2-24
reporting on status, 2-24
reporting status, 2-62
restoring and saving the analyzer instrument state example program, 2-79
RL1 (complete remote/local capability), 2-5
routing debugging, 2-31
R (remote operation), 2-6
rules for code naming, 2-10

S
S11 1-port measurement calibration example program, 2-37
saving and restoring the analyzer instrument state example program, 2-79
seconds terminator code, 2-11
segment selection from a table of segments example program, 2-86
selecting a segment from a table of segments example program, 2-86
serial poll, 2-9
serial port interface for printing example program, 2-101
service request asserted by the analyzer (S), 2-6
service request (SRQ) control line, 2-4
setting HP-IB addresses, 2-8
setting parameters example program, 2-32
setting up limit lines example program, 2-89
setting up the instrument, 2-28
SH1 (full-source handshake), 2-5
single bus concept, 2-5
SR1 (complete service request capabilities), 2-5
SRQ (service request) control line, 2-4
S (service request asserted by the analyzer), 2-6
status bit definitions, 2-24
status byte, 2-24, 2-26
status indicators, 2-6
status reporting, 2-24, 2-62
step 1 of a measurement, 2-28
step 2 of a measurement, 2-28
step 3 of a measurement, 2-28
step 4 of a measurement, 2-29
step 5 of a measurement, 2-29
step 6 of a measurement, 2-29
storing and recalling instrument states example program, 2-74
string for calibration kit, 2-23
structure of command syntax, 2-11
structure of HP-IB bus, 2-3
structure of status reporting, 2-24
sweep user-controlled, 2-31
synchronization of process example program, 2-62
syntax for commands, 2-10
syntax for output, 2-15
syntax structure, 2-11
syntax types, 2-12
system controller capabilities (C1,C2,C3), 2-5
system-controller mode, 2-6, 2-7
system initialization, 2-32
system setups example program, 2-74

T
T6 (basic talker), 2-5
take-control command, 2-9
taking the measurement data, 2-29
talker interface function, 2-2
talker/listener mode, 2-7
talker/listener mode operation example program, 2-96
talk mode (T), 2-6
TE0 (no extended talker capabilities), 2-5
terminators and units, 2-11
testing with limit lines example program, 2-83
test setup calibration, 2-28
tests that PASS/FAIL example program, 2-93
trace-data formats and transfers, 2-48
trace-data transfers, 2-19
trace memory, 2-21
trace-related data, 2-16
transfer of data, 2-4
transfer of data example program, 2-48
transfer of data using floating-point numbers example program, 2-54
transfer of data using form 11 example program, 2-60
transfer of data using form 4 example program, 2-51
transfer of data using frequency array information example program, 2-56
transferring file data to a plotter example program, 2-103
transferring plot-file data to a plotter example program, 2-106
transferring plot-file data to a printer example program, 2-106
transferring the measurement data, 2-29
transfers and formats of trace-data, 2-48
transfers of trace-data, 2-19
trigger device, 2-9

tri-state drivers (E2), 2-5
T (talk mode), 2-6
types of syntax, 2-12
U
units and terminators, 2-11
units as a function of display format, 2-16
universal commands, 2-8
user-controllable sweep, 2-31
V
valid characters, 2-11
verifying parameters example program, 2-35
volts terminator code, 2-11
W
waiting-for-group-execute-trigger, 2-9
waiting-for-reverse-get bit, 2-9
# Contents

<table>
<thead>
<tr>
<th>3. HP BASIC Programming Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction</strong></td>
</tr>
<tr>
<td><strong>Required Equipment</strong></td>
</tr>
<tr>
<td><strong>Optional Equipment</strong></td>
</tr>
<tr>
<td><strong>System Setup and HP-IB Verification</strong></td>
</tr>
<tr>
<td><strong>HP 8753D/Option 011 Network Analyzer Instrument Control Using BASIC</strong></td>
</tr>
<tr>
<td><strong>Command Structure in BASIC</strong></td>
</tr>
<tr>
<td><strong>Command Interrogate</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Held Commands</strong></td>
</tr>
<tr>
<td><strong>Operation Complete</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Preparing for Remote (HP-IB) Control</strong></td>
</tr>
<tr>
<td><strong>I/O Paths</strong></td>
</tr>
<tr>
<td><strong>BASIC Programming Examples</strong></td>
</tr>
<tr>
<td><strong>Example 1: Measurement Setup</strong></td>
</tr>
<tr>
<td><strong>Example 1A: Setting Parameters</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Example 1B: Verifying Parameters</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Example 2: Calibration</strong></td>
</tr>
<tr>
<td><strong>Performing a Measurement Calibration</strong></td>
</tr>
<tr>
<td><strong>Example 2A: S11 1-Port Calibration</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Example 2B: Full 2-Port Measurement Calibration</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Example 3: Measurement Data Transfer</strong></td>
</tr>
<tr>
<td><strong>Example 3A: Data Transfer Using Markers</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Example 3B: Data Transfer Using FORM 4 (ASCII Transfer)</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Example 3C: Data Transfer Using Floating-Point Numbers</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Example 3D: Data Transfer Using Frequency-Array Information</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Example 3E: Data Transfer Using FORM 1, Internal-Binary Format</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Example 4: Measurement Process Synchronization</strong></td>
</tr>
<tr>
<td><strong>Status Reporting</strong></td>
</tr>
<tr>
<td><strong>Example 4A: Using the Error Queue</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Example 4B: Generating Interrupts</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Example 4C: Power Meter Calibration</strong></td>
</tr>
<tr>
<td><strong>Running the Program</strong></td>
</tr>
<tr>
<td><strong>Example 5: Network Analyzer System Setups</strong></td>
</tr>
<tr>
<td>Example 6A: Using the Learn String</td>
</tr>
<tr>
<td>Running the Program</td>
</tr>
<tr>
<td>Example 5B: Reading Calibration Data</td>
</tr>
<tr>
<td>Running the Program</td>
</tr>
<tr>
<td>Example 5C: Saving and Restoring the Analyzer Instrument State</td>
</tr>
<tr>
<td>Running the Program</td>
</tr>
<tr>
<td>Example 6: Limit-Line Testing</td>
</tr>
<tr>
<td>Using List-Frequency Mode</td>
</tr>
<tr>
<td>Example 6A: Setting Up a List-Frequency Sweep</td>
</tr>
<tr>
<td>Running the Program</td>
</tr>
<tr>
<td>Example 6B: Selecting a Single Segment from a Table of Segments</td>
</tr>
<tr>
<td>Running the Program</td>
</tr>
<tr>
<td>Using Limit Lines to Perform PASS/FAIL Tests</td>
</tr>
<tr>
<td>Example 6C: Setting Up Limit Lines</td>
</tr>
<tr>
<td>Running the Program</td>
</tr>
<tr>
<td>Example 6D: Performing PASS/FAIL Tests While Tuning</td>
</tr>
<tr>
<td>Running the Program</td>
</tr>
<tr>
<td>Example 7: Report Generation</td>
</tr>
<tr>
<td>Example 7A1: Operation Using Talker/Listener Mode</td>
</tr>
<tr>
<td>Running the Program</td>
</tr>
<tr>
<td>Example 7A2: Controlling Peripherals Using Pass-Control Mode</td>
</tr>
<tr>
<td>Running the Program</td>
</tr>
<tr>
<td>Example 7A3: Printing with the Serial Port</td>
</tr>
<tr>
<td>Example 7B1: Plotting to a File and Transferring File Data to a Plotter</td>
</tr>
<tr>
<td>Running the Program</td>
</tr>
<tr>
<td>Plotting to a File and Utilizing PC-Graphics Applications</td>
</tr>
<tr>
<td>Example 7B2: Reading Plot Files From a Disk</td>
</tr>
<tr>
<td>Running the Program</td>
</tr>
<tr>
<td>Example 7C: Reading ASCII Disk Files to the System Controller Disk File</td>
</tr>
<tr>
<td>Running the Program</td>
</tr>
</tbody>
</table>

Index
Figures

3-1. The HP 8753D/Option 011 Network Analyzer System with Controller 3-2

Tables

3-1. Additional BASIC 6.2 Programming Information 3-1
3-2. Additional HP-IB Information 3-1
3-3. HP 8753D/Option 011 Network Analyzer Array-Data Formats 3-22
HP BASIC Programming Examples

Introduction

This is an introduction to the remote operation of the HP 8753D/Option 011 network analyzer using an HP 9000 Series 300 computer. It is a tutorial introduction using BASIC programming examples to demonstrate the remote operation of the network analyzer. The examples used in this chapter are on the "HP 8753D/Option 011 HP BASIC Programming Examples" disk.

The user should be familiar with the operation of the analyzer before attempting to remotely control the analyzer via the Hewlett-Packard Interface Bus (HP-IB). See the The HP 8753D/Option 011 Network Analyzer User's Guide for analyzer operating information.

The Hewlett-Packard computers specifically addressed in these examples are the HP 9000 Series 300 computers, operating with BASIC 6.2.

This document is not intended to teach BASIC programming or to discuss HP-IB theory except at an introductory level. For more information concerning BASIC, see the Table 3-1 for a list of manuals supporting the BASIC revision being used. For more information concerning the Hewlett-Packard Interface Bus, see Table 3-2.

Table 3-1. Additional BASIC 6.2 Programming Information

<table>
<thead>
<tr>
<th>Description</th>
<th>HP Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP BASIC 6.2 Programming Guide</td>
<td>98615-90010</td>
</tr>
<tr>
<td>HP BASIC 6.2 Language Reference (2 Volumes)</td>
<td>98615-90004</td>
</tr>
<tr>
<td>Using HP BASIC for Instrument Control, Volume I</td>
<td>82303-90001</td>
</tr>
<tr>
<td>Using HP BASIC for Instrument Control, Volume II</td>
<td>82303-90002</td>
</tr>
</tbody>
</table>

Table 3-2. Additional HP-IB Information

<table>
<thead>
<tr>
<th>Description</th>
<th>HP Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP BASIC 6.2 Interface Reference</td>
<td>98616-90013</td>
</tr>
<tr>
<td>Tutorial Description of the Hewlett-Packard Interface Bus</td>
<td>5021-1927</td>
</tr>
</tbody>
</table>

Required Equipment

To run the examples in this chapter, in addition to the HP 8753D/Option 011 network analyzer, the following equipment is required:

- Computer ................................................................................. HP 9000 Series 300
- BASIC operating system ......................................................... BASIC 6.2
- "HP 8753D/Option 011 HP BASIC Programming Examples" disk .......... 08753-10028
- HP-IB interconnect cables ..................................................... HP 10833A/B/C/D
- Test device ............................................................................. such as a 125 MHz bandpass filter
Note
The HP 9000 Series 300 computer must have enough memory to store:
- BASIC 6.2 (4 MBytes of memory is required)
- the required binaries

Upon receipt, make copies of the "HP 8753D/Option 011 Programming Examples" disks. Label them "HP 8753D/Option 011 Programming Examples BACKUP". These disks will act as reserves in the event of loss or damage to the original disks.

Optional Equipment
- 50Ω type-N calibration kit .................................. HP 85032B
- Test port return cables ................................... HP 11852D
- Plotter ........................................................ HP 7440A ColorPro
- Printer .......................................................... HP 2225A ThinkJet
- Disk drive .................................................... HP 9122 or HP 9153 CS80

See the “Compatible Peripherals” chapter in the HP 8753D/Option 011 Network Analyzer User’s Guide for a more complete list of compatible peripherals.

System Setup and HP-IB Verification

This section describes how to:
- Connect the test system.
- Set the test system addresses.
- Set the network analyzer’s control mode.
- Verify the operation of the system’s interface bus (HP-IB).

Figure 3-1. The HP 8753D/Option 011 Network Analyzer System with Controller
1. Connect the analyzer to the computer with an HP-IB cable as shown in Figure 3-1.

2. Switch on the computer.

3. Load the BASIC 6.2 operating system.

4. Switch on the analyzer.

   a. To verify the analyzer’s address, press:

   
   ![Address Set](image)

   The analyzer has only one HP-IB interface, though it occupies two addresses: one for the instrument and one for the display. The display address is equal to the instrument address with the least-significant bit incremented. The display address is automatically set each time the instrument address is set.

   The default analyzer addresses are:

   - 16 for the instrument
   - 17 for the display

   **Caution** Other devices connected to the bus cannot occupy the same address as the analyzer.

   The analyzer displays the instrument’s address in the upper right section of the display. If the address is not 16, return the address to its default setting (16) by pressing:

   ![Address Reset](image)

   b. Set the system control mode to either “pass-control” or “talker/listener” mode. These are the only control modes in which the analyzer will accept commands over HP-IB. For more information on control modes, see Chapter 2, “HP-IB Programming Reference.” To set the system-control mode, press:

   ![Mode Set](image)

   or

   ![Mode Set](image)

5. Check the interface bus by performing a simple command from the computer controller.

   Type the following command on the controller:

   ![Command](image)

   **Note** HP 9000 Series 300 computers use the (RETURN) key as both execute and enter. Some other computers may have an (ENTER), (EXECUTE), or (EXEC) key that performs the same function. For reasons of simplicity, the notation (RETURN) is used throughout this chapter.

   This command should preset the analyzer. If an instrument preset does not occur, there is a problem. Check all HP-IB addresses and connections. Most HP-IB problems are caused by an incorrect address and faulty/loose HP-IB cables.
HP 8753D/Option 011 Network Analyzer Instrument Control Using BASIC

A remote controller can manipulate the functions of the analyzer by sending commands to the analyzer via the Hewlett-Packard Interface Bus (HP-IB). The commands used are specific to the analyzer. Remote commands executed over the bus take precedence over manual commands executed from the instrument’s front panel. Remote commands are executed as soon as they are received by the analyzer. A command only applies to the active channel (except in cases where functions are coupled between channels). Most commands are equivalent to front-panel hardkeys and softkeys.

Command Structure in BASIC

Consider the BASIC command for setting the analyzer’s start frequency to 10 MHz:

```
OUTPUT 716;"STAR 10 MHZ;"
```

The command structure in BASIC has several different elements:

- the BASIC command statement
- the appendage
- data
- unit
- terminator

<table>
<thead>
<tr>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTPUT</td>
<td>The BASIC data-output statement.</td>
</tr>
<tr>
<td>716</td>
<td>The data is directed to interface 7 (HP-IB), and on to the device at address 16 (the HP 8753D/Option 011). This appendage is terminated with a semicolon. The next appendage is STAR, the instrument mnemonic for setting the analyzer’s start frequency.</td>
</tr>
<tr>
<td>10</td>
<td>a single operand used by the root mnemonic STAR to set the value.</td>
</tr>
<tr>
<td>MHZ</td>
<td>the units that the operand is expressed in.</td>
</tr>
<tr>
<td>;</td>
<td>indicates the end of a command, enters the data, and deactivates the active-entry area.</td>
</tr>
</tbody>
</table>

The "STAR 10 MHZ;" command performs the same function as pressing the following keys on the analyzer’s front panel:

```
START 10 MHZ
```

STAR is the root mnemonic for the start key, 10 is the data, and MHZ are the units. Where possible, the analyzer’s root mnemonics are derived from the equivalent key label. Otherwise they are derived from the common name for the function. Chapter 1, “Command Reference,” lists all the root mnemonics and all the different units accepted.

The semicolon (;) following MHZ terminates the command within the analyzer. It removes start frequency from the active-entry area, and prepares the analyzer for the next command. If there is a syntax error in a command, the analyzer will ignore the command and look for the next terminator. When it finds the next terminator, it starts processing incoming commands normally. Characters between the syntax error and the next terminator are lost. A line feed also acts as a terminator. The BASIC OUTPUT statement transmits a carriage return/line feed following the data. This can be suppressed by putting a semicolon at the end of the statement.
The OUTPUT 716; statement will transmit all items listed (as long as they are separated by commas or semicolons) including:

- literal information enclosed in quotes,
- numeric variables,
- string variables,
- and arrays.

A carriage return/line feed is transmitted after each item. Again, this can be suppressed by terminating the commands with a semicolon. The analyzer automatically goes into remote mode when it receives an OUTPUT command from the controller. When this happens, the front-panel remote (R) and listen (L) HP-IB status indicators illuminate. In remote mode, the analyzer ignores any data that is input with the front-panel keys, with the exception of LOCAL. Pressing LOCAL returns the analyzer to manual operation, unless the universal HP-IB command LOCAL LOCKOUT 7 has been issued. There are two ways to exit from a local lockout. Either issue the LOCAL 7 command from the controller or cycle the line power on the analyzer.

Setting a parameter such as start frequency is just one form of command the analyzer will accept. It will also accept simple commands that require no operand at all. For example, execute:

```
OUTPUT 716;"AUTO;"
```

In response, the analyzer autoscales the active channel. Autoscale only applies to the active channel, unlike start frequency, which applies to both channels as long as the channels are stimulus-coupled.

The analyzer will also accept commands that switch various functions ON and OFF. For example, to switch on dual-channel display, execute:

```
OUTPUT 716;"DUACION;"
```

DUACION is the analyzer root mnemonic for "dual-channel display on". This causes the analyzer to display both channels. To go back to single-channel display mode, for example, switch off dual-channel display, execute:

```
OUTPUT 716;"DUACOFF;"
```

The construction of the command starts with the root mnemonic DUAC (dual-channel display,) and ON or OFF is appended to the root to form the entire command.

The analyzer does not distinguish between upper- and lower-case letters. For example, execute:

```
OUTPUT 716;"auto;"
```

**Note**

The analyzer also has a debug mode to aid in troubleshooting systems. When the debug mode is ON, the analyzer scrolls incoming HP-IB commands across the display. To manually activate the debug mode, press LOCAL HP-IB DIAG ON. To deactivate the debug mode from the controller, execute:

```
OUTPUT 716;"DEBUGOFF;"
```
Command Interrogate

Suppose the operator has changed the power level from the front panel. The computer can find the new power level using the analyzer’s command-interrogate function. If a question mark is appended to the root of a command, the analyzer will output the value of that function.

For instance, **POWE 7 DB**; sets the analyzer’s output power to 7 dB, and **POWE?**; outputs the current RF output power at the test port to the system controller. For example:

Type **SCRATCH** and press **RETURN** to clear old programs.

Type **EDIT** and press **RETURN** to access the edit mode.

Then type in:

```
10 OUTPUT 716;"POWE?;"
20 ENTER 716; Reply
30 DISP Reply
40 END
```

Running the Program

The computer will display the source-power level in dBm. The preset source-power level is 0 dBm. Change the power level by pressing **LOCAL MENU POWER 1 5 2**. Now run the program again.

When the analyzer receives **POWE?**, it prepares to transmit the current RF source-power level. The **BASIC** statement **ENTER 716** allows the analyzer to transmit information to the computer by addressing the analyzer to talk. This illuminates the analyzer front-panel talk (T) light. The computer places the data transmitted by the analyzer into the variables listed in the **ENTER** statement. In this case, the analyzer transmits the output power, which gets placed in the variable **Reply**.

The **ENTER** statement takes the stream of binary-data output from the analyzer and reformats it back into numbers and ASCII strings. With the formatting set to its default state, the **ENTER** statement will format the data into real variables, integers, or ASCII strings, depending on the variable being filled. The variable list must match the data the analyzer has to transmit. If there are not enough variables, data is lost. If there are too many variables for the data available, a **BASIC** error is generated.

The formatting done by the **ENTER** statement can be changed. For more information on data formatting, see Chapter 2, “HP-IB Programming Reference” under the section titled “Array Data Formats.” The formatting can be deactivated to allow binary transfers of data. Also, the **ENTER USING** statement can be used to selectively control the formatting.

**ON/OFF** commands can also be interrogated. The reply is a one (1) if the function is active, a zero (0) if it is not active. Similarly, if a command controls a function that is underlined on the analyzer softkey menu when active, interrogating that command yields a one (1) if the command is underlined, a zero (0) if it is not. For example, press **MEAS**. Though there are seven options on the measurement menu, only one is underlined at a time. The underlined option will return a one (1) when interrogated.

For instance, rewrite line 10 as:

```
10 OUTPUT 716;"DUAC?;"
```

Run the program once and note the result. Then press **LOCAL DISPLAY DUAL CHAN** to toggle the display mode, and run the program again.
Another example is to rewrite line 10 as:

```
10 OUTPUT 716;"PHAS?;"
```

In this case, the program will display a one (1) if phase is currently being displayed. Since the command only applies to the active channel, the response to the PHAS? inquiry depends on which channel is active.

**Held Commands**

When the analyzer is executing a command that cannot be interrupted, it will not process any new HP-IB commands until the previous command has completed. These "uninterruptable" commands are called "held" commands. During the processing of a held command, the analyzer will fill the 16-character input buffer, and then halt all HP-IB operation until the held command has completed execution.

**Note**

This action will be transparent to a program unless HP-IB timeouts have been set with the ON TIMEOUT statement.

While a held command is executing, the analyzer will still service the HP-IB interface commands, such as SPOIL(716), CLEAR 716, and ABORT 7. Executing CLEAR 716 or CLEAR 7 will abort a hold off, leaving the held command to complete execution as if it had been input from the front panel. These commands also clear the input buffer, destroying any commands received after the held command. If the analyzer has halted the bus because its input buffer was full, ABORT 7 will release the bus.

**Operation Complete**

Occasionally, there is a need to query the analyzer as to when certain analyzer operations have completed. For instance, a program should not have the operator connect the next calibration standard while the analyzer is still measuring the current one. To provide such information, the analyzer has an "operation complete" reporting mechanism, or OPC command, that will indicate when certain key commands have completed operation. The mechanism is activated by sending either OPC or OPC? immediately before an OPC-compatible command. When the command completes execution, bit 0 of the event-status register will be set. If OPC was interrogated with OPC?, the analyzer will also output a one (1) when the command completes execution.

As an example, type SCRATCH and press RETURN.

Type EDIT and press RETURN.

Type in the following program:

```
10 OUTPUT 716;"SWET 3 S;OPC?;SING;"

20 DISP "Sweeping"
30 ENTER 716;Reply

40 DISP "DONE"
50 END
```

- **Set the sweep time to 3 seconds, and OPC a single sweep.**
- **The program will halt at this point until the analyzer completes the sweep and issues a one (1).**
Running the Program

Running this program causes the computer to display the sweeping message as the instrument executes the sweep. The computer will display DONE just as the instrument goes into hold. When DONE appears, the program could then continue on, being assured that there is a valid data trace in the instrument.

Preparing for Remote (HP-IB) Control

At the beginning of a program, the analyzer is taken from an unknown state and brought under remote control. This is done with an abort/clear sequence. ABORT 7 is used to halt bus activity and return control to the computer. CLEAR 716 will then prepare the analyzer to receive commands by:

- clearing syntax errors
- clearing the input-command buffer
- clearing any messages waiting to be output

The abort/clear sequence readies the analyzer to receive HP-IB commands. The next step involves programming a known state into the analyzer. The most convenient way to do this is to preset the analyzer by sending the PRES (preset) command. If preset cannot be used, the status-reporting mechanism may be employed. When using the status-reporting register, CLES (Clear Status) can be transmitted to the analyzer to clear all of the status-reporting registers and their enables.

Type SCRATCH and press [RETURN].
Type EDIT and press [RETURN].
Type in the following program:

```
10 ABORT 7
20 CLEAR 716

30 OUTPUT 716,"PRES;"

40 END
```

This halts all bus action and gives active control to the computer.
This clears all HP-IB errors, resets the HP-IB interface, and clears the syntax errors. It does not affect the status-reporting system.

Presets the instrument. This clears the status-reporting system, as well as resets all of the front-panel settings, except for the HP-IB mode and the HP-IB addresses.

Running this program brings the analyzer to a known state, ready to respond to HP-IB control.

The analyzer will not respond to HP-IB commands unless the remote line is asserted. When the remote line is asserted, the analyzer is addressed to listen for commands from the controller. In remote mode, all the front-panel keys are disabled (with the exception of LOCAL) and the line-power switch. ABORT 7 asserts the remote line, which remains asserted until a LOCAL 7 statement is executed. Another way to assert the remote line is to execute:

```
REMOTE 716
```

This statement asserts the analyzer's remote-operation mode and addresses the analyzer to listen for commands from the controller. Press any front-panel key except LOCAL. Note that none of the front-panel keys will respond until LOCAL has been pressed.

LOCAL can also be disabled with the sequence:

```
REMOTE 716
LOCAL LOCKOUT 7
```
After executing the code above, none of front-panel keys will respond. The analyzer can be returned to local mode temporarily with:

   LOCAL 716

As soon as the analyzer is addressed to listen, it goes back into local-lockout mode. The only way to clear the local-lockout mode, aside from cycling line power, is to execute:

   LOCAL 7

This command un-asserts the remote line on the interface. This puts the instrument into local mode and clears the local-lockout command. Return the instrument to remote mode by pressing:

   LOCAL TALKER/LISTENER

or

   LOCAL USE PASS CONTROL

I/O Paths

One of the features of HP BASIC is the use of input/output paths. The instrument may be addressed directly by the instrument's device number as shown in the previous examples. However, a more sophisticated approach is to declare I/O paths such as: ASSIGN @Nwa TO 716. Assigning an I/O path builds a look-up table in the computer's memory that contains the device-address codes and several other parameters. It is easy to quickly change addresses throughout the entire program at one location. I/O operation is more efficient because it uses a table, in place of calculating or searching for values related to I/O. In the more elaborate examples where file I/O is discussed, the look-up table contains all the information about the file. Execution speed is increased, because the computer no longer has to calculate a device's address each time that device is addressed.

For example:

Type SCRATCH and press \textbf{RETURN}.

Type EDIT and press \textbf{RETURN}.

Type in the following program:

\begin{verbatim}
10 ASSIGN @Nwa TO 716
20 OUTPUT @Nwa;"STAR 10 MHZ;"
\end{verbatim}

\textit{Assigns the analyzer to ADDRESS 716.}

\textit{Sets the analyzer's start frequency to 10 MHz.}

\textbf{Note}

The use of I/O paths in binary-format transfers allows the user to quickly distinguish the type of transfer taking place. I/O paths are used throughout the examples and are highly recommended for use in device input/output.
BASIC Programming Examples

The following sample programs provide the user with factory-tested solutions for several remotely-controlled analyzer processes. The programs can be used in their present state or modified to suit specific needs. The programs discussed in this section can be found on the “HP 8753D/Option 011 HP BASIC Programming Examples” disk received with the analyzer.

Example 1: Measurement Setup

The programs included in Example 1 provide the user the option to perform instrument-setup functions for the analyzer from a remote controller. Example 1A is a program designed to setup the analyzer’s measurement parameters. Example 1B is a program designed to verify the measurement parameters.

Example 1A: Setting Parameters

Note This program is stored as EXAMP1A on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

In general, the procedure for setting up measurements on the network analyzer via HP-IB follows the same sequence as if the setup was performed manually. There is no required order, as long as the desired frequency range, number of points, and power level are set prior to performing the calibration first, and the measurement second.

The following is an outline of the program’s processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The analyzer is adjusted to measure return loss on channel 1 and display it in log magnitude.
- The analyzer is adjusted to measure return loss on channel 2 and display the phase.
- The dual-channel display mode is activated.
- The system operator is prompted to enter the frequency range of the measurement.
- The displays are autoscaled.
- The analyzer is released from remote control and the program ends.
The program is written as follows:

10 ! This program selects the S-parameter to be measured, the display
20 ! format and then sets the specified start and stop frequencies.
30 ! The analyzer display is then autoscaled.
40 !
50 ! EXAMPLE
60 !
70 ASSIGN @Nwa TO 716                      ! Assign an I/O path for the analyzer
80 !
90 CLEAR SCREEN
100 ! Initialize the system
110 ABORT 7                                ! Generate an IFC (Interface Clear)
120 CLEAR @Nwa                              ! SDC (Selected Device Clear) analyzer
130 OUTPUT @Nwa:"OPC?;PRES;"               ! Preset the analyzer and wait
140 ENTER @Nwa;Reply                        ! Read in the 1 returned
150 !
160 ! Set up measurement and display
170 OUTPUT @Nwa:"CHAN1;"                    ! Channel 1
180 OUTPUT @Nwa:"S11;"                     ! Return Loss measurement
190 OUTPUT @Nwa:"LOGM;"                    ! Log magnitude display
200 !
210 OUTPUT @Nwa:"CHAN2;"                    ! Channel 2
220 OUTPUT @Nwa:"S11;"                     ! Return Loss measurement
230 OUTPUT @Nwa:"PHAS;"                    ! Phase display
240 !
250 OUTPUT @Nwa:"DUALCON;"                 ! Dual channel display
260 !
270 ! Request start and stop frequency
280 INPUT "ENTER START FREQUENCY (MHz):",F_start
290 INPUT "ENTER STOP FREQUENCY (MHz):",F_stop
300 !
310 ! Program the analyzer settings
320 OUTPUT @Nwa:"STAR";F_start;"MHZ;"       ! Set the start frequency
330 OUTPUT @Nwa:"STOP";F_stop;"MHZ;"        ! Set the stop frequency
340 !
350 ! Autoscale the displays
360 OUTPUT @Nwa:"CHAN1;AUTO;"              ! Autoscale channel 1 display
370 OUTPUT @Nwa:"CHAN2;AUTO;"              ! Autoscale channel 2 display
380 !
390 OUTPUT @Nwa:"OPC?;WAIT;"               ! Wait for the analyzer to finish
400 ENTER @Nwa;Reply                        ! Read the 1 when complete
410 LOCAL @Nwa                             ! Release HP-IB control
420 END

Running the Program

The analyzer is initialized and the operator is queried for the measurement's start and stop frequencies. The analyzer is setup to display the S11 reflection measurement as a function of log magnitude and phase over the selected frequency range. The displays are autoscaled and the program ends.
Example 1B: Verifying Parameters

This program is stored as EXAMP1B on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

This example shows how to read analyzer settings into your program. Chapter 1, "Command Reference," contains additional information on the command formats and operations. Appending a "?" to a command that sets an analyzer parameter will return the value of that setting. Parameters that are set as ON or OFF when queried will return a zero (0) if OFF or a one (1) if active. Parameters are returned in ASCII format, FORM 4. This format is varying in length from 1 to 24 characters-per-value. In the case of marker or other multiple responses, the values are separated by commas.

The following is an outline of the program's processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The number of points in the trace is queried and dumped to a printer.
- The start frequency is queried and output to a printer.
- The averaging is queried and output to a printer.
- The analyzer is released from remote control and the program ends.

The program is written as follows:

```
10 ! This program performs some example queries of network analyzer
20 ! settings. The number of points in a trace, the start frequency
30 ! and if averaging is turned on, are determined and displayed.
40 !
50 ! EXAMP1B
60 !
70 ASSIGN @Nwa TO 716 ! Assign an I/O path for the analyzer
80 !
90 CLEAR SCREEN
100 ! Initialize the system
110 ABORT 7 ! Generate an IFC (Interface Clear)
120 CLEAR @Nwa ! SDC (Selected Device Clear)
130 OUTPUT @Nwa:"OPC?;PRES;"
140 ENTER @Nwa;Reply ! Preset the analyzer and wait
150 !
160 ! Query network analyzer parameters
170 OUTPUT @Nwa:"POIN?;"
180 ENTER @Nwa;Num_points ! Read in the default trace length
190 PRINT "Number of points ";Num_points
200 PRINT
210 !
220 OUTPUT @Nwa:"STAR?;"
230 ENTER @Nwa;Start_f
240 PRINT "Start Frequency ";Start_f
250 PRINT
260 !
270 OUTPUT @Nwa:"AVERO?;"
280 ENTER @Nwa;Flag
```

3-12 HP BASIC Programming Examples
290 PRINT "Flag =";Flag;" ";                ! Test flag and print analyzer state
300 IF Flag=1 THEN
310 PRINT "Averaging ON"
320 ELSE
330 PRINT "Averaging OFF"
340 END IF
350 !
360 OUTPUT @Nwa;"OPC?;WAIT;"
370 ENTER @Nwa;Reply
380 LOCAL @Nwa
390 END

Running the Program

The analyzer is preset. The preset values are returned and printed out for: the number of
points, the start frequency, and the state of the averaging function. The analyzer is released
from remote control and the program ends.
Example 2: Calibration

Performing a Measurement Calibration
This section will demonstrate how to perform a measurement calibration over the HP-IB. The HP-IB command sequence follows the key sequence required to perform a front-panel calibration. There is a command for every step of the sequence.

The remote measurement calibration sequence has three steps:

- selecting the calibration
- measuring the calibration standards
- declaring the calibration finished

The actual sequence depends on the calibration kit and changes slightly for 2-port calibrations, which are divided into three calibration sub-sequences.

Example 2A is a program designed to perform an S11 1-port calibration. Example 2B is a program designed to perform a full 2-port measurement calibration.

Example 2A: S11 1-Port Calibration

Note

This program is stored as EXAMP2A on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

The following program performs an S11 1-port calibration, using the HP 85032B 500 type-N calibration kit. This program simplifies the calibration by providing explicit directions on the analyzer display while allowing the user to run the program from the controller keyboard. More information on selecting calibration standards can be found in the Optimizing Measurement Results chapter of the HP 8753D/Option 011 Network Analyzer User’s Guide.

The following is an outline of the program’s processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The appropriate calibration kit is selected.
- The softkey menu is deactivated.
- The S11-calibration sequence is run.
- The S11-calibration data is saved.
- The softkey menu is activated.
- The analyzer is released from remote control and the program ends.
The program is written as follows:

10 ! This program performs a 1-port calibration on the HP 8753.LOAD "EXAMP2A"
20 ! It guides the operator through a 1-port calibration
30 ! using the HP 85032B 50 ohm type N calibration kit.
40 !
50 ! The routine Waitforkey displays a message on the instrument's
60 ! display and the console, to prompt the operator to connect the
70 ! calibration standard. Once the standard is connected, the
80 ! ENTER key on the computer keyboard is pressed to continue.
90 !
100 ! EXAMP2A
110 !
120 ASSIGN @Nwa TO 716 ! Assign an I/O path for the analyzer
130 !
140 CLEAR SCREEN
150 ! Initialize the system
160 ABORT 7 ! Generate an IFC (Interface Clear)
170 CLEAR @Nwa ! SDC (Selected Device Clear)
180 !
190 OUTPUT @Nwa:"CALKN50;" ! Select CAL kit type
200 OUTPUT @Nwa:"MENUOFF;" ! Turn softkey menu off.
210 !
220 OUTPUT @Nwa:"CALIS111;" ! 511 1 port CAL initiated
230 !
240 CALL Waitforkey("CONNECT OPEN AT PORT 1")
250 OUTPUT @Nwa:"CLASS11A;" ! Open reflection CAL
260 OUTPUT @Nwa:"OPC?;STANB;" ! Select the second standard, B
270 ENTER @Nwa;Reply ! Read in the 1 returned
280 !
290 CALL Waitforkey("CONNECT SHORT AT PORT 1")
300 OUTPUT @Nwa:"CLASS11B;" ! Short reflection CAL
310 OUTPUT @Nwa:"OPC?;STANB;" ! Select the second standard, B
320 ENTER @Nwa;Reply ! Read in the 1 returned
330 !
340 CALL Waitforkey("CONNECT LOAD AT PORT 1")
350 OUTPUT @Nwa:"OPC?;CLASS11C;" ! Reflection load CAL
360 ENTER @Nwa;Reply ! Read in the 1 returned
370 !
380 OUTPUT 717;"PG;" ! Clear the analyzer display
390 !
400 DISP "COMPUTING CALIBRATION COEFFICIENTS"
410 !
420 OUTPUT @Nwa;"DONE;" ! Finished with the CAL cycle
430 OUTPUT @Nwa;"OPC?;SAV1;" ! Save the ONE PORT CAL
440 ENTER @Nwa;Reply ! Read in the 1 returned
450 !
460 DISP "S11 1-PORT CAL COMPLETED. CONNECT TEST DEVICE."
470 OUTPUT @Nwa;"MENUON;" ! Turn on the softkey menu
480 !
490 OUTPUT @Nwa;"OPC?;WAIT;" ! Wait for the analyzer to finish
500 ENTER @Nwa;Reply ! Read the 1 when complete
510 LOCAL @Nwa ! Release HP-IB control
520 !
530 END
540 !
550 ! ********************************** Subroutines **********************************
560 !
570 Waitforkey:  ! Prompt routine to read a keypress on the controller
580 SUB Waitforkey(Lab$)
590 !  Position and display text on the analyzer display
600 OUTPUT 717;"PG;PU;PA390,3700;PD;LB";Lab$;" , PRESS ENTER WHEN READYA;"
610 !
620 DISP Lab$;&" Press ENTER when ready";  ! Display prompt on console
630 INPUT A$  ! Read ENTER key press
640 !
650 OUTPUT 717;"PG;"  ! Clear analyzer display
660 SUBEND

Running the Program

Note: This program does not modify the instrument state in any way. Before running the program, set up the desired instrument state.

Run the program and connect the standards as prompted. When a standard is connected, press the ENTER on the controller keyboard to measure it.

The program assumes that the port being calibrated is a 500 type-N female test port. The prompts appear just above the message line on the analyzer display. Pressing RETURN on the controller keyboard continues the program and measures the standard. The program will display a message when the measurement calibration is complete.

Example 2B: Full 2-Port Measurement Calibration

Note: This program is stored as EXAMP2B on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

The following example program performs a full 2-port measurement calibration using the HP 86032B 500 type-N calibration kit. A full 2-port calibration removes both the forward- and reverse-error terms so all four S-parameters of the device under test can be measured. PORT 1 is a female test port and PORT 2 is a male test port.

The following is an outline of the program's processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The appropriate calibration kit is selected.
- The softkey menu is deactivated.
- The 2-port calibration sequence is run.
- The operator is prompted to choose or skip the isolation calibration.
- The softkey menu is activated.
- The analyzer is released from remote control and the program ends.
The program is written as follows:

10 ! This program performs a full 2-port measurement calibration.
20 ! It guides the operator through a full 2-port calibration
30 ! using the HP 85032B 50 ohm type N calibration kit.
40 ! The routine Waitforkey displays a message on the instrument's
50 ! display and the console to prompt the operator to connect the
60 ! calibration standard. Once the standard is connected, the
70 ! ENTER key on the computer keyboard is pressed to continue.
80 !
90 ! EXAMP2B
100 !
110 ASSGN @Nwa TO 716 ! Assign an I/O path to the analyzer
120 !
130 CLEAR SCREEN
140 ! Initialize the analyzer
150 ABORT 7 ! Generate an IFC (Interface Clear)
160 CLEAR @Nwa ! SDC (Selected Device Clear)
170 !
180 OUTPUT @Nwa;"CALKN50;MENUTOF;" ! Select CAL kit type and turn off menu
190 !
200 OUTPUT @Nwa;"CALIFUL2;" ! Full 2 port CAL
210 !
220 OUTPUT @Nwa;"REFL;" ! Reflection CAL
230 !
240 CALL Waitforkey("CONNECT OPEN AT PORT 1")
250 OUTPUT @Nwa;"CLASS11A;" ! S11 open CAL
260 OUTPUT @Nwa;"OPC;STANDB;" ! Select the second standard, B
270 ENTER @Nwa;Reply ! Read in the 1 returned
280 !
290 CALL Waitforkey("CONNECT SHORT AT PORT 1")
300 OUTPUT @Nwa;"CLASS11B;" ! S11 short CAL
310 OUTPUT @Nwa;"OPC;STANDB;" ! Select the second standard, B
320 ENTER @Nwa;Reply ! Read in the 1 returned
330 !
340 CALL Waitforkey("CONNECT LOAD AT PORT 1")
350 OUTPUT @Nwa;"OPC;CLASS11C;" ! S11 load CAL
360 ENTER @Nwa;Reply ! Read in the 1 returned
370 !
380 CALL Waitforkey("CONNECT OPEN AT PORT 2")
390 OUTPUT @Nwa;"CLASS22A;" ! S22 open CAL
400 OUTPUT @Nwa;"OPC;STANA;" ! Select the first standard, A
410 ENTER @Nwa;Reply ! Read in the 1 returned
420 !
430 CALL Waitforkey("CONNECT SHORT AT PORT 2")
440 OUTPUT @Nwa;"CLASS22B;" ! S22 short CAL
450 OUTPUT @Nwa;"OPC;STANA;" ! Select the first standard, A
460 ENTER @Nwa;Reply ! Read in the 1 returned
470 !
480 CALL Waitforkey("CONNECT LOAD AT PORT 2")
490 OUTPUT @Nwa;"OPC;CLASS22C;" ! S22 load CAL
500 ENTER @Nwa;Reply
510 !
520 DISP "COMPUTING REFLECTION CALIBRATION COEFFICIENTS"
530 !
540 OUTPUT @Nwa:"REFD;"                   ! Reflection portion complete
550 !
560 OUTPUT @Nwa:"TRAN;"                   ! Transmission portion begins
570 !
580 CALL Waitforkey("CONNECT THRU [PORT1 TO PORT 2]")
590 DISP "MEASURING FORWARD TRANSMISSION"
600 OUTPUT @Nwa:"OPC?;FWDT;"            ! Measure forward transmission
610 ENTER @Nwa;Reply                     ! Read in the 1 returned
620 !
630 OUTPUT @Nwa:"OPC?;FWDM;"            ! Measure forward load match
640 ENTER @Nwa;Reply                     ! Read in the 1 returned
650 !
660 DISP "MEASURING REVERSE TRANSMISSION"
670 OUTPUT @Nwa:"OPC?;REVT;"            ! Measure reverse transmission
680 ENTER @Nwa;Reply                     ! Read in the 1 returned
690 !
700 OUTPUT @Nwa:"OPC?;REVM;"            ! Measure reverse load match
710 ENTER @Nwa;Reply                     ! Read in the 1 returned
720 !
730 OUTPUT @Nwa:"TRAD;"                 ! Transmission CAL complete
740 !
750 INPUT "SKIP ISOLATION CAL? Y OR N ",An$   ! Skip isolation cal
760 IF An$="Y" THEN
770 OUTPUT @Nwa:"OMII;"                 ! Skip isolation cal
780 GOTO 940
790 END IF
800 !
810 CALL Waitforkey("ISOLATE TEST PORTS")
820 !
830 OUTPUT @Nwa:"ISOL;"                  ! Isolation CAL
840 OUTPUT @Nwa:"AVERFACT10;"           ! Average for 10 sweeps
850 OUTPUT @Nwa:"AVEROON;"              ! Turn on averaging
860 DISP "MEASURING REVERSE ISOLATION"
870 OUTPUT @Nwa:"OPC?;REVI;"            ! Measure reverse isolation
880 ENTER @Nwa;Reply                     ! Read in the 1 returned
890 !
900 DISP "MEASURING FORWARD ISOLATION"
910 OUTPUT @Nwa:"OPC?;FWD1;"            ! Measure forward isolation
920 ENTER @Nwa;Reply                     ! Read in the 1 returned
930 !
940 OUTPUT @Nwa:"ISOD;AVEEROFF;"        ! Isolation complete averaging off
950 OUTPUT 717;"PG;"                     ! Clear analyzer display prompt
960 !
970 DISP "COMPUTING CALIBRATION COEFFICIENTS"
980 OUTPUT @Nwa;"DONE;"                 ! End the CAL sequence
990 OUTPUT @Nwa;"OPC?;SAV2;"            ! Save THE TWO PORT CAL
1000 ENTER @Nwa;Reply                   ! Read in the 1 returned
1010!
1020 DISP "DONE WITH FULL 2-PORT CAL. CONNECT TEST DEVICE."
1030 OUTPUT @Nwa;"MENON;"               ! Turn softkey menu on
1040 !
1050 OUTPUT @Nwa;"OPC?;WAIT;"           ! Wait for the analyzer to finish
1060 ENTER @Nwa;Reply                   ! Read the 1 when complete
1070 LOCAL @Nwa                         ! Release HP-IB control
1080!
1090 END
1100!
1110! **************************** Subroutines ****************************
1120!
1130 SUB Waitforkey(Lab$)
1140  ! Position and display prompt on the analyzer display
1150  OUTPUT 717;"PG;PU;PA390,3700;PD;LB";Lab$;". PRESS ENTER WHEN READYA;"
1160  !
1170 DISP Lab$&"  Press ENTER when ready";  ! Display prompt on console
1180 INPUT A$  ! Read ENTER keypress on controller
1190 OUTPUT 717;"PG;"  ! Clear analyzer display
1200 SUBEND

Running the Program

Note
Before running the program, set the desired instrument state. This program does not modify the instrument state in any way.

Run the program and connect the standards as prompted. After the standard is connected, press [ENTER] on the controller keyboard to continue the program. The program assumes that the test ports being calibrated are 500 type-N, PORT 1 being a female test port and PORT 2 being a male test port. The HP 85032B 500 type-N Calibration Kit is used. The prompts appear just above the message line on the analyzer display. After the prompt is displayed, pressing [ENTER] on the computer console continues the program and measures the standard. The operator has the option of omitting the isolation calibration. If the isolation calibration is performed, averaging is automatically employed to insure a good calibration. The program will display a message when the measurement calibration is complete.
Example 3: Measurement Data Transfer

Trace information can be read out of the analyzer in several ways. Data can be read from the trace selectively using the markers, or the entire trace can be read out. If only specific information such as a single point on the trace or the result of a marker search is required, the marker output command can be used to read the information. If all the trace data is required, see Examples 3B through 3E.

Example 3A: Data Transfer Using Markers

Note: This program is stored as EXAMP3A on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

Markers are the simplest form of trace-data transfer. A marker may be positioned using one of three methods:

- by a frequency location
- by an actual data point location
- by a trace-data value

In the following example, the marker is positioned on the trace's maximum value. Once positioned on the trace, the trace data at that point can be read into the controller. The marker data is always returned in FORM 3, ASCII format. Each number is sent as a 24-character string. Characters can be digits, signs, or decimal points. All characters should be separated by commas. In the case of markers, three numbers are sent. The display format determines the values of the marker responses. See Table 3-3, "HP 8753D/Option 011 Network Analyzer Array-Data Formats."

When using trace data, it is important to control the network analyzer's sweep function (and therefore the trace data) from the computer. Using the computer to control the instrument's sweep insures that the data you read into the controller is in a quiescent or steady state. It also insures that the measurement is complete.

The following is an outline of the program's processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The selected frequency span is swept once.
- The marker is activated and placed on the maximum trace value.
- The three marker values are output to the controller and displayed.
- The instrument is returned to local control and the program ends.
The program is written as follows:

10 ! This program takes a sweep on the analyzer and turns on a marker.
20 ! The marker is positioned on the trace maximum and the marker data
30 ! is output in ASCII format.
40 !
50 ! EXAMP3A
60 !
70 ASSIGN @Nwa TO 716 ! Assign an I/O path for the analyzer
80 !
90 CLEAR SCREEN
100 ! Initialize the analyzer
110 ABORT 7 ! Generate an IPC (Interface Clear)
120 CLEAR @Nwa ! SDC (Selective Device Clear)
130 OUTPUT @Nwa;"OPC?;PRES;"
140 ENTER @Nwa;Reply ! Preset the analyzer and wait
150 !
160 OUTPUT @Nwa;"OPC?;SING"
170 ENTER @Nwa;Reply ! Read in the 1 returned
180 !
190 OUTPUT @Nwa;"MARK1;"
200 OUTPUT @Nwa;"SEAMAX;"
210 !
220 OUTPUT @Nwa;"OUTPMARK;"
230 ENTER @Nwa;Value1,Value2,Stim ! Request the current marker value
240 !
250 ! Show the marker data received.
260 PRINT " Value 1"," Value 2"," Stimulus (Hz)"
270 PRINT Value1,Value2,Stim ! Print the received values
280 PRINT
290 PRINT " Compare the active marker block with the received values"
300 !
310 LOCAL @Nwa ! Release HP-IB control
320 END

Running the Program

Run the program. The analyzer is preset and a sweep is taken. Marker 1 is enabled and positioned on the largest value in the trace. The marker is output to the controller and printed on the controller display. The analyzer is returned to local control. Position the marker using the RPG or data-entry keys, and compare the displayed value on the analyzer with the value that was transmitted to the controller.
Example 3B: Data Transfer Using FORM 4 (ASCII Transfer)

**Note**
This program is stored as EXAMP3B on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

<table>
<thead>
<tr>
<th>Format type</th>
<th>Type of Data</th>
<th>Bytes per Data Value</th>
<th>Bytes-per-point 2 data values</th>
<th>201 Bytes-per-trace</th>
<th>Total Bytes add header</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORM 1</td>
<td>Internal Binary</td>
<td>3</td>
<td>6</td>
<td>1205</td>
<td>1210</td>
</tr>
<tr>
<td>FORM 2</td>
<td>IEEE 32-bit Floating-Point</td>
<td>4</td>
<td>8</td>
<td>1608</td>
<td>1612</td>
</tr>
<tr>
<td>FORM 3</td>
<td>IEEE 64-bit Floating-Point</td>
<td>8</td>
<td>16</td>
<td>3216</td>
<td>3220</td>
</tr>
<tr>
<td>FORM 4</td>
<td>ASCII Numbers</td>
<td>24</td>
<td>50</td>
<td>10,050</td>
<td>10,050*</td>
</tr>
<tr>
<td>FORM 5</td>
<td>PC-DOS 32-bit Floating-Point</td>
<td>4</td>
<td>8</td>
<td>1205</td>
<td>1220</td>
</tr>
</tbody>
</table>

*No header is used in FORM 4.

The next most common data transfer is to transfer a trace array from the analyzer. Table 3-3 shows the relationship of the two values-per-point that are transferred to the analyzer. When FORM 4 is used, each number is sent as a 24-character string, each character represented by a digit, sign, or decimal point. Each number is separated from the previous number with a comma. Since there are two numbers-per-point, a 201-point transfer in FORM 4 takes 10,050 bytes. This form is useful only when input-data formatting is difficult with the instrument controller. Refer to Table 3-3 for a comparison with the other formats.

An example of a simple data transfer using FORM 4 (ASCII data transfer) is shown in this program. A fairly common requirement is to create frequency-amplitude data pairs from the trace data. No frequency information is included with the trace data transfer, because the frequency data must be calculated. Relating the data from a linear frequency sweep to frequency can be done by interrogating the analyzer start frequency, the frequency span, and the number of points in the sweep. Given that information, the frequency of point N in a linear frequency sweep is:

\[ F = \text{Start frequency} + (N-1) \times \frac{\text{Span}}{\text{Points}-1} \]

Example 3B illustrates this technique. It is a straight-forward solution for linear uniform sweeps. For other sweep types, frequency data is more difficult to construct and may best be read directly from the analyzer’s limit-test array. See Example 3D for an explanation of this technique.
The following is an outline of the program's processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The trace-data array is allocated.
- The trace length is set to 11.
- The selected frequency span is swept once.
- The FORM 4, ASCII format is set.
- The formatted trace is read from the analyzer and displayed.
- The frequency increments between the points are calculated.
- The marker is activated and placed at 30 kHz.
- The instrument is returned to local control and the program ends.
The program is written as follows:

10 ! This program shows an ASCII format trace data transfer using form 4.
20 ! The data is received as a string of ASCII characters, 24 characters
30 ! per data point and transferred into a real array in the controller. The
35 ! corresponding frequency data is calculated from the analyzer settings.
40 !
45 ! EXAMPLE
50 !
55 ASSIGN @Nwa TO 716 ! Assign an I/O path to the analyzer
60 !
70 ! Initialize
80 CLEAR SCREEN
90 !
100 ABORT 7
110 CLEAR @Nwa
120 OUTPUT @Nwa:"OPC?;PRES;"
130 ENTER @Nwa;Reply
140 ! Generate an IFC (Interface Clear)
150 ! SDC (Selective Device Clear)
160 !
170 ! Trace values are two elements per point, display format dependent
180 DIM Dat(1;11,1;2) ! Trace data array
190 !
200 OUTPUT @Nwa:"P0IN 11;"
210 OUTPUT @Nwa:"OPC?;SING;"
220 ENTER @Nwa;Reply
230 !
240 OUTPUT @Nwa:"FORM4;"
250 OUTPUT @Nwa:"OUTPFORM;"
260 ENTER @Nwa;Dat(*)
270 !
280 ! Now to calculate the frequency increments between points
290 OUTPUT @Nwa:"P0IN?:"
300 ENTER @Nwa;Num_points
310 OUTPUT @Nwa:"STAR?:"
320 ENTER @Nwa;Startf
330 OUTPUT @Nwa:"SPAN?;"
340 ENTER @Nwa;Span
350 !
360 F_inc=Span/(Num_points-1) ! Calculate fixed frequency increment
370 !
380 PRINT "Point","Freq (MHz)"," Value 1"," Value 2"
390 IMAGE 3D,7X,5D.3D,3X,3D.3D,3X,3D.3D ! Formatting for controller display
400 !
410 FOR I=1 TO Num_points ! Loop through data points
420 Freq=Startf+(I-1)*F_inc ! Calculate frequency of data point
430 PRINT USING 390;I,Freq/1.E+6,Dat(I,1),Dat(I,2)! Print analyzer data
440 NEXT I
450 !
460 OUTPUT @Nwa:"MARKDISC;"
470 OUTPUT @Nwa:"MARK1 .3E+6;" ! Discrete marker mode
480 !
490 OUTPUT @Nwa:"OPC?;WAIT;"
500 ENTER @Nwa;Reply
510 LOCAL 7 ! Position marker at 30 KHz
520 !
530 PRINT

3-24 HP BASIC Programming Examples
540 PRINT "Position the marker with the knob and compare the values"
550 !
560 END

Running the Program

Run the program and watch the controller console. The analyzer will perform an instrument preset. The program will then print out the data values received from the analyzer. The marker is activated and placed at the left-hand edge of the analyzer's display. Position the marker with the knob and compare the values read with the active marker with the results printed on the controller console. The data points should agree exactly. Keep in mind that no matter how many digits are displayed, the analyzer is specified to measure:

- magnitude to a resolution of 0.001 dB
- phase to a resolution of 0.01 degrees
- group delay to a resolution of 0.01 ps

Example 3C: Data Transfer Using Floating-Point Numbers

This program is stored as EXAMP3C on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

This example program illustrates data transfer using FORM 3 in which data is transmitted in the floating-point formats. FORM 2 is nearly identical except for the IEEE 32-bit format of 4 bytes-per-value. FORM 5 reverses the order of the bytes to conform with the PC conventions for defining a real number.

The block-data formats have a four-byte header. The first two bytes are the ASCII characters "#A" that indicate that a fixed-length block transfer follows, and the next two bytes form an integer containing the number of bytes in the block to follow. The header must be read in so that data order is maintained.

This transfer is more than twice as fast than a FORM 4 transfer. With the FORM 4 transfer, 10,050 bytes are sent (201 points × 2 values-per-point × 24 bytes-per-value). Using FORM 2 to transfer the data, only 1612 bytes are sent (201 points × 2 values-per-point × 4 bytes-per-value). See "Array-Data Formats."

The following is an outline of the program's processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The integer variables are defined to contain the header information.
- The number of points in the trace is set to 11.
- The selected frequency span is swept once.
- Data-transfer format 3 is set.
- The headers are read from the trace.
- The array size is calculated and allocated.
- The trace data is read in and printed.
- The marker is activated and placed at 30 kHz.
- The instrument is returned to local control and the program ends.
The program is written as follows:

10 ! This program shows how to read in a data trace in IEEE 64 bit
20 ! format. The array header is used to determine the length of the
30 ! array and to allocate the array size.
40 !
50 ! Program Example 3C
60 !
70 CLEAR SCREEN
80 ! Initialize the analyzer
90 ASSIGN @Nwa TO 716
100 ASSIGN @Nwadat TO 716;FORMAT OFF ! Assign an I/O path for the analyzer
110 ! Binary data path definition
120 ABORT 7
130 CLEAR @Nwa
140 OUTPUT @Nwa;"OPC?;PRES;"
150 ENTER @Nwa;Reply ! Generate an IFC (Interface Clear)
160 ! SDC (Selected Device Clear)
170 INTEGER Dheader,Dlength
180 Numpoints=11 ! Integer variables for header info
190 OUTPUT @Nwa;"POIN";Numpoints;";" ! Number of points in the trace
200 ! Set number of points in trace
210 ! Set up data transfer
220 OUTPUT @Nwa;"OPC?;SING"
230 ENTER @Nwa;Reply ! Single sweep and wait
240 !
250 OUTPUT @Nwa;"FORM3;"
260 OUTPUT @Nwa;"OUTPFORM;"
270 ! Send formatted output trace
280 ENTER @Nwadat;Dheader,Dlength ! Read headers from trace data
290 !
300 ALLOCATE Dat(1:Dlength/16,1:2) ! Use length to determine array size
310 ENTER @Nwadat;Dat(*) ! Read in trace data
320 !
330 PRINT "Size of array ";Dlength/16;" elements"
340 PRINT "Number of bytes ";Dlength
350 !
360 ! Print out the data array
370 PRINT "Element","Value 1","Value 2"
380 IMAGE 3D,6X,3D.6D,6X,3D.6D
390 FOR I=1 TO Numpoints ! Loop through the data points
400 PRINT USING 380;I,Dat(I,1),Dat(I,2)
410 NEXT I
420 !
430 OUTPUT @Nwa;"MARKDISC;" ! Discrete marker mode
440 OUTPUT @Nwa;"MARK1 .3E+6;" ! Position marker at 30 KHz
450 !
460 OUTPUT @Nwa;"OPC?;WAIT;" ! Wait for the analyzer to finish
470 ENTER @Nwa;Reply ! Read the I when complete
480 LOCAL @Nwa ! Release HP-IB control
490 !
500 PRINT
510 PRINT "Position the marker with the knob and compare the values."
520 !
530 END
Running the Program

Run the program. The computer displays the number of elements and bytes associated with the transfer of the trace, as well as the first 10 data points. Position the marker and examine the data values. Compare the displayed values with the analyzer's marker values.

Example 3D: Data Transfer Using Frequency-Array Information

Note  
This program is stored as EXAMP3D on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

Example 3C was used to read in the trace-data array. Example 3D explains how to use the limit-test array to read in the when corresponding frequency values for the completed trace array. The analyzer is set to sweep from 10 MHz to 200 MHz in log-frequency mode with the number of points in the trace set to 11. This makes it very difficult to compute the frequency-point spacing in the trace. The points are equally spaced across the trace, but not equally spaced in relation to frequency (because the frequency span is displayed in a logarithmic scale, as opposed to a linear scale). The limit-test data array may be read from the analyzer to provide the frequency values for each data point. All four values-per-trace location must be read from the analyzer. The test results and limit values are not used in this example. Only the frequency values are used. This technique is the only method of obtaining the non-linear frequency data from the analyzer’s display. The test data and frequencies are printed on the controller display and the marker enabled to allow the operator to examine the actual locations on the analyzer’s display.

The following is an outline of the program's processing sequence:
- An I/O path is assigned for the analyzer.
- The system is initialized.
- The integer variables for the header information are defined.
- The number of points in the trace is set to 11.
- The frequency span (10 MHz to 200 MHz) is selected.
- The log-frequency sweep is selected.
- The data-transfer format 3 is set.
- The headers are read from the trace.
- The array size is calculated and allocated.
- The trace data is read in.
- The limit-test array is calculated and allocated.
- The limit-line test array is read in.
- The table header is printed.
- The program cycles through the trace values.
- The trace data and frequency are printed.
- The discrete-marker mode is activated.
- The marker is activated and placed at 10 MHz.
- The instrument is returned to local control and the program ends.
The program is written as follows:

10 ! This program shows how to read in a trace and create the frequency
20 ! value associated with the trace data value. EXAMP3C is used to
30 ! read in the data from the analyzer. The start and stop
40 ! frequencies are set to provide two decades of log range. Log sweep
50 ! is set and the frequency data points are read from the limit test
60 ! array and displayed with the data points.
70 !
80 ! EXAMP3D
90 !
100 ASSIGN @Nwa TO 716 ! Assign an I/O path for the analyzer
110 ASSIGN @Nwadat TO 716;FORMAT OFF ! Binary path for data transfer
120 !
130 CLEAR SCREEN
140 ! Initialize the analyzer
150 ABORT 7
160 CLEAR @Nwa
170 OUTPUT @Nwa;"OPC?;PRES;"
180 ENTER @Nwa;Reply
190 !
200 INTEGER Dheader,Dlength
210 !
220 OUTPUT @Nwa;"POIN 11;"
230 OUTPUT @Nwa;"STAR 10.E+6;"
240 OUTPUT @Nwa;"STOP 200.E+6;"
250 OUTPUT @Nwa;"LOGFREQ;"
260 !
270 ! Set up data transfer
280 OUTPUT @Nwa;"OPC?;SING"
290 ENTER @Nwa;Reply
300 !
310 OUTPUT @Nwa;"FORM3;"
320 OUTPUT @Nwa;"OUTPFORM;"
330 !
340 ENTER @Nwadat;Dheader,Dlength
350 !
360 ALLOCATE Dat(1:Dlength/16,1:2)
370 ENTER @Nwadat;Dat(*)
380 !
390 ! Create the corresponding frequency values for the array
400 !
410 ! Read the frequency values using the limit test array
420 ALLOCATE Freq(1:Dlength/16,1:4) ! Limit line results array
430 ! Limit line values are frequency, test results, upper and lower limits
440 !
450 OUTPUT @Nwa;"OUTPLIML;"
460 ENTER @Nwa;Freq(*)
470 !
480 ! Display table of freq and data
490 !
500 PRINT " Freq (MHz)","Mag (dB)"
510 FOR I=1 TO 11
520 Freqm=Freq(I,1)/1.E+6 ! Convert frequency to MHz
530 PRINT USING "4D.6D,9X,3D.3D";Freqm,Dat(I,1) ! Print trace data
540 NEXT I
550 !
560 ! Set up marker to examine frequency values
570 OUTPUT @Nwa:"MARKDISC;" ! Discrete marker mode
580 OUTPUT @Nwa:"MARK1 10.E+6;" ! Turn on marker and place at 10 MHz
590 !
600 OUTPUT @Nwa:"OPC?;WAIT;" ! Wait for the analyzer to finish
610 ENTER @Nwa;Reply  ! Read the 1 when complete
620 LOCAL @Nwa  ! Release HP-IB control
630 PRINT  ! Blank line
640 PRINT "Position marker and observe frequency point spacing"
650 !
660 END

Running the Program

Run the program. Observe the controller display. The corresponding frequency values are shown with the trace-data values. Position the marker and observe the relationship between the frequency values and the point spacing on the trace. Compare the trace-data values on the analyzer with those shown on the controller display.

Example 3E: Data Transfer Using FORM 1, Internal-Binary Format

Note This program is stored as EXAMP3E on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

FORM 1 is used for rapid I/O transfer of analyzer data. It contains the least number of bytes-per-trace and does not require re-formatting in the analyzer. This format is more difficult to convert into a numeric array in the controller. Analyzer-state information, such as learn strings and calibration arrays, may be easily transferred in this format because data conversion is not required. Recalling an instrument state that has been stored in a file and transferring instrument state information to the analyzer are excellent applications of a FORM 1 data transfer.

The following is an outline of the program's processing sequence:

■ An I/O path is assigned for the analyzer.
■ The system is initialized.
■ The integer variables for the header information are defined.
■ The string variable for the header is defined.
■ The selected frequency span is swept once.
■ The internal-binary format is selected.
■ The error-corrected data is output from the analyzer.
■ The two data-header characters and the two length bytes are read in.
■ The string buffer is allocated for data.
■ The trace data is read into the string buffer.
■ The analyzer is restored to continuous-sweep mode and queried for command completion.
■ The instrument is returned to local control and the program ends.
The program is written as follows:

```
10 ! This program is an example of a form 1, internal format data
20 ! transfer. The data is stored in a string dimensioned to the
30 ! length of the data being transferred.
40 !
50 ! EXAMP3E
60 !
70 ASSIGN @Nwa TO 716               ! Assign an I/O path for the analyzer
80 ASSIGN @Nwa_bin TO 716;FORMAT OFF    ! Binary path for data transfer
90 !
100 CLEAR SCREEN
110 ! Initialize the analyzer
120 ABORT 7               ! Send IFC Interface Clear
130 CLEAR @Nwa             ! SDC (Selective Device Clear)
140 OUTPUT @Nwa:"DPC?=PRES;"! Preset the analyzer and wait
150 ENTER @Nwa;Reply       ! Read the 1 when completed
160 !
170 INTEGER Length       ! Header length 2 bytes
180 DIM Header$[2]        ! Header string 2 bytes
190 !
200 OUTPUT @Nwa:"DPC?=SING;"! Single sweep and wait
210 ENTER @Nwa;Reply      ! Read the 1 when completed
220 !
230 OUTPUT @Nwa:"FORM1;"   ! Select internal binary format
240 OUTPUT @Nwa:"OUTDATA;"! Output error corrected data
250 !
260 ! Read in the data header two characters and two bytes for length
270 ! "#,2A"
280 !                     # no early termination, terminate when ENTER is complete
290 !                     2A read two chars
300 !
310 ENTER @Nwa_bin USING ",2A";Header$  ! Read header as 2 byte string
320 ENTER @Nwa_bin;Length       ! Read length as 2 byte integer
330 PRINT "Header ";Header$,"Array length";Length
340 !
350 ALLOCATE Data$[Length]    ! String buffer for data bytes
360 ! "+,-,K" format statement
370 ! EOI as a terminator LF is suppressed and read as data
380 ! -K All characters are read and not interpreted LF is included
390 ENTER @Nwa_bin USING "+,-,K";Data$  ! Read trace into string array
400 !
410 PRINT "Number of bytes received ";LEN(Data$)
420 !
430 OUTPUT @Nwa;"CONT;"       ! Restore continuous sweep
440 OUTPUT @Nwa:"DPC?=WAIT;"  ! Wait for the analyzer to finish
450 ENTER @Nwa;Reply          ! Read the 1 when complete
460 !
470 LOCAL @Nwa               ! Release HP-IB control
480 END
```

Running the Program

The analyzer is initialized. The header and the number of bytes in the block transfer are printed on the controller display. Once the transfer is complete, the number of bytes in the data string is printed. Compare the two numbers to be sure that the transfer was completed.
Example 4: Measurement Process Synchronization

Status Reporting

The analyzer has a status-reporting mechanism that provides information about specific internal analyzer functions and events. The status byte is an 8-bit register with each bit summarizing the state of one aspect of the instrument. For example, the error-queue summary bit will always be set if there are any errors in the queue. The value of the status byte can be read with the HP-IB serial-poll operation. Serial Poll is described in Chapter 2, “HP-IB Programming Reference” under the section titled “Response to HP-IB Meta-Message (IEEE-488 Universal Commands).” This command does not automatically put the instrument in remote mode, thus giving the operator access to the analyzer front-panel functions. The status byte can also be read by sending the command OUTPSTAT. Reading the status byte does not affect its value.

The status byte also summarizes two event-status registers that monitor specific conditions within the instrument. The status byte also has a bit that is set when the instrument is issuing a service request over HP-IB and a bit that is set when the analyzer has data to send out over HP-IB. See “Error Reporting” in Chapter 2 of this guide for a discussion of the event-status registers.

Example 4A: Using the Error Queue

Note: This program is stored as EXAMP4A on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

The error queue holds up to 20 instrument errors and warnings in the order that they occurred. Each time the analyzer detects an error condition, the analyzer displays a message on the CRT, and puts the error in the error queue. If there are any errors in the queue, bit 3 of the status byte will be set. The errors can be read from the queue with the OUTPERRO command. OUTPERRO causes the analyzer to transmit the error number and message of the oldest error in the queue.

The following is an outline of the program’s processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The error-message string is allocated.
- The analyzer is released from remote control.
- The program begins an endless loop to read the error queue.
- The status byte is read with a serial poll.
- The program tests to see if an error is present in the queue.
- The error-queue bit is set.
- The program requests the content of the error queue.
- The error number and string are read.
- The error messages are printed until there are no more errors in the queue.
- The instrument is returned to local control.
- The controller emits a beep to attach the attention of the operator and resumes searching for errors.
The program is written as follows:

```
10 ! This program is an example of using the error queue to detect
20 ! errors generated by the analyzer. The status byte is read and
30 ! bit 3 is tested to determine if an error exists. The error queue
40 ! is printed out and emptied.
50 !
60 ! EXAMP4A
70 !
80 ASSIGN @Nwa TO 716
90 !
100 CLEAR SCREEN
110 ! Initialize the analyzer
120 ABORT 7
130 CLEAR @Nwa
140 OUTPUT @Nwa;"OPC?;PRES;"
150 ENTER @Nwa;Reply
160 !
170 DIM Error$[50]
180 !
190 LOCAL @Nwa
200 !
210 LOOP
220 REPEAT
230 Stat=SPOLL(@Nwa)
240 UNTIL BIT(Stat,3)
250 !
260 ! Error queue bit is set
270 REPEAT
280 OUTPUT @Nwa;"OUTPERRO;"
290 ENTER @Nwa;Err,Error$
300 PRINT Err,Error$
310 UNTIL Err=0
320 !
330 LOCAL @Nwa
340 BEEP 600,.2
350 END LOOP
360 !
370 END
```

**Running the Program**

Run the program. The analyzer goes through the preset cycle. Nothing will happen at first. The program is waiting for an error condition to activate the error queue. To cause an error, press a blank softkey. The message **CAUTION: INVALID KEY** will appear on the analyzer. The computer will beep and print out two error messages. The first line will be the invalid key error message, and the second line will be the **NO ERRORS** message. To clear the error queue, you can either loop until the **NO ERRORS** message is received, or until the bit in the status register is cleared. In this case, we wait until the status bit in the status register is clear. Note that while the program is running, the analyzer remains in the local mode and the front-panel keys may be accessed.

The error queue will hold up to 20 errors until all the errors are read out or the instrument is preset. It is important to clear the error queue whenever errors are detected. Otherwise, old errors may be mistakenly associated with the current instrument state. Press **SYSTEM**. Next,
press the unlabeled key several times quickly and watch the display. The number of errors observed should correspond to the number of times you pressed the key.

As another example, press CAL. Then the CALIBRATE MENU key. Select the RESPONSE calibration. Press DONE RESPONSE without performing any calibrations. Note the error message on the analyzer and on the controller display. Push the THROUGH key and then DONE RESPONSE. We are not concerned with the validity of the calibration, just setting a simple calibration on the analyzer. Note that COR is displayed in the upper left-hand section of the graticule. Now, press START and 7. This will generate an error because the start frequency has been changed, invalidating the calibration. This error is reported on the controller display as well. A complete list of error messages and their descriptions can be found in Chapter 10 of the HP 8753D/Option 011 Network Analyzer User’s Guide.

The program is in an infinite loop waiting for errors to occur. End the program by halting it with RESET or BREAK.

Note Not all messages displayed by the analyzer are put in the error queue: operator prompts and cautions are not included.

Example 4B: Generating Interrupts

Note This program is stored as EXAMP4B on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

It is also possible to generate interrupts using the status-reporting mechanism. The status-byte bits can be enabled to generate a service request (SRQ) when set. In turn, the instrument controller can be set up to generate an interrupt on the SRQ and respond to the condition which caused the SRQ.

To generate an SRQ, a bit in the status byte is enabled using the command SREN. A one (1) in a bit position enables that bit in the status byte. Hence, SRE 8 enables an SRQ on bit 3, the check-error queue, since the decimal value 8 equals 00001000 in binary representation. Whenever an error is put into the error queue and bit 3 is set, the SRQ line is asserted, illuminating the (S) indicator in the HP-IB status block on the front panel of the analyzer. The only way to clear the SRQ is to disable bit 3, re-enable bit 3, or read out all the errors from the queue.

A bit in the event-status register can be enabled so that it is summarized by bit 5 of the status byte. If any enabled bit in the event-status register is set, bit 5 of the status byte will also be set. For example ESE 66 enables bits 1 and 6 of the event-status register, since in binary, the decimal number 66 equals 01000010. Hence, whenever active control is requested or a front-panel key is pressed, bit 5 of the status byte will be set. Similarly, ESWEn enables bits in event-status register B so that they will be summarized by bit 2 in the status byte.

To generate an SRQ from an event-status register, enable the desired event-status register bit. Then enable the status byte to generate an SRQ. For instance, ESE 32; SRE 32; enables the syntax-error bit. When the syntax-error bit is set, the summary bit in the status byte will be set. This will, in turn, enable an SRQ on bit 5 of the status byte, the summary bit for the event-status register.

The following is an outline of the program’s processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
The status registers are cleared.

The event-status register bit 5 is enabled.

The status-register bit 5 is enabled.

The interrupt pointer is enabled and points to a subroutine.

A bad command is sent to the analyzer to generate errors.

The controller reads a serial-poll byte from HP-IB in the event of an interrupt.

The program tests for an SRQ.

If the SRQ is not generated by the analyzer, the subroutine stops and displays SRQ FROM OTHER DEVICE.

If the SRQ was generated by the analyzer, the program reads the status byte and event-status register.

If bit 5 in the event-status register is set, program prints: SYNTAX ERROR FROM ANALYZER.

If bit 5 in the event-status register is NOT set, program prints: SYNTAX ERROR BIT NOT SET.

The SRQ interrupt is re-enabled on the bus.

At the finish, the interrupt is deactivated.

The analyzer is released from remote control and the program ends.

The program is written as follows:

```
10 ! This program is an example of using an SRQ based interrupt to
20 ! detect an error condition in the analyzer. In this example, a
30 ! syntax error is generated with an invalid command. The status byte
40 ! is read in and tested. The error queue is read, printed out and
50 ! then cleared.
60 !
70 ! EXAMP4B
80 !
90 ASSIGN @Nwa T0 716 ! Assign an I/O path for the analyzer
100 !
110 CLEAR SCREEN
120 ! Initialize the analyzer
130 ABORT 7 ! Generate and IFC (Interface Clear)
140 CLEAR @Nwa ! SDC (Selective Device Clear)
150 OUTPUT @Nwa,"DPC?;PRES;"
160 ENTER @Nwa;Reply ! Preset the analyzer and wait
170 !
180 DIM Error$[50] ! Read the one from the analyzer
190 ! Set up syntax error interrupt
200 OUTPUT @Nwa, "CLES;"
210 ! Generate SRQ when bit 5 is set
220 !
230 OUTPUT @Nwa,"ESE 32;" ! Clear the status registers
240 !
250 ! Generate bit 5 in status register when syntax error occurs
260 OUTPUT @Nwa,"SRE 32;" ! Event status register bit 5 enabled
270 !
280 ! Setup the interrupt pointer to a subroutine
290 ON INTR 7 GOSUB Srq_det ! When interrupt occurs go to Srq_det
300 Stat=SPOLL(@Nwa) ! Clear any pending SRQs
```

3-34 HP BASIC Programming Examples
310  ENABLE INTR 7;2  ; Set interrupt on HP-IB bit 2 (SRQ)
320  !
330  DISP "Waiting for bad syntax"  ; Pause for 2 seconds
340  WAIT 2
350  !
360  OUTPUT @Nwa;"STIP 2GHZ;;"
370  ; Send bad STOP command syntax
380  WAIT 2
390  DISP ""
400  GOTO Finish
410  !
420  !**************************** Subroutines ******************************
430  !
440  Srq_det:
450  Stat=SPOLL(@Nwa)  ; Read serial poll byte from HP-IB
460  PRINT "Stat from Serial Poll";Stat
470  IF BIT(Stat,6) THEN
480  PRINT "SRQ received from analyzer"
490  ELSE
500  PRINT "SRQ from other device"
510  STOP
520  END IF
530  !
540  IF BIT(Stat,5) THEN  ; Event status register bit set
550  PRINT "Event Status Register caused SRQ"
560  ELSE
570  PRINT "Some other bit caused the SRQ"
580  STOP
590  END IF
600  !
610  REPEAT
620  OUTPUT @Nwa;"OUTPERRO;"
630  ENTER @Nwa;Err,Error$  ; Read analyzer error queue
640  PRINT Err,Error$  ; Read error number and string
650  UNTIL Err=0
660  !
670  PRINT
680  ENABLE INTR 7;2  ; Re-enable SRQ interrupt on HP-IB
690  RETURN
700  !
710  !**************************** End Subroutines ******************************
720  !
730  Finish:
740  DISP "Finished"
750  OFF INTR 7
760  LOCAL @Nwa
770  END
Running the Program

Run the program. The computer will preset the analyzer, then pause for a second or two. After pausing, the program sends an invalid command string "STIP 2 GHz;" to cause a syntax error. This command is intended to be "STOP 2 GHz;". The computer will display a series of messages from the SRQ-handler routine. The analyzer will display CAUTION: SYNTAX ERROR and the incorrect command, pointing to the first character it did not understand.

The SRQ can be cleared by reading the event-status register and clearing the latched bit, or by clearing the enable registers with CLES. The syntax-error message on the analyzer display can only be cleared by the HP-IB Device Clear (DCL) message or Selected Device Clear (SDC) message. Device Clear is not commonly used because it clears every device on the bus. Selected Device Clear can be used to reset the input and output queue and the registers of a specific instrument on the bus. This will also clear all the interrupt definitions.

Syntax errors are created when a command cannot be interpreted by the input parser of the analyzer. The controller releases the analyzer from remote control and the program ends.

Example 4C: Power Meter Calibration

Note
This program is stored as EXAMP4C on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

For increased accuracy during calibration of the 8753D's PORT 1-output power, a power meter calibration is available. This measurement-accuracy enhancement technique is described in the Optimizing Measurement Results chapter of the HP 8753D/Option 011 Network Analyzer User's Guide. The example described will perform the same and sweep calibration under HP-IB remote control.

The power meter is usually connected to PORT 1 for the forward measurements. Its address must be set correctly and it must be connected to the HP-IB. The power meter address can be set by pressing: LOCAL SET ADDRESSES ADDRESS P MTR/HP-IB and using the 1 and 1 keys to complete the process. The appropriate command must be selected for the model number of power meter being used. Press POWER MTR: [ ] until the model being used is displayed between the brackets. All of these steps are explained in the Compatible Peripherals chapter in the HP 8753D/Option 011 Network Analyzer User's Guide.

The correction factors for the power sensor are entered into the analyzer. All of these steps are explained in the Optimizing Measurement Results chapter of the HP 8753D/Option 011 Network Analyzer User's Guide.

The number of readings-per-point must also be selected before starting. The number of points directly affects the measurement time of the calibration sequence. The power meter must interact with the analyzer for each of the selected points and read the number of values specified for each trace point. Typically, two readings per point is considered appropriate. More than two readings-per-point could lead to unacceptable processing time.

Controlling a power meter calibration via HP-IB requires that the analyzer to be set to pass-control mode. The analyzer must position the local oscillator to a point in the sweep and read the power present at the power meter sensor. For this operation to take place, the system controller must set up the measurement and then pass control to the analyzer to read in each point in the sweep. After reading the data point from the power meter, the analyzer passes control back to the system controller. The analyzer then sets up to measure the next point and again requests control from the system controller. This process continues until the analyzer signals that the entire sweep has been measured point by point.
The following is an outline of the program's processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The number of points in the trace is set.
- The number of readings-per-point is set.
- The frequency span is set.
- The reference channel is measured.
- The power meter calibration array is allocated.
- The power meter model is chosen.
- The status registers are cleared.
- The request-control summary bit is enabled.
- The pass-control mode is enabled.
- A calibration sweep is taken to begin the sequence.
- The status byte is read until control is requested.
- The computer passes control to the analyzer.
- The display is cleared and the analyzer is set to talker/listener mode.
- The HP-IB interface status is read until control is returned.
- The program loops until all the points have been measured.
- The power meter calibration is enabled.
- The calibration data is output to the controller in FORM 4, ASCII format.
- The power meter-calibration factors are read into the controller.
- The analyzer is released from remote control and the program ends.

The program is written as follows:

```
10 ! This routine does a power meter cal using pass control.
20 ! A measurement cycle takes place on each point of the trace. The
30 ! point is measured by the power meter and the measured value read
40 ! into the analyzer. The command TAKCS; arms this measurement mode.
50 ! The number of measurements is determined by the number of points in
60 ! the trace, the number of readings per point and an extra measurement
70 ! cycle to release the pour meter.
80 ! Control is passed to the analyzer, the point is measured and
90 ! the data is transferred to the analyzer. Control is passed back to
100 ! the controller and the cycle begins again. Serial poll is used to
110 ! read the status byte of the analyzer and test the logic.
120 ! The HP-IB interface status register is monitored to determine when
130 ! control is returned to the interface from the analyzer.
140 !
150 ! EXAMP4C
160 !
170 ASSIGN @Nwa T0 716 ! Assign an I/O path for the analyzer
180 !
190 CLEAR SCREEN
200 ! Initialize the analyzer
```
210 ABORT 7
220 CLEAR @Nwa
230 OUTPUT @Nwa;"OPC?;PRES;"
240 ENTER @Nwa;Reply
250 !
260 INTEGER Stat
270 !
280 ! Set up the analyzer parameters
290 Numpoints=11
300 Numreads=2
310 Startf=1.00E+8
320 Stopf=5.0E+8
330 !
340 OUTPUT @Nwa;"POIN";Numpoints;";"
350 OUTPUT @Nwa;"NUMR";Numreads;";"
360 OUTPUT @Nwa;"STAR";Startf
370 OUTPUT @Nwa;"STOP";Stopf
380 OUTPUT @Nwa;"MEASR;"
390 !
400 ALLOCATE Pmcal(1:Numpoints)
410 !
420 ! Store the original trace for comparison
430 OUTPUT @Nwa;"DATI;"
440 OUTPUT @Nwa;"DISPDATM;"
450 OUTPUT @Nwa;"AUTO;"
460 !
470 ! Select the power meter being used for cal
480 ! OUTPUT @Nwa;"POWM ON;" ! Select 436A power meter
490 OUTPUT @Nwa;"POWMOFF;DEBUON;" ! Select 437B/438A power meter
500 !
510 ! Set analyzer HP-I/B, status regs to interrupt on pass control
520 OUTPUT @Nwa;"CLES;"
530 OUTPUT @Nwa;"ESE2;"
540 OUTPUT @Nwa;"SRE32;"
550 !
560 PRINT "Beginning Power Meter CAL"
570 OUTPUT @Nwa;"USEPASC;"
580 OUTPUT @Nwa;"TAKCS;"
590 !
600 FOR I=1 TO Numpoints*Numreads+1 ! Points * Number of readings plus 1
610 ! Serial poll does not place analyzer in remote operation
620 ! and does not require the analyzer to process the command.
630 !
640 REPEAT ! Repeat until SRQ detected
650 Stat=SPOLL(@Nwa)
660 DISP "Stat ";Stat;" Waiting for request"
670 UNTIL BIT(Stat,6) ! SRQ detected for request control
680 OUTPUT @Nwa;"ESR?;"
690 ENTER @Nwa;Reply
700 !
710 PRINT "Passing Control" ! status read and passing control
720 PASS CONTROL @Nwa
730 !
740 REPEAT
750 ! Read HP-I/B interface state information register.
760 STATUS 7,6;Hpb  ! Test HP-IB register for control
770 !
780 ! Reading the interface status register does not interact with the
790 ! analyzer. Bit 6 is set when control is returned.
800 !
810 DISP "Waiting for control"
820 UNTIL BIT(Hpb,6)  ! Loop until control is returned
830 NEXT I
840 !
850 PRINT "Finished with Power meter Cal"
860 DISP ""
870 !  ! Clear display message
880 OUTPUT @Nwa;"TALKLIST;"
890 OUTPUT @Nwa;"CLES;"
900 !  ! Restore Talker/Listener operation
910 OUTPUT @Nwa;"PWMCONES;"
920 OUTPUT @Nwa;"DPC?;WAIT;"
930 ENTER @Nwa;Reply  ! Wait for the analyzer to finish
940 !  ! Read the 1 when complete
950 ! Read the power meter cal correction factors
960 OUTPUT @Nwa;"FORM4;"
970 OUTPUT @Nwa;"OUTPPMCAL1;"
980 ENTER @Nwa;Pmcal(*)  ! ASCII data format to read cal data
990 !  ! Request the power meter cal factors
1000! Display the power meter cal factors
1010 PRINT "Point","Factor"
1020 FOR I=1 TO Numpoints  ! Cycle through the factors
1030 PRINT I,Pmcal(I)
1040 NEXT I
1050!
1060 LOCAL @Nwa
1070 END

Running the Program

The analyzer is preset and the power meter-calibration routine begins. The analyzer displays the message "WAITING FOR HP-IB CONTROL" when it is requesting control. The system controller display prints "Passing Control" when control is passed to the analyzer. The interaction of the messages and the movement of the cursor allow observation of the calibration process. Once the calibration is complete, the analyzer displays "POWER METER CAL IS COMPLETE" and the system controller displays "Finished with Power meter Cal". The power meter-calibration mode (with one sweep of correction data) is enabled and the calibration is switched ON. At the completion of the program, talker/listener mode is restored, the event-status registers are cleared (to halt the status-byte interaction), the sweep is placed in continuous-sweep mode, the analyzer is released from HP-IB control, and the program ends.
Example 5: Network Analyzer System Setups

Saving and Recalling Instrument States

Note  The most efficient option for storing and recalling analyzer states is using the analyzer's internal registers to save the CAL data. Recalling these registers is the fastest solution to restoring analyzer setups. See the Printing, Plotting, and Saving Measurement Results chapter in the HP 8753D/Option 011 Network Analyzer User's Guide for detailed information on the analyzer's internal storage registers.

In the event that all the registers have been used, or if internal memory limitations exist, then these external solutions become viable.

The purpose of this example is to demonstrate several programming options for storing and recalling entire instrument states over HP-IB. The examples describe two different processes for storing and recalling instrument states. The first example accomplishes the task using the learn string. The second example involves reading both the learn string and the calibration arrays out of the analyzer and storing them to disk or storing them in the system controller itself.

Using the learn string is a very rapid way of saving the instrument state, but using direct disk access has the advantage of automatically storing calibrations, cal kits, and data along with the instrument state.

A complete analyzer setup requires sending the learn string and a calibration array to set the analyzer parameters. The CAL array may also be placed in the analyzer, just as if a calibration was performed. By sending both sets of data, the analyzer may be quickly setup for a measurement.

Several different measurements may be required in the course of testing a device. An efficient way of performing multiple measurements is to send both the calibration array and the learn string, and then perform the measurements.

Example 5A: Using the Learn String

Note  This program is stored as EXAMP5A on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

The learn string is a very fast and easy way to read an instrument state. The learn string includes all front-panel settings, the limit table for each channel, and the list-frequency table. It can be read out of the analyzer with the command OUTPLEAS, and input to the analyzer with the command INPULEAS. This array is transmitted in FORM 1, the internal format for the analyzer. It cannot be longer than 3000 bytes. The example for a FORM 1 transfer could also have been used. However, Example 5A is the simplest solution for reading the learn string from the analyzer.

The following is an outline of the program’s processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The string storage is allocated.
- The learn string is requested.
The string is read without any processing.
- The analyzer is released from remote control.
- The instrument state is changed by the operator.
- The learn string is sent back to the analyzer.
- The analyzer is released from remote control and the program ends.

The program is written as follows:

```
10 ! This program shows how to retrieve a learn string from the analyzer
20 ! into a string array. The state of the analyzer is then changed and the
30 ! learn string re-loaded to return the analyzer to the previous settings.
40 !
50 ! EXAMP5A
60 !
70 ASSIGN @Nwa TO 716            ! Assign an I/O path for the analyzer
80 !
90 CLEAR SCREEN
100 ! Initialize the analyzer
110 ABORT 7
120 CLEAR @Nwa
130 !
140 DIM State$[3000]             ! Define a string for contents
150 !
160 OUTPUT @Nwa;"OUTPLEAS;"
170 ENTER @Nwa USING "+,-K";State$  ! Output the learn string
180        ! + Terminate on EOI only
190        ! -K ignore LF as terminator treat as data
200 !
210 LOCAL @Nwa                     ! Release HP-IB control
220 !
230 INPUT "Change state and press ENTER",A$
240 !
250 OUTPUT @Nwa;"INPULEAS;";State$;  ! Send the learnstring to analyzer
260 DISP "Analyzer state has been restored!"
270 !
280 OUTPUT @Nwa;"DPC?;WAIT;"
290 ENTER @Nwa;Reply               ! Wait for the analyzer to finish
300 LOCAL @Nwa                    ! Read the 1 when complete
310 END
```

Running the Program

Run the program. When the program stops, change the instrument state and press ENTER on
the controller. The analyzer will be returned to its original state by using the learn string.
Example 5B: Reading Calibration Data

Note
This program is stored as EXAMP5B on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

This example demonstrates:

- How to read measurement calibration data out of the analyzer.
- How to read it back into the analyzer.
- How to determine which calibration is active.

The data used to perform measurement-error correction is stored inside the analyzer in one (or more) of twelve calibration-coefficient arrays. Each array is a specific error coefficient, and is stored and transmitted as an error-corrected data array. Each point is a real/imaginary pair, and the number of points in the array is the same as the number of points in the sweep. The four data formats also apply to the transfer of calibration-coefficient arrays. The Preset State and Memory Allocation chapter in the *HP 8753D/Option 011 Network Analyzer User’s Guide* contains information on the storage locations for calibration coefficients and different calibration types.

A computer can read out the error coefficients using the commands OUTPCALC01, OUTPCALC02, ..., through OUTPCALC12. Each calibration type uses only as many arrays as required, beginning with array 1. Hence, it is necessary to know the type of calibration about to be read out: attempting to read an array not being used in the current calibration causes the "REQUESTED DATA NOT CURRENTLY AVAILABLE" warning.

A computer can also store calibration coefficients in the analyzer. To do this, declare the type of calibration data about to be stored in the analyzer just as if you were about to perform that calibration. Then, instead of calling up different classes, transfer the calibration coefficients using the INPUCALCnn commands. When all the coefficients are stored in the analyzer, activate the calibration by issuing the mnemonic SAVC, and trigger a sweep on the analyzer.

This example reads the calibration coefficients into a very large array, from which they can be examined, modified, stored, or put back into the instrument. If the data is to be directly stored on to disk, it is usually more efficient to use FORM 1 (analyzer’s internal-binary format), and to store each coefficient array as it is read in.

The following is an outline of the program’s processing sequence:

- An I/O path is assigned for the analyzer.
- A binary path is assigned.
- The system is initialized.
- The calibration types and number of arrays are defined.
- The integer variables for reading the headers are defined.
- The calibration type and number of arrays are read by the controller.
- The output is formatted in FORM 3.
- The number of points in the trace is read.
- The memory is allocated for the calibration arrays.
- Each calibration array is requested from the analyzer.
- Header information is read with a binary I/O path.
The elements from each calibration array are read in.

The next calibration array is requested until all the arrays have been read.

The calibration type is sent to the analyzer.

Each calibration array is sent.

The calibration is activated.

The analyzer is released from remote control and the program ends.

The program is written as follows:

10 ! This program shows how to manipulate calibration data from the analyzer.
20 ! It demonstrates how to read calibration data from the analyzer, and
30 ! how to replace it. The type of calibration active is determined and
40 ! the program reads in the correct number of arrays. The number of points
50 ! in the trace, and in the cal array, is determined and used to dimension
60 ! storage arrays.
70 !
80 ! EXAMPSB
90 !
100 ASSIGN @Nwa TO 716                      ! Assign an I/O path for the analyzer
110 ASSIGN @Nwa_bin TO 716;FORMAT OFF       ! Assign binary path
120 !
130 CLEAR SCREEN
140 ! Initialize the analyzer
150 ABORT 7                                 ! Generate an IFC (Interface Clear)
160 CLEAR @Nwa                              ! SDC (Selected Device Clear)
170 !
180 ! Data for determining CAL type and number of arrays
190 DATA "CALIRESP",1,"CALIRAI",2,"CALIS111",3
200 DATA "CALIS221",3,"CALIFUL2",12
210 DATA "NOOP",0

220!
230 INTEGER Hdr,Lgth,I,J                    ! Integers for reading headers
240 !
250 READ Calt$,Numb                         ! Read CAL type from data statement
260 IF Numb=0 THEN GOTO 690                 ! If no CAL type is present Exit
270 OUTPUT @Nwa;Calt$;"?;"                 ! Query if CAL type is active
280 ENTER @Nwa;Active                      ! Read 1 if active
290 IF NOT Active THEN GOTO 250            ! Load another CAL type and re-try
300 !
310 PRINT Calt$,Numb                        ! Active CAL and number of arrays
320 !
330 OUTPUT @Nwa;"FORM3;"                   ! Form 3 IEEE 64 bit floating point
340 OUTPUT @Nwa;"POIN?;"                   ! Request trace length
350 ENTER @Nwa;Poin                         ! Read number of points
360 ALLOCATE Cal(1:Numb,1:Poin,1:2)        ! Arrays for CAL arrays
370 ! Number of arrays, number of points real and imag value per point
380 !
390 FOR I=1 TO Numb                         ! Read arrays
400 OUTPUT @Nwa USING "K,ZZ";"OUTPCALC",I   ! Format I to add 0 in command
410 ENTER @Nwa_bin;Hdr,Lgth                ! Read header & length from array
420 FOR J=1 TO Poin                         ! Read elements for CAL array
430 ENTER @Nwa_bin;Cal(I,J,1),Cal(I,J,2)    ! Read real & imag pair elements
440 NEXT J                                 ! Next location in array
450 NEXT I ! Next CAL array
460 !
470 ! All CAL arrays have been read
480 !
490 INPUT "PRESS RETURN TO RE-TRANSMIT CALIBRATION",Dum$
500 !
510 OUTPUT @Nwa;"FORM3;"
520 OUTPUT @Nwa;Calt$;";
530 !
540 FOR I=1 TO Numb
550  DISP "TRANSMITTERING ARRAY: ",I
560  OUTPUT @Nwa USING "K,ZZ";"INPUCALC",I
570  OUTPUT @Nwa_bin;Hdr,Lgth
580  FOR J=1 TO Poin
590  OUTPUT @Nwa_bin;Cal(I,J,1),Cal(I,J,2)
600  NEXT J ! Next element in array
610  NEXT I ! Next array
620 !
630 OUTPUT @Nwa;"SAVC;"
640 !
650 OUTPUT @Nwa;"CONT;"
660 OUTPUT @Nwa;"OPC?,WAIT;"
670 ENTER @Nwa;Reply
680 !
690 DISP "Finished with CAL transfer"
700 LOCAL @Nwa
710 END

Running the Program

Before executing the program, perform a calibration.

The program is able to detect which type of calibration is active. With that information, it predicts how many arrays to read out. When all the arrays have been sent to the computer, the program prompts the user. The operator then turns the calibration OFF or performs a completely different calibration on the analyzer and continues the program. The computer reloads the old calibration. The operator should not preset the analyzer because the instrument settings must be the same as those that were present when the calibration was taken.

Note

The re-transmitted calibration is associated with the current instrument state: the instrument has no way of knowing the original state associated with the calibration data. For this reason, it is recommended that the learn string be used to store the instrument state whenever calibration data is stored. The next example demonstrates how to reload the analyzer state with both the learn string and the calibration arrays.
Example 5C: Saving and Restoring the Analyzer Instrument State

Note
This program is stored as EXAMP5C on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

Note
The instrument state may also be stored in the analyzer's internal registers. This is the fastest and most efficient method for toggling between instrument states. This example is for use when the analyzer's internal memory is full, or when there are other internal-memory limitations.

This example demonstrates using both the learn string and the calibration arrays to completely re-program the analyzer state. If you were performing two entirely different measurements on a device and wanted to quickly change between instrument states and perform the measurements, this example program is a potential solution.

The example will request the learn string and calibration array from the analyzer and store them in a disk file on the system controller. Once the storage is complete, the operator will be prompted to change the state of the analyzer and then re-load the state that was previously stored in the disk file. Once the file is created on the disk, the state information can be retrieved from the controller and restored on the analyzer.

Note
The disk file can only be created once. Errors will occur if the operator repeatedly tries to re-create the file.

For this example, only a through calibration will be performed and transferred. This means only one calibration array will be read from the analyzer and written to the disk file with the instrument state. To work with more elaborate calibrations, additional arrays will need to be defined and transferred to the disk file. This is not difficult, but requires some further programming steps which were omitted in the interest of presenting a simple example.

The following is an outline of the program's processing sequence:

- An I/O path is assigned for the analyzer.
- A binary path is assigned.
- The integers for reading the headers are defined.
- The system is initialized.
- An array is created to hold the learn string.
- The learn string is requested by the controller.
- The number of points in the trace is read.
- The controller allocates an array for the calibration data.
- The calibration data is read into the controller.
- The controller creates and assigns a data file for the calibration array and the learn string.
- The learn string and calibration array are stored in the disk file.
- The operator presses [ENTER] on the controller to read the calibration data back into the analyzer.
- The learn string is read from the disk file and output to the analyzer.
- The calibration array is read in from the disk file and stored in the analyzer.
The analyzer is returned to continuous-sweep mode.

The analyzer is released from remote control and the program ends.

The program is written as follows:

```
10 ! This program reads an instrument state and stores it in a disk file.
20 ! The learn string and CAL array are both read into the controller and
30 ! then transferred to a disk file for storage. The file contents are
40 ! then restored to the analyzer. The analyzer is preset to the default
50 ! settings before the instrument state is transferred back.
60 !
70 ! EXAMPSC
80 !
90 ASSIGN @Nwa TO 716 ! Assign an I/O path for the analyzer
100 ASSIGN @Nwa_bin TO 716;FORMAT OFF ! Assign a binary path
110 !
120 INTEGER Head,Length ! Integer 2 byte format for headers
130 !
140 CLEAR SCREEN
150 ! Initialize the analyzer
160 ABORT 7 ! Generate an IFC (Interface Clear)
170 CLEAR @Nwa ! SDC (Selected Device Clear)
180 !
190 ! Read in the learn string as a form 1 binary data trace
200 DIM Learn$[3000] ! Array to hold learn string
210 !
220 OUTPUT @Nwa;"OPC?;SING;" ! Place analyzer in single sweep
230 ENTER @Nwa;Reply ! Read the 1 when complete
240 !
250 OUTPUT @Nwa;"OUTPLEAS;" ! Request learn string
260 ENTER @Nwa USING "+,-,K";Learn$
270 !
280 ! Allocate an array for storing the CAL data
290 OUTPUT @Nwa;"POIN?;" ! Find number of points in trace
300 ENTER @Nwa;Num_points ! Read number to allocate array
310 ALLOCATE Cal_array(1:Num_points,1:2) ! Real and Imag for each point
320 !
330 ! Read Cal array
340 OUTPUT @Nwa;"FORM3;" ! Form 3 64 bit floating point data
350 OUTPUT @Nwa;"OUTPCALC01;" ! Request the cal array
360 !
370 ! Read the #A and 2 byte length as integers
380 ENTER @Nwa_bin;Head,Length,Cal_array(*) ! Read cal array data
390 !
400 ! Write instrument state data to disk file
410 ! CREATE BDAT "DATA_FILE:;1406",1,Length+3000 ! Create data file once!
420 ASSIGN @File TO "DATA_FILE:;1406" ! Assign I/O path to file
430 OUTPUT @File;Learn$ ! Send learn string to disk file
440 OUTPUT @File;Head,Length,Cal_array(*) ! Send CAL arrays to disk file
450 ASSIGN @File TO * ! Close file
460 !
470 INPUT "Cal data received. Press ENTER to send it back.",A$
480 !
490 ! Read arrays from file
500 !
```

3-46  HP BASIC Programming Examples
510 DIM Learn2$(3000)
520 ASSIGN @File TO "DATA_FILE:;1406"
530 ENTER @File;Learn2$
540 !
550 ENTER @File;Head,Length
560 Size=Length/16
570 ALLOCATE Cal_array2(1:Size,1:2)
580 ENTER @File;Cal_array2(*)
590 !
600 ! Send Learn string back
610 OUTPUT @Nwa:"INPULEAS:".Learn2$
620 !
630 ! Send Cal array back
640 OUTPUT @Nwa:"CALIRESP;"
650 OUTPUT @Nwa:"INPUCALCO1;"
660 OUTPUT @Nwa_bin;Head,Length,Cal_array2(*)
670 OUTPUT @Nwa:"OPC?;SAVC;"
680 ENTER @Nwa;Reply
690 !
700 OUTPUT @Nwa;";CONT;"
710 OUTPUT @Nwa;"OPC?;WAIT;"
720 ENTER @Nwa;Reply
730 LOCAL @Nwa
740 END

Running the Program

Setup the analyzer and perform a thorough calibration.

Run the program. The program prompts the operator to change the state of the analyzer and then press [ENTER] to continue. At this point, the analyzer state is stored on the disk file in the controller. Pressing [ENTER] will begin the transfer from the disk file to internal arrays within the controller and then on to the analyzer.

Once completed:

- The original state will be restored.
- The analyzer will be sweeping.
- The analyzer will be calibrated.
- CCR will be displayed on the analyzer’s CRT.
Example 6: Limit-Line Testing

Using List-Frequency Mode

The analyzer normally takes data points spaced at regular intervals across the overall frequency range of the measurement. For example, for a 2 GHz frequency span using 201 points, data will be taken at intervals of 10 MHz. The list-frequency mode allows the operator to select the specific points, or frequency spacing between points, at which measurements are to be made. This mode of operation allows flexibility in setting up tests to insure device performance in an efficient manner. By only sampling specific points, measurement time is reduced, since additional time is not spent measuring device performance at frequencies which are of no concern. List-frequency sweeps are also discussed in the Application and Operation Concepts chapter of the HP 8753D/Option 011 Network Analyzer User’s Guide. These programs emulate operation from the analyzer’s front panel when using list sweeps.

The following two examples illustrate the use of the analyzer’s list-frequency mode to perform arbitrary frequency testing. Example 6A lets the operator construct a table of list-frequency segments which is then loaded into the analyzer’s list-frequency table. There are a maximum of 30 segments available. Each segment stipulates a start and stop frequency, and the number of data points to be taken over that frequency range. Example 6B lets the operator select a specific segment to “zoom-in” on. A single instrument can programmed to measure several different devices, each with its own frequency range, using a single calibration performed with all of the segments active. When a specific device is connected, the operator selects the appropriate segment for that device. Note that list-frequency segments can be overlapped, but the total number of points in all the segments must not exceed 1632.

Example 6A: Setting Up a List-Frequency Sweep

Note

This program is stored as EXAMP6A on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

The purpose of this example is to show how to create a list-frequency table and transmit it to the analyzer.

The command sequence for entering a list-frequency table imitates the key sequence followed when entering a table from the front panel: there is a command for every key-press.

Editing a segment is also the same as the front-panel key sequence, but remember the analyzer automatically reorders each edited segment in order of increasing start frequency.

The list-frequency table is also carried as part of the learn string. While the table cannot be modified as part of the learn string, it can be stored and recalled with very little effort by storing and recalling the learn string. See Chapter 2 under the section titled “Data Processing Chain” for details on using learn strings.

This example takes advantage of the computer’s capabilities to simplify:

- creating a list-frequency table
- editing a list-frequency table

The table is entered and completely edited before being transmitted to the analyzer. To simplify the programming task, options such as entering center frequency, frequency span, or step size are not included.

The list-frequency information may be acquired using the limit-test results array. The actual stimulus points are available as the first element in the array.
The following is an outline of the program’s processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The existing list frequencies are edited and cleared.
- The number of segments to define is read in.
- An array for the list segments is defined.
- The parameters for each segment are requested.
- If the operator wants to edit, the segment parameters are re-entered.
- The new list is sent to the analyzer.
- The analyzer is released from remote control and the program ends.

The program is written as follows:

```
10 ! This program shows how to enter and edit a list frequency table.
20 ! Any existing table is deleted and a new table is defined and
30 ! edited. This list is then sent to the analyzer. Any number of
40 ! segments or points may be entered. Be sure not to enter more than
50 ! 1632 points or 30 segments.
60 !
70 ! EXAMPE6A
80 !
90 ASSIGN @Nwa TO 716
  ! Assign an I/O path for the analyzer
100 !
110 CLEAR SCREEN
  ! Generate an IFC (Interface Clear)
120 ! Initialize the analyzer
  ! SDC (Selective Device Clear)
130 ABORT 7
140 CLEAR @Nwa
150 OUTPUT @Nwa;"OPC?;PRES;"
  ! Preset the analyzer and wait
160 ENTER @Nwa;Reply
  ! Read the 1 when complete
170 !
180 OUTPUT @Nwa;"EDITLIST;"
  ! Begin editing the frequency list
190 OUTPUT @Nwa;"CLEL;"
  ! Clear the existing list frequencies
200 !
210 INPUT "Number of segments?",Numb
  ! Read number of segments to define
220 ALLOCATE Table(1:Numb,1:3)
  ! Define an array for the list segments
230 !
240 PRINT USING "10A,15A,15A,20A";"SEGMENT","START(MHZ)","STOP(MHZ)","NUMBER OF
  ! POINTS"
250 !
260 FOR I=1 TO Numb
  ! Cycle through the segments and read in the values
270  GGSUB Loadpoint
280 NEXT I
290 !
300 LOOP
310 INPUT "DO YOU WANT TO EDIT? Y OR N",An$"N"
320 EXIT IF An$="N"
330 INPUT "ENTRY NUMBER?",I
  ! Get an entry number
340  GGSUB Loadpoint
350 END LOOP
360 !
370 OUTPUT @Nwa;"EDITLIST"
  ! Send the new list to the analyzer
```

HP BASIC Programming Examples 3-49
380 FOR I=1 TO Numb
390 OUTPUT @Nwa;"SADD;"
400 OUTPUT @Nwa;"STAR";Table(I,1);"MHZ;"
410 OUTPUT @Nwa;"STOP";Table(I,2);"MHZ;"
420 OUTPUT @Nwa;"POIN";Table(I,3);";"
430 OUTPUT @Nwa;"SDON;"
440 NEXT I
450 !
460 OUTPUT @Nwa;"EDITDONE;"
470 OUTPUT @Nwa;"LISFREQ;"
480 !
490 OUTPUT @Nwa;"OPC?;WAIT;"
500 ENTER @Nwa;Reply
510 LOCAL @Nwa
520 STOP
530 !
540 ! ***************Subroutines ***************
550 !
560 Loadpoin:
570 INPUT "START FREQUENCY? (MHZ)",Table(I,1) ! Read start frequency
580 INPUT "STOP FREQUENCY? (MHZ)",Table(I,2) ! Read stop frequency
590 INPUT "NUMBER OF POINTS?",Table(I,3) ! Read number of points in seg
600 IF Table(I,3)=1 THEN Table(I,2)=Table(I,1) ! Single point same start stop
610 !
620 ! Print new segment into table on display
630 PRINT TABXY(0,I+1);I;TAB(10);Table(I,1);TAB(25);
640 PRINT Table(I,2);TAB(40),Table(I,3)
650 RETURN
660 END

Running the Program

Caution

This example program will delete any existing limit lines before entering the new limits. If this is not desired, omit the line(s) that clear the existing limits (in this case LINE 190). This program begins by presetting the analyzer. The programmer will have to add the necessary command lines to set the analyzer to the specific operating conditions required for testing. The example program will show the limit lines defined, but the limits will always fail without additional analyzer setup.

The program displays the frequency-list table as it is entered. During editing, the displayed table is updated as each line is edited. The table is not re-ordered. At the completion of editing, the table is entered into the analyzer, and list-frequency mode is switched ON. During editing, simply pressing [RETURN] leaves an entry at the old value.

The table will be sorted by the analyzer and displayed.
Example 6B: Selecting a Single Segment from a Table of Segments

Note  This program is stored as EXAMP6B on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

The program demonstrated how to define a single segment as the operating-frequency range of the analyzer from a table of segments stored in the controller. The program assumes that a list-frequency table has already been entered into the analyzer, either manually, or using the program in Example 6A, “Setting Up a List-Frequency Sweep.”

The program first loads the list-frequency table into the computer by reading the start and stop frequencies of each segment and the number of points for each segment. The segment’s parameters are then displayed on the computer screen, and the operator can choose which segment is to be used by the analyzer. Note that only one segment can be chosen at a time.

The following is an outline of the program’s processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The list-frequency segment is edited.
- The largest segment number is set.
- The highest segment number is requested.
- The number of actual segments is read in.
- A list-frequency table is defined and the segments are read in to the controller from the analyzer.
- The operator selects one of the segments of the sweep.
- The controller “zooms-in” and sweeps the defined segment.
- The operator presses [O] and the analyzer returns to sweeping all the segments in the table.
- The activation loop is ended and the program ends.

The program is written as follows:

```
10 ! This program shows how to select a single segment from a list
20 ! frequency sweep and activate it as the sweep. The list frequency
30 ! table is read from the analyzer and displayed on the computer
40 ! screen. The operator is prompted to select a segment and the
50 ! program then activates it. All the segments are activated upon
60 ! completion.
70 !
80 ! EXAMP6B
90 !
100 ASSIGN @Nwa T0 716 ! Assign an I/O path for the analyzer
110 !
120 CLEAR SCREEN
130 ! Initialize the analyzer ! Generate an IFC (Interface Clear)
140 ABORT 7
150 CLEAR @Nwa
160 !
170 ! Print header for table of existing segments
180 PRINT USING "10A,15A,15A,20A","SEGMENT","START(MHZ)","STOP(MHZ)","NUMBER OF
```
POINTER"  
190 OUTPUT @Nwa;"EDITLIST;"  ! Edit list frequency segment  
200 OUTPUT @Nwa;"SEDI30;"  ! Set largest segment number  
210 OUTPUT @Nwa;"SEDI?;"  ! Request number of highest segment  
220 ENTER @Nwa;Numsegs  ! Read number of actual segments  
230 !  
240 ! Setup table and read segments from analyzer  
250 ALLOCATE Table(1:Numsegs,1:3)  ! Allocate table of segments  
260 FOR I=1 TO Numsegs  ! Cycle through segments  
270 GOSUB Readlist  ! Read in segment definitions  
280 NEXT I  ! Next segment  
290 !  
300 ! Loop and read segment to be activated  
310 LOOP  ! Request operator to enter segment  
320 INPUT "SELECT SEGMENT NUMBER: (0 TO EXIT)",Segment  
330 EXIT IF Segment=0  ! Exit point  
340 OUTPUT @Nwa;"EDITDONE;";"SSEG";Segment;";"  ! Set active segment to sweep  
350 END LOOP  ! End activation loop  
360 !  
370 OUTPUT @Nwa;"ASEG;"  ! Set all segment sweep  
380 DISP "PROGRAM ENDED"  
390 !  
400 OUTPUT @Nwa;"OPC?;WAIT;"  ! Wait for analyzer to finish  
410 ENTER @Nwa;Reply  ! Read the 1 when complete  
420 LOCAL @Nwa  ! Release HP-IB control  
430 STOP  ! End of main program  
440 !  
450 !***************************************************************************** Subroutines*****************************************************************************  
460 !  
470 Readlist:  ! Read segment list from analyzer  
480 OUTPUT @Nwa;"EDITLIST;"  ! Edit segment list  
490 OUTPUT @Nwa;"SEDI","I",";"  ! Select segment to edit  
500 OUTPUT @Nwa;"STAR;"  ! Send start freq to display value  
510 OUTPUT @Nwa;"OUTFACTI;"  ! Output active function value  
520 ENTER @Nwa;Table(I,1)  ! Read start frequency  
530 OUTPUT @Nwa;"STOP;"  ! Send stop freq to display value  
540 OUTPUT @Nwa;"OUTFACTI;"  ! Output active function value  
550 ENTER @Nwa;Table(I,2)  ! Read stop frequency  
560 OUTPUT @Nwa;"POIN;"  ! Send number of points to display  
570 OUTPUT @Nwa;"OUTFACTI;"  ! Output active function value  
580 ENTER @Nwa;Table(I,3)  ! Read number of points  
590 !  
600 IF I=18 THEN  ! Pause if more than 17 segments  
610 INPUT "PRESS RETURN FOR MORE",A$  ! Read Return to continue  
620 END IF  
630 ! Print new header for segment data  
640 IMAGE 4D,6X,4D,6D,3X,4D,6D,3X,4D  ! Format image to disp segment data  
650 PRINT USING 640;I;Table(I,1)/1.E+6;Table(I,2)/1.E+6;Table(I,3)  
660 RETURN  
670 !  
680 END
Running the Program

The program will read the parameters for each list-frequency segment from the analyzer, and build a table containing all the segments. The parameters of each segment will be printed on the computer screen. If there are more than 17 segments, the program will pause. Press RETURN to see more segments. The maximum number of segments that can be read is 30 (the maximum number of segments the analyzer can hold). Use the computer’s Page Up and Page Down keys to scroll through the list of segments if there are more than 17.

After all the segments are displayed, the program will prompt the operator for a specific segment to be used. Type in the number of the segment, and the analyzer will then “zoom-in” on that segment. The program will continue looping, allowing continuous selection of different segments. To exit the loop, type 0. This will restore all the segments (with the command ASEG), allowing the analyzer to sweep all of the segments, and the program will terminate.

Using Limit Lines to Perform PASS/FAIL Tests

There are two steps to performing limit testing on the analyzer via HP-IB. First, limit specifications must be defined and loaded into the analyzer. Second, the limits are activated, the device is measured, and its performance to the specified limits is signaled by a pass or fail message on the analyzer’s display.

Example 6C illustrates the first step, setting up limits. Example 6D performs the actual limit testing.

Example 6C: Setting Up Limit Lines

Note
This program is stored as EXAMP6C on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

The purpose of this example is to show how to create a limit-test table and transmit it to the analyzer.

The command sequence for entering a limit-test table imitates the key sequence followed when entering a table from the analyzer’s front panel: there is a command for every key-press. Editing a limit is also the same as the key sequence, but remember that the analyzer automatically re-orders the table in order of increasing start frequency.

The limit-test table is also carried as part of the learn string. While it cannot be modified as part of the learn string, the learn string can be stored and recalled with very little effort. See the section titled “Data Processing Chain” in Chapter 2 for details on using learn strings.

This example takes advantage of the computer’s capabilities to simplify creating and editing the table. The table is entered and completely edited before being transmitted to the analyzer. To simplify the programming task, options such as entering offsets are not included.

This example automates the front-panel operation of entering a limit-test table. Front-panel operation and limits are discussed in the Application and Operation Concepts chapter of the HP 8753D/Option 011 Network Analyzer User’s Guide. The following is an outline of the program’s processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The limit lines are edited and cleared.
- The number of limits is requested.
The limit table is created.
- The string array of limit types is created.
- The operator is prompted to enter the new limit values.
- The new limit table is sent back to the analyzer.
- The limit line is activated.
- The limit test is activated.
- The analyzer is returned to local control and the program ends.

The program is written as follows:

```
10 ! This program shows how to create a limit table and send it to the
20 ! analyzer. The operator enters the desired limits when prompted for
30 ! the stimulus value, upper and lower value and type of limit
40 ! desired. Once the table is created, the limits are sent to the
50 ! analyzer and activated.
60 !
70 ! EXAMP6C
80 !
90 ASSIGN @Nwa TO 716 ! Assign an I/O path for the analyzer
100 !
110 CLEAR SCREEN
120 ! Initialize the analyzer
130 ABORT 7
140 CLEAR @Nwa
150 OUTPUT @Nwa;"DPC?;PRES;"
160 ENTER @Nwa;Reply
170 !
180 OUTPUT @Nwa;"EDITLIML;"
190 OUTPUT @Nwa;"CLEL;"
200 INPUT "NUMBER OF LIMITS?",Numb
210 ALLOCATE Table(1:Numb,1:3)
220 ALLOCATE Limtype$(Numb)[2]
230 !
240 ! Print out the header for the table
250 PRINT USING "10A,20A,15A,20A","SEG","STIMULUS (MHz)","UPPER (dB)","
260 ! LOWER (dB)", "TYPE"
260 !
270 ! Prompt the operator to enter the limit values
280 FOR I=1 TO Numb
290 GOSUB Loadlimit
300 NEXT I
310 !
320 ! Allow the operator to edit the array entered
330 LOOP
340 INPUT "DO YOU WANT TO EDIT? Y OR N",An$
350 EXIT IF An$="N"
360 INPUT "ENTRY NUMBER?",I
370 GOSUB Loadlimit
380 END LOOP
390 !
400 ! Send the limit line array segments to the analyzer
410 OUTPUT @Nwa;"EDITLIML;" ! Edit the limit
```
FOR I=1 TO Numb
  OUTPUT @Nwa;"SADD;" ! Each segment of the limit
  OUTPUT @Nwa;"LIMS;"Table(I,1);"MHZ" ! Add segment
  OUTPUT @Nwa;"LIMU;"Table(I,2);"DE" ! Send segment stimulus value
  OUTPUT @Nwa;"LIML;"Table(I,3);"DB" ! Upper limit value
  OUTPUT @Nwa;"LIMT;"Table(I,3);"DB" ! Lower limit value
IF Limtype$(I)="FL" THEN OUTPUT @Nwa;"LIMTFL;" ! Flat limit
IF Limtype$(I)="SL" THEN OUTPUT @Nwa;"LIMTSL;" ! Sloping limit
IF Limtype$(I)="SP" THEN OUTPUT @Nwa;"LIMTSP;" ! Point limit
OUTPUT @Nwa;"SDON;" ! Segment done
NEXT I ! next segment
OUTPUT @Nwa;"EDITDONE;" ! Edit complete
OUTPUT @Nwa;"LIMLINEON;" ! Turn limit line on
OUTPUT @Nwa;"LIMTESTON;" ! Turn limit test on
OUTPUT @Nwa;"OPC;WAIT;" ! Wait for the analyzer to finish
ENTER @Nwa;Reply ! Read the 1 when complete
LOCAL @Nwa ! Release HP-IB control
STOP ! End of main program

Subroutines

Loadlimit: ! Sub to interact to load data
  INPUT "STIMULUS VALUE? (MHz)",Table(I,1) ! and print table created
  INPUT "UPPER LIMIT VALUE? (DB)",Table(I,2)
  INPUT "LOWER LIMIT VALUE? (DB)",Table(I,3)
  INPUT "LIMIT TYPE? (FL=FLAT, SL=SLOPED, SP=SINGLE POINT)",Limtype$(I)
  PRINT TABXY(0,I+1);1;TAB(10);Table(I,1);TAB(30);Table(I,2);TAB(45),
  Table(I,3),TAB(67);Limtype$(I)
RETURN ! Next limit value
RETURN
END

Running the Program

Caution
This example program will delete any existing limit lines before entering the new limits. If this is not desired, omit the line(s) that clear the existing limits (in this case LINE 190). This program begins by presetting the analyzer. The programmer will have to add the necessary command lines to set the analyzer to the operating conditions required for testing. The example program will show the limit lines defined, but the limits will always fail without additional analyzer setup.

The program displays the limit table as it is entered. During editing, the displayed table is updated as each line is edited. The table is not reordered. At the completion of editing, the table is entered into the analyzer, and limit-testing mode switched ON. The analyzer will rearrange the table in ascending order starting with the lowest start frequency entry. During editing, simply pressing [RETURN] leaves an entry at the old value.
Example 6D: Performing PASS/FAIL Tests While Tuning

**Note**
This program is stored as EXAMP6D on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

The purpose of this example is to demonstrate the use of the limit/search-fail bits in event-status register B, to determine whether a device passes the specified limits. Limits can be entered manually, or using the Example 5A.

The limit/search-fail bits are set and latched when limit testing or a marker search fails. There are four bits, one for each channel for both limit testing and marker search. See Figure 2-5 “Status Reporting Structure” and Table 2-4 “Units as a Function of Display Format” for additional information. Their purpose is to allow the computer to determine whether the test/search executed was successful. They are used in the following sequence:

1. Clear event-status register B.
2. Trigger the limit test or marker search.
3. Check the appropriate fail bit.

When using limit testing, the best way to trigger the limit test is to trigger a single sweep. By the time the single sweep command (SING) finishes, limit testing will have occurred.

**Note**
If the device is tuned during the sweep, it may be tuned into and then out of limit, causing a limit test to qualify as “passed” when the device is not in fact within the specified limits.

When using marker searches (max, min, target, and widths), outputting marker or bandwidth values automatically triggers any related searches. Therefore, all that is required is to check the fail bit after reading the data.

In this example, several consecutive sweeps must qualify as “passing” in order to insure that the limit-test pass was not extraneous due to the device settling or operator tuning during the sweep. Upon running the program, the number of “passed” sweeps for qualification is entered. For very slow sweeps, a small number of sweeps such as two are appropriate. For relatively fast sweeps, where the device requires time to settle after tuning, as many sweeps as six or more sweeps may be more appropriate.

A limit-test table can be entered over HP-IB. The sequence is very similar to that used in entering a list-frequency table, as shown in Example 5D. The manual (front-panel entry) sequence is closely followed.

The following is an outline of the program’s processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The pass counter is initialized on entry.
- The analyzer takes a sweep.
- The event-status register B byte is output and the channel-1 limit is tested.
- If the device fails the first sweep, the operator is prompted to insure it is tuned correctly and the device is measured again.
- If the device passes the first sweep, the operator is prompted not to touch the device as testing continues.

3-56  HP BASIC Programming Examples
If the device passes the required number of sweeps, the operator is prompted that the device has passed and to connect the next device for testing.

The program initializes the pass counter and begins to measure the new device.

The program is written as follows:

```
10 ! This program demonstrates Pass/Fail tests using limit lines. The
20 ! program uses the latch-on-fail limit bits in event status register
30 ! B to determine if the device performance passes the specified test
40 ! limit lines. It then requires that the device passes for multiple
50 ! consecutive sweeps in order to ensure that the device is static in
60 ! the response and not varying. The operator specifies how many sweeps
70 ! are required to pass the test.
80 !
90 ! EXAMPLE
100 !
110 ASSIGN @Nwa TO 716 ! Assign an I/O path for the analyzer
120 !
130 CLEAR SCREEN
140 ! Initialize the analyzer No preset to retain settings for testing
150 ABORT 7 ! Generate an IFC (Interface Clear)
160 CLEAR @Nwa ! SDC (Selected Device Clear)
170 !
180 INPUT "Number of consecutive passed sweeps for qualification?",Qual
190 Pass=0 ! Initialize pass counter on entry
200 !
210 Tune: DISP "TUNE DEVICE AS NECESSARY" ! Device is not passing warning
220 !
230 Measure:OUTPUT @Nwa;"OPC?;SING;" ! Single sweep and wait
240 ENTER @Nwa;Reply ! Read the 1 when completed
250 !
260 OUTPUT @Nwa;"ESB?;" ! Event status register B byte
270 ENTER @Nwa;Estat ! Reading byte clears the register
280 !
290 IF BIT(Estat,4) THEN ! Bit 4 is failed limit on channel 1
300 IF Pass>0 THEN BEEP 1200,.05 ! passed before? Now not passing beep
310 Pass=0 ! Reset pass to 0
320 GOTO Tune ! Adjust and measure again
330 END IF
340 !
350 BEEP 2500,.01 ! Limit test passed passing beep
360 Pass=Pass+1 ! Increment number of passes
370 DISP "LEAVE DEVICE ALONE" ! Warn not to adjust as it passed
380 !
390 IF Pass<Qual THEN GOTO Measure ! If not enough passes to qualify
400 !
410 ! Device passed
420 DISP "DEVICE PASSED!" ! Number of passes enough to qualify
430 FOR I=1 TO 10 ! Announce the device passed and
440 BEEP 1000,.05 ! prompt operator to connect new
450 BEEP 2000,.01 ! device to test.
```
460 NEXT I
470 !
480 INPUT "PRESS RETURN FOR NEXT DEVICE",Dum$
490 Pass=0               ! Initialize pass counter
500 GOTO Measure        ! Begin measurement
510 !
520 END

Running the Program

| Note | This program assumes a response calibration (through calibration) or a full |
|      | 2-port calibration has been performed prior to running the program.      |

Set up a limit-test table on channel 1 for a specific device either manually, or using the program in Example 5A.

Run the program, and enter the number of passed sweeps desired for qualification. After entering the qualification number, connect the device. When a sweep passes, the computer beeps. When enough consecutive sweeps qualify the device as "passing," the computer emits a dual-tone beep to attract the attention of the operator, and then prompts for a new device.

To test the program's pass/fail accuracy, try causing the DUT to fail by loosening the cables connecting the DUT to the analyzer and running the program again.
Example 7: Report Generation

The analyzer has three operating modes with respect to HP-IB. These modes can be changed by accessing softkeys in the [LOCAL] menu. System-controller mode is used when no computer is present. This mode allows the analyzer to control the system. The other two modes allow a remote system controller to coordinate certain actions: in talker/listener mode the remote system controller can control the analyzer, as well as coordinate plotting and printing; and in pass-control mode the remote system controller can pass active control to the analyzer so that the analyzer can plot, print, control a power meter, or load/store to disk. Peripheral control is the main difference between talker/listener and pass-control mode. See Chapter 2 for more details.

Example 7A1: Operation Using Talker/Listener Mode

| Note | This program is stored as EXAMP7A1 on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer. |

The commands OUTFPLOT and OUTPRIN allow talker/listener mode plotting and printing via a one way data path from the analyzer to the plotter or printer. The computer sets up the path by addressing the analyzer to talk, the plotter to listen, and then releasing control of the analyzer in order to transfer the data. The analyzer will then make the plot or print. When it is finished, it asserts the End or Identify (EOI) control line on HP-IB. The controller detects the presence of EOI and re-asserts control of the HP-IB. This example program makes a plot using the talker/listener mode.

| Note | One of the attributes of the OUTPLOT; command is that the plot can include the current softkey menu. The plotting of the softkeys is enabled with the command PSOFTON; and disabled with PSOFTOFF. |

The following is an outline of the program’s processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The selected frequency span is swept once.
- The plot command is sent to the analyzer.
- The analyzer is set to talker mode and the plotter is set to listener mode.
- The plot is spooled to the plotter.
- The analyzer is set to listener mode when the controller detects an EOI from the analyzer.
- The controller puts the analyzer back in continuous-sweep mode.
- The analyzer is returned to local control and the program ends.
The program is written as follows:

10 ! This example shows a plot operation under the control of the
20 ! analyzer. The analyzer is commanded to output plot data, the
30 ! plotter is addressed to listen, and the analyzer to talk. The
40 ! controller watches for EOI at the end of the plot sequence and
50 ! then regains control of the HP-IB operations.
60 !
70 ! EXAMP7A1
80 !
90 ASSIGN @Nwa TO 716
100 ! Assign an I/O path for the analyzer
110 CLEAR SCREEN
120 ! Initialize analyzer without preset to preserve data
130 ABORT 7
140 CLEAR @Nwa
150 !
160 OUTPUT @Nwa;"OPC?;SING;"
170 ENTER @Nwa;Reply
180 ! Generate an IFC (Interface Clear)
190 OUTPUT @Nwa;"OUTPPLOT;"
200 SEND 7;UNL LISTEN 5 TALK 16 DATA
210 DISP "Plotting and waiting for EOI"
220 WAIT .5
230 ! Generate an SDC (Selected Device Clear)
240 REPEAT
250 STATUS 7,7;Stat
260 UNTIL BIT(Stat,11)
270 !
280 End_plot:DISP "End of plot"
290 !
300 OUTPUT @Nwa;"CONT;"
310 OUTPUT @Nwa;"OPC?;WAIT;"
320 ENTER @Nwa;Reply
330 LOCAL @Nwa
340 END

Running the Program

The analyzer will go into remote, and make the plot. During the plot, the computer will display the message Plotting and waiting for EOI. When the plot is completed, the analyzer asserts the EOI line on the HP-IB. The computer detects this and displays the End of plot message.

If a problem arises with the plotter, such as no pen or paper, the analyzer cannot detect the situation because it only has a one-way path of communication. Hence, the analyzer will attempt to continue plotting until the operator intervenes and aborts the plot by pressing the analyzer's Local key.

Pressing Local will do the following:

- Aborts the plot.
- Causes the warning message CAUTION: PLOT ABORTED
- Asserts EOI to return control of the bus to the system controller.

Because of possible peripheral malfunctions, it is generally advisable to use pass-control mode, which allows two way communication between the peripherals and the analyzer.

3-60 HP BASIC Programming Examples
Example 7A2: Controlling Peripherals Using Pass-Control Mode

Note
This program is stored as EXAMP7A2 on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

If the analyzer is in pass-control mode and it receives a command telling it to plot, print, control a power meter, or store/load to disk, it sets bit 1 in the event-status register to indicate that it requires control of the bus. If the computer then uses the HP-IB pass-control command to pass control to the analyzer, the analyzer will take control of the bus and access the peripheral. When the analyzer no longer requires control, it will pass control back to the computer. For a discussion on the pass-control mode, see Chapter 2.

In this example, the pass-control mode is used to allow the network analyzer to dump a screen display to a printer.

Pass-control mode allows the analyzer to control the printer while sending the screen display to be printed. Once the printer-dump operation is complete, the analyzer passes control back to the controller and the controller continues programming the analyzer. The analyzer requests control from the instrument controller and the controller allows the analyzer to take control of the HP-IB and dump the plot. The instrument controller must not interact with the HP-IB while this remote analyzer control is taking place.

Note
The analyzer assumes that the address of the computer is correctly stored in its HP-IB addresses menu under LOCAL ADDRESS: CONTROLLER. If this address is incorrect, control will not return to the computer. Similarly, if control is passed to the analyzer while it is in talker/listener mode, control will not return to the computer.

Control should not be passed to the analyzer before it has set event-status-register bit 1 making it Request Active Control. If the analyzer receives control before the bit is set, control is passed immediately back to the controller.

When the analyzer becomes the active system controller, it is free to address devices to talk and listen as required. The only functions denied the analyzer are the ability to assert the interface clear line (IFC), and the remote line (REN). These are reserved for the master system controller. As the active system controller, the analyzer can send and receive messages from printers, plotters, and disk drives.

The following is an outline of the program’s processing sequence:

- An I/O path is assigned for the analyzer.
- The system is initialized.
- The status registers are cleared.
- Bit 1 of ESR request control is set.
- The ESR interrupt for SRQ is enabled.
- The pass-control mode is enabled.
- The data is dumped to the printer.
- The program loops until the SRQ bit is set.
- The status byte is read with a serial poll.
- The program tests for bit 6, SRQ.
- If SRQ is detected, the program tests for pass control (bit 5 of the status byte).
- If the analyzer requests control, the system controller gives the analyzer control of the bus.
- The program loops and waits for the analyzer to complete the print dump.
- The analyzer reads the interface.
- If bit 6 is active in the controller, control is returned from the analyzer to the controller.
- The status-byte assignments are cleared.
- The analyzer returns to continuous-sweep mode.
- The analyzer is returned to local control and the program ends.

The program is written as follows:

```
10 ! This example shows a pass-control operation to print the display
20 ! under the analyzer HP-IB control. The controller reads the status
30 ! of the analyzer looking for SRQ to indicate that the analyzer is
40 ! requesting control. Once control is passed to the analyzer, the
50 ! controller monitors the status of its interface registers to detect
60 ! when the interface is again the active controller. The analyzer will
70 ! pass control back to the controller when finished.
80 !
90 ! EXAMP7A2
100 !
110 ASSIGN @Nwa TO 716
120 ! Assign an I/O path for the analyzer
130 CLEAR SCREEN
140 ! Initialize the analyzer without preset to preserve data
150 ABORT 7
160 CLEAR @Nwa
170 !
180 OUTPUT @Nwa:"OPC?;SING;"
190 ENTER @Nwa;Reply
200 ! Single sweep and stop for print
210 OUTPUT @Nwa:"CLES;"
220 OUTPUT @Nwa:"ESE2;"
230 OUTPUT @Nwa:"SRE32;"
240 ! Clear status registers
250 OUTPUT @Nwa:"USEPASC;"
260 OUTPUT @Nwa:"PRINALL;"
270 ! Enable pass control mode
280 REPEAT
290 Stat=SPOLL(@Nwa)
300 UNTIL BIT(Stat,6)
310 ! Loop until SRQ bit is set
320 Pass_control:
330 IF BIT(Stat,5) THEN
340 PASS CONTROL @Nwa
350 ELSE
360 DISP "SRQ but not request pass control"
370 STOP ! Not bit 5, some other event
380 END IF
390 !
400 DISP "Printing from analyzer and waiting for control"
410 !
```

3-62 HP BASIC Programming Examples
420 REPEAT  
430 STATUS 7,6;Hpib  
440 UNTIL BIT(Hpib,6)  
450 ! Loop and wait for completion  
460 DISP "Control returned from analyzer"  
470 OUTPUT @Nwa;"TALKLIST;"  
480 OUTPUT @Nwa;"CLES;"  
490 ! Set talker/listener mode again  
500 OUTPUT @Nwa;"CONT;"  
510 OUTPUT @Nwa;"OPC?;WAIT;"  
520 ENTER @Nwa;Reply  
530 ! Clear status byte assignments  
540 LOCAL @Nwa  
550 END  

Running the Program

The analyzer will briefly flash the message WAITING FOR CONTROL, before actually receiving control and generating the printer output. The computer will display the PRINTING FROM analyzer and WAITING FOR CONTROL message.

When the printer output is complete, the analyzer passes control back to the address stored as the controller address under the LOCAL SET ADDRESSES menu. The computer will detect the return of active control and exit the wait loop. The controller will display the message CONTROL returned from analyzer and then release the analyzer from remote control.

Because the program waits for the analyzer's request for control, it can be used to responding to front-panel requests as well. Remove the PRINALL; command from the program and run the program again. Nothing will happen until a print, plot, or disk access is requested from the analyzer's front panel. For example, press LOCAL COPY and PRINT.

Example 7A3: Printing with the Serial Port

Note  This program is stored as EXAMP7A3 on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

This program will select the serial port and program the analyzer to copy its display to a printer. There are a number of commands associated with the serial and parallel ports which allow the programmer to configure the output modes, for example the baud rate and the handshake type used by the port and the printer. In this example, the serial port is configured by the program. The interface may also be configured from the analyzer's front-panel keys by pressing LOCAL SET ADDRESSES: PRINTER-PORT. This menu allows manual selection of the serial-interface parameters.

Since the HP-IB port is not being used for the copy operation, programming of the analyzer and measurement operations may continue once the copy operation has been initiated. An internal spooler in the analyzer's memory provides buffering of the printer operation. In the example which follows, the status byte of the analyzer is checked to determine when the print operation is complete.

- An I/O path is assigned to the analyzer.
- The analyzer is initialized.
- A single sweep is taken and the analyzer is placed in hold mode.
- The status registers are cleared.
The copy-complete bit is set and enabled.
The printer operation and communication modes are set.
The print command is sent.
The analyzer is released from remote control and placed in continuous-sweep mode.
The analyzer is polled until the status bit representing copy complete is detected.
The analyzer is released from remote control and the program ends.

The program is written as follows:

10 ! This program shows how to set up and print the display through the serial printer port.
20 ! serial printer port.
30 !
40 ! EXAMP7A3
50 !
60 ASSIGN @Nwa TO 716 ! Assign an I/O path for the analyzer
70 !
80 CLEAR SCREEN
90 ! Initialize the analyzer without preset to preserve the data
100 ABORT 7 ! Generate an IFC (Interface Clear)
110 CLEAR @Nwa ! SDC (Selected Device Clear)
120 !
130 OUTPUT @Nwa,"OPC?;SING;" ! Single sweep and stop for print
140 ENTER @Nwa;Reply ! Read the 1 when complete
150 !
160 OUTPUT @Nwa,"CLES;" ! Clear status registers
170 OUTPUT @Nwa,"ESNB128;" ! Enable copy complete
180 OUTPUT @Nwa,"SRE4;" ! Enable Event Status Register B
190 OUTPUT @Nwa,"PRNTRAUTF OFF;" ! Set printer auto feed off
200 OUTPUT @Nwa,"PRNTYPTJ;" ! Select ThinkJet printer
210 OUTPUT @Nwa,"PRNPRTSERI;" ! Select serial port for output
220 OUTPUT @Nwa,"PRNTRBAUD 9600;" ! Set baud rate to 9600 bps
230 OUTPUT @Nwa,"PRNHDHSHK XON;" ! Use Xon-Xoff handshake
240 !
250 OUTPUT @Nwa,"PRIMALL;" ! Print screen
260 !
270 DISP "PRINTING"
280 !
290 ! Set up next measurement over HP-IB
300 OUTPUT @Nwa,"CONT;" ! Restore continuous sweep
310 !
320 ! Measurements can continue but wait for print to finish
330 REPEAT ! Test for bit 2 (4) ESRB
340 Stat=SPOLL(@Nwa)
350 UNTIL BIT(Stat,2) ! Wait for printer to complete
360 !
370 DISP "DONE"
380 LOCAL @Nwa ! Release HP-IB control
390 END

3-64 HP BASIC Programming Examples
Example 7B1: Plotting to a File and Transferring File Data to a Plotter

Note
This program is stored as EXAMP7B1 on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

Another report-generation technique is to transfer the plotter string to a disk file, and retrieve and plot the disk file at another time. Test time is increased when a plot occurs during the process. It may be more convenient to plot the data at another site or time. One solution to this problem is to capture the plot data using the controller and store it to a disk file. This disk file may then be read from the controller and the contents transferred to a plotter. This next example shows a method of accomplishing this task.

The analyzer is initialized without presetting the analyzer. The data that is in place on the analyzer is not disturbed by the program operation. A large string is dimensioned to hold the plotter commands as they are received from the analyzer. The length of this string depends upon the complexity of the analyzer's display. The analyzer is placed in the single-sweep mode and 0PC?;STING; is used to make sure that operation is complete before plotting. The plotting begins with the 0UTPLOT; command.

The string transfer is ended by the controller detecting the EOI line which the analyzer pulls at the end of the transfer. The string transfer terminates and the plot data is now stored in a string in the analyzer.

These strings contain ASCII characters which represent the plotter commands in HP-GL (Hewlett-Packard Graphics Language). A disk file is created and the string is written into the file containing the display-plot commands.

Once the strings are transferred to the disk file, the file pointer is rewound and the data read out into a string for plotting. The string is sent to the plotter which uses the commands to generate a plot.

The following is an outline of the program's processing sequence:

- An I/O path is assigned for the analyzer
- An I/O path is assigned for the plotter
- The system is initialized
- The string for plotter commands is defined
- The frequency span is swept once
- The plotter output is requested and read into the plot string
- A plot file is created in the controller
- The plot string is stored into the disk file
- The plot string is read from the disk file and sent to the plotter
- The analyzer returns to continuous-sweep mode
- The analyzer is returned to local control and the program ends
The program is written as follows:

10 ! This program shows how to read the plotter output from the analyzer
20 ! and store it in a disk file as an ASCII file. The disk file is then
30 ! read back into the controller and the plot commands sent to a
40 ! plotter to generate the plot of the analyzer display. This allows
50 ! plotting at a different time than data collection.
60 !
70 ! EXAMP7B
80 !
90 ASSIGN @Nwa TO 716 ! Assign an I/O path for the analyzer
100 ASSIGN @Plt TO 705 ! Assign an I/O path for the plotter
110 !
120 CLEAR SCREEN
130 ! Initialize the analyzer without preset to preserve data
140 ABORT 7 ! Generate an IFC (Interface Clear)
150 CLEAR @Nwa ! SDC (Selected Device Clear)
160 !
170 DIM Plot$[32000] ! Define string for plotter commands
180 !
190 OUTPUT @Nwa;"OPC?;SING;" ! Stop sweep for plot and wait
200 ENTER @Nwa;Reply ! Read the 1 when complete
210 OUTPUT @Nwa;"OUTP;PLOT;" ! Request plotter output
220 !
230 ENTER @Nwa;Plot$ ! Plotter output of analyzer display
240 !
250 INPUT "Plotter output complete. Press RETURN to store on disk.",Reply$
260 !
270 ! Disk file operations
280 ! Create data file on disk 32000/256 = 125 records
290 !CREATE ASCII "PLOTFILE:,1400",125 ! Use only once to generate file
300 ASSIGN @File TO "PLOTFILE:,1400" ! Assign file I/O path
310 OUTPUT @File;Plot$ ! Write plot string to file
320 !
330 INPUT "Plot to file is complete. Press Return to plot.",A$!'
340 !
350 ! Read plotter commands from file and send to plotter
360 RESET @File ! Reset file pointer to beginning
370 ENTER @File;Plot$ ! Read plot string from file
380 OUTPUT @Plt;Plot$ ! Send plot string to plotter
390 !
400 !
410 DISP "Plot is complete. End of program."
420 OUTPUT @Nwa;"CONT;" ! Restore continuous sweep
430 OUTPUT @Nwa;"OPC?;WAIT;" ! Wait for analyzer to finish
440 ENTER @Nwa;Reply ! Read the 1 when complete
450 LOCAL @Nwa ! Release HP-IB control
460 END

3-66 HP BASIC Programming Examples
Running the Program

The program begins by initializing the analyzer and placing it into single-sweep mode. The plotter commands are captured into strings in the controller. The controller display prompts Plotter output complete. Press RETURN to store on disk. Pressing [RETURN] causes the data to be stored to disk. Once this task is complete, the program prompts once more, Plot to file is complete. Press Return to plot. After pressing [RETURN] again, the string output is sent to the plotter and the plot begins. Once the plot is complete, the program prompts Plot is complete. End of program, and the analyzer begins sweeping and returns to local control.

Plotting to a File and Utilizing PC-Graphics Applications

A slightly modified version of the previous example may be used to merely generate a file on the controller disk. This disk file can then be read into the PC and used in several different graphics-generation programs. HP-GL is a commonly recognized graphic format and may be used to transfer information to PC application programs such as CorelDRAW®, Lotus Freelance® and other graphics packages. Importing the graphics data into these application packages allows report generation in a variety of word-processors. Graphic-data files may then be used to document test results, generate data sheets from testing results, or for simply archiving on a digital-storage medium that can be recalled for later use.

When working with HP BASIC, a utility is available to convert the HP LIF (Logical Information Format, used in the Series 9000 workstations) to a PC-DOS format file. The utility is called "LIF2DOS.EXE". This utility is available on the “HP BASIC Programming Examples” disk (HP part number 08753-10028) that was received with the analyzer. A more extensive utility called “LIFUTIL” allows file transformation in both directions and is available as a commercial product. Contact the nearest HP sales and service office for ordering information.

Once the HP-GL file is present in the DOS file system, the HP-GL file is imported and examined with the graphics package. The text labels may need to be rescaled, but on the whole, the graphics results are quite usable.
Example 7B2: Reading Plot Files From a Disk

Note
This program is stored as EXAMP7B2 on the “HP 8753D/Option 011 Programming Examples” disk received with the network analyzer.

This example program reads and plots files which have been stored on a LIF formatted disk by the HP 8753D/Option 011. The plots may be sent to either a plotter with auto-feed capability, such as the HP 7550B, or an HP-GL/2 compatible printer, such as a LaserJet 4 Series (monochrome) printer or a DeskJet 1200C (color) printer.

Note
Sending plots to disk is discussed in the chapter titled “Printing, Plotting, and Saving Measurement Results” in the HP 8753D/Option 011 Network Analyzer User’s Guide.

This section provides detailed information on file naming conventions and instructions on printing multiple plots-per-page.

This program requires HP BASIC 6.0 or greater which provides the use of wild cards with the catalog command.

The peripheral HP-IB addresses and the hard copy device selection are hard coded and may need to be changed for different systems configurations. Turn to “Example 7B2” in the next chapter and refer to lines 1130 through 1240 in the example program.

This program example provides the form feeds to separate the plots. If the HP 8753D/Option 011 has been configured to store multiple plots-per-page, this program will generate those plots. A file naming convention has been devised to allow the program perform several printer-setup functions. These include: initializing the printer for HP-GL/2 at the beginning of a page, configuring the printer to plot multiple plots to the same page, if desired, and then sending a page eject (form feed) to the harcopy device at the completion of the printing process.

The plot file name is made up of four parts. The first three are generated automatically by the HP 8753D/Option 011 whenever a plot is requested:

1. the prefix “PLOT”
2. a two-digit sequence number in the range of 00 to 31
3. a two-letter output-format code to indicate the plot quadrant position:
   - LU (Left Upper)
   - LL (Left Lower)
   - RU (Right Upper)
   - RL (Right Lower)
   - FP (Full Page)

For example, the first full page plot to a disk would be named “PLOT00FP”. The second plot, to be located in the lower-right hand corner of the page would be named “PLOT01RL,” and so on.

The fourth part is an optional character. It is used to indicate that the file is part of a multiple-file plot on the same graticule. See “Printing Multiple Measurements Per Page” in chapter 3 of the HP 8753D/Option 011 Network Analyzer User’s Guide.
For detailed information on plotting to disk and outputting the plot files to a printer/plotter see "Printing Plotting, and Saving Measurements Results" in the HP 8753D/Option 011 Network Analyzer User's Guide.

The following is an outline of the program's processing sequence:

- Hardcopy device control strings are created.
- The hardcopy output device is defined.
- The disc storing the file names is cataloged.
- Files that match the name specifier in the file name array (Flnm$) are counted.
- Flnm$ is dimensioned for the number of files that match the name specifier.
- The file name array is sorted.
- A form feed is sent to the hardcopy device.
- Each of the files in the file name array is processed and sent to the output device.
  - The current file name root string is defined.
  - If the hardcopy device is a printer, then an HP-GL initialization string is output.
  - Each file which matches the file name root string to the hardcopy device is output.
  - After a form feed is sent to the hardcopy device, the printing/plotting process begins.

The program is written as follows:

```
10 ! This example program reads and plots files which have been stored on
20 ! a LIF formatted disk by the HP 8753D. The plots are sent to either a
30 ! plotter with auto-feed capability, such as the HP7550B, or an HP-GL/2
40 ! compatible printer, such as the LaserJet 4 series (monochrome) or the
50 ! DeskJet 1200C (color).
60 !
70 ! Sending plots to disk is discussed in the Printing, Plotting and Saving
80 ! Measurement Results section of the HP 8752C Users Guide. The file
90 ! naming conventions are discussed in this section and will provide more
100 ! details.
110 !
120 ! This program example will provide form feeds to separate the plots. If
130 ! multiple plots per page have been stored, this program will generate
140 ! those plots.
150 ! A file naming convention has been devised to allow the program
160 ! to initialize the printer for HP-GL/2 at the beginning of a
170 ! page, to plot multiple plots to the same page, if desired, and
180 ! when all plots to the same page have been completed then send a
190 ! page eject (form feed) to the hardcopy device.
200 !
210 ! The plot file name is made up of four parts, the first three of which
220 ! are generated automatically by the HP 8753D whenever a plot is requested:
230 ! The prefix, "PLOT", a two digit sequence number, in the range of 00 to
240 ! 31, a two letter output format code to indicate the plot quadrant
250 ! position or full page, LU (Left Upper), LL (Left Lower), RU (Right
260 ! upper), RL (Right Lower) or FP (Full Page). For example, the first
270 ! full page plot to a disk would be named "PLOT00FP".
280 !
290 ! a. Build hardcopy device control strings.
300 ! b. Define output hardcopy device.
```
c. Catalog disc storing the file names which match file name specifier
in the file name array, Flnm$, and setting the number of files that
match.

d. Dimension Flnm$ for the number of files matched.
e. Sort the file name array.
f. Form feed the hardcopy device.
g. Process each of the files in the file name array.

1. Define the current file name root string.

2. If hardcopy device is a printer then output HP-GL initialization

3. Output each file which matches the file name root string to the

4. Output a form feed to the hardcopy device.

This program requires HP BASIC 6.0 or greater which provides the use of
wild cards with the catalog command.

The peripheral HP-IB addresses and the hard copy device selection are
hard coded and may need to be changed for different systems
configurations. See lines 1130 to 1240.

EXAMP7E2

OPTION BASE 1
DIM Flnm$(1:200)[14]
DIM Hpglinit$[80]
DIM Srch$[60]
DIM Esc_chr$[1]
INTEGER Plt_array1(1:32767)
INTEGER Plt_array2(1,2,1:32767)
INT Pltttr
INTEGER Prntx
INTEGER Outputdcv
INTEGER Root_match
INTEGER Flnm_idx
INTEGER Nbr_files
INTEGER Prfx_length
INTEGER Root_length
INTEGER Array1sz
INTEGER Array2sz
INTEGER Plttr_addr
INTEGER Prtnr_addr
REAL Rcrd_lengths
REAL Nmbr_records
REAL Nmbr_words

! Enable HP-UX style wild cards
! Set numeric arrays to start at 1
! Plot filename array
! Printer HPGL initialization string
! Search string for plot filenames
! Escape character ASCII 27
! Plotter command array
! Additional plot arrays if required
! Plotter for output
! Printer for output
! Output device selected
! Root plot file flag
! Pointer to filename array
! Number of files which are plot files
! Length of prefix defined in filename
! Length of root name in filename
! Number of data works in plot file
! Number of arrays in plot file
! Plotter address
! Printer address
! Record length in plot file
! Number of records in plot file
! Number of data words in plot file

! Escape character (1B hex) ASCII 27 (Decimal)

*** Build control string for printers ***

Build hardcopy device control string containing setup commands for
printer output.

Reset, conditional page eject

Hpglinit$=Esc_chr$"E"
860 ! Page size A 8.5 x 11
870 Hpglinit$=Hpglinit$&Esc_chr$"&12A"
880 ! Landscape orientation
890 Hpglinit$=Hpglinit$&Esc_chr$"&110"
900 ! No left margin
910 Hpglinit$=Hpglinit$&Esc_chr$"&a0L"
920 ! No right margin
930 Hpglinit$=Hpglinit$&Esc_chr$"&a400M"
940 ! No top margin
950 Hpglinit$=Hpglinit$&Esc_chr$"&l0E"
960 ! Picture frame size 10.66 inches x 7.847 inches
970 ! (720 decipoints per inch)
980 Hpglinit$=Hpglinit$&Esc_chr$"*c7680x5650Y"
990 ! Move cursor to anchor point
1000 Hpglinit$=Hpglinit$&Esc_chr$"*p50x50Y"
1010 ! Set picture frame anchor point
1020 Hpglinit$=Hpglinit$&Esc_chr$"*cOT"
1030 ! Set CMY palette
1040 Hpglinit$=Hpglinit$&Esc_chr$"*r-3U"
1050 ! Enter HPGL mode with the cursor (pen) at the PCL current save position
1060 !
1070 ! Exit HPGL mode to accept printer command
1080 Hpgl_exit$=Esc_chr$"%0A"
1090 !
1100 ! Conditional form feed (page eject)
1110 Form_feed$=Esc_chr$"E"
1120 !
1130 ! *** Initialize variables and assign output device ***
1140 !
1150 ! Define device selection flags to determine plotter or printer
1160 ! Select device with 1 and set Outputdvc to define it
1170 Plttr=1 ! Select plotter for output
1180 Prntr=0 ! Select printer for output
1190 Outputdvc=Plttr ! define output device as plotter
1200 !
1210 False=0 ! Define flags for logic tests
1220 True=1
1230 !
1240 Prfx$="PLOT" ! define plot file name prefix string
1250 !
1260 ! *** Initialize HP-IB device addresses ***
1270 !
1280 Prntr_addr=701 ! Printer HP-IB address
1290 Plttr_addr=705 ! Plotter HP-IB address
1300 !
1310 ! Set address of flexible disk drive containing plot files
1320 Msis$=":1400"
1330 !
1340 ! Define I/O paths for plot output with no formatting on data
1350 IF Outputdvc=Plttr THEN ! select output device for plotting
1360 ASSIGN @Prntpltdvc TO Plttr_addr;FORMAT OFF ! Select plotter
1370 ELSE
1380 ASSIGN @Prntpltdvc TO Prntr_addr;FORMAT OFF ! Select printer
1390 END IF
1400 !
*** Search disk for plot files ***

! Define the plot file name specifier: Prefix (Prfx$), two
! sequence digits([0-9][0-9]), two output format specifier
! characters ([FLR][LPU]) and an optional character (s)
Srch$=Prfx$&"[0-9][0-9][FLR][LPU]*"

! Catalog the files that match the plot file name specifier by
! putting the names in the string array Flnm$ and the number of files
! in the integer Nbr_files. Suppress the catalog header text.
CAT Srch$&Msi$ TO Flnm$(*) ; COUNT Nbr_files, NO HEADER, NAMES

! If no files are found then print message and stop
IF Nbr_files=0 THEN
 PRINT "No files found; program terminated"
 STOP
 END IF

! *** Sort the plot file names found ***

! Re-dimension the file name array to the actual number of files that
! were found.
REDIM Flnm$(1:Nbr_files)

! Sort the file names into alphabetical order
MAT SORT Flnm$(*)

! Send a form feed to hardcopy device
GOSUB Frm_fd
 PRINT

! *** Cycle through the filenames and plot the data ***

! Initialize the Flnm$ array index
Flnm_idx=1
 Prfx_length=LEN(Prfx$)

! Find the length of the Prfx$ string

! Process each of the files in the Flnm$ array
WHILE Flnm_idx<=Nbr_files
! Define Root$ = file name prefix string plus the two sequence digits
 Root$=Flnm$(Flnm_idx)[1;Prfx_length+2]

! Find length of the Root$
 Root_length=LEN(Root$)

! If the two output specifier characters are "FP" (full page) then
! include them as part of the Root$
 IF Flnm$(Flnm_idx)[Root_length+1;2]="FP" THEN
 Root$=Root$&"FP"

 END IF
 Root_length=Root_length+2

 END IF

! initialize file matches Root$ flag
Root_match=True

! If the output device is a printer send HP-GL initialization string
IF OutputDvc=Prnt THEN
 OUTPUT @Prntpltdvc USING ":,K" ; Hpglinit$

 END IF

! *** Plot files on the same page ***
1960 !
1970 ! While the root portion of the file names match Root$, output the
1980 ! plot files to the same page.
1990 WHILE Root_match=True
2000 ! Print the name of the plot file
2010 PRINT Flnm$(Flnm_idx),
2020 ! Output the plot file to the hardcopy device
2030 GOSUB 0pt_fl
2040 ! Increment Flnm$ array index
2050 Flnm_idx=Flnm_idx+1
2060 ! If all plot files have been output or the plot file name does not
2070 ! match Root$ then set Root_Match=False to end plotting of the same
2080 ! page.
2090 IF Flnm_idx>Nbr_files THEN
2100     Root_match=False
2110 ELSE
2120     IF Root$Flnm$(Flnm_idx)[1;Root_Lngth] THEN Root_match=False
2130 END IF
2140 END WHILE ! Loop to plot to the same page
2150 !
2160 PRINT
2170 ! Output form feed to output device to eject page
2180 GOSUB Frm_fd
2190 END WHILE ! Loop to plot next page
2200 !
2210 STOP
2220!
2230 !*************************************************************************Subroutines*************************************************************************
2240 !
2250 0pt_fl: ! Read the file into an array(s) and then output the array(s)
2260 ! to the hardcopy device.
2270 !
2280 ! Open and read the plot file size
2290 ASSIGN @Ldisc TO Flnm$(Flnm_idx)&Msi$
2300 !
2310 ! Get number of records in file from I/O path status registers
2320 STATUS @Ldisc,3;Nmbr_rcrd$
2330 !
2340 ! Get record length
2350 STATUS @Ldisc,4;Rcrd_lngth
2360 !
2370 ! Compute the number of words of data in the file
2380 Nmbr_wrds=Nmbr_rcrd*$Rcrd_lngth/2
2390 !
2400 ! Determine the number of arrays necessary to hold the plot file data.
2410 ! The maximum size of an RMB array is 32767. If the number of words of
2420 ! data is greater than 32767 then multiple arrays must be used to hold
2430 ! all of the data.
2440 !
2450 ! Compute the dimensions of Plt_arr1 and Plt_arr2 that are required
2460 ! to hold the plot data.
2470 Arry1_sz=Nmbr_wrds MOD 32767
2480 Arry2_sz=INT(Nmbr_wrds/32767)
2490 !
2500 ! Re-dimension Plt_arr1 to the correct size
2510 REDIM Plt_arr1(1:Arry1_sz)
2520 !
2530 ! If the number of words of data is less than 32767 then use one array
2540 IF Arry2_sz=0 THEN
2550 ENTER @Ldisc;Plt_arr1(*)  ! Read the plot data from the file
2560 ELSE
2570 ! Use 2 arrays to read data
2580 ENTER @Ldisc;Plt_arr2(*),Plt_arr1(*)  ! Read the plot data from file
2590 END IF
2600 !
2610 ASSIGN @Ldisc TO *  ! Close plot file
2620 !
2630 ! Output the data to the hardcopy device
2640 IF Arry2_sz=0 THEN  ! Only one array <32767 words
2650 OUTPUT @Prntpltdvc;Plt_arr1(*)
2660 ELSE  ! Data > 32767 so send 2 arrays
2670 OUTPUT @Prntpltdvc;Plt_arr2(*),Plt_arr1(*)
2680 END IF
2690 RETURN
2700 !
2710 !***************************************************************************
2720 !
2730 Frm_fd:! Send a form feed (page eject) command to the hardcopy device
2740 IF Outputdvc=Plttxt THEN  ! Plotter output
2750 OUTPUT @Prntpltdvc USING "#,K","PG,"
2760 ELSE  ! For printer output. The
2770 ! printer first must exit HP-GL mode before sending form feed command
2780 OUTPUT @Prntpltdvc USING ",,K",Hpgl_exit$&Form_feed$
2790 END IF
2800 RETURN
2810 !
2820 END

Running the Program

This program allows you to plot or print as many as four HP 8753D/Option 011 display dumps (stored on a LIF formatted disk) on one side of a single sheet of paper. Refer to the instructions detailed in the chapter “Printing, Plotting, and Saving Measurement Results,” in the HP 8753D/Option 011 Network Analyzer User’s Guide.
Example 7C: Reading ASCII Disk Files to the System Controller Disk File

Note This program is stored as EXAMP7C on the "HP 8753D/Option 011 Programming Examples" disk received with the network analyzer.

Another way to access the analyzer’s test results is to store the data onto a disk file from the analyzer. This operation generates an ASCII file of the analyzer data in a CITIFILE format. A typical file generated by Example 7C is shown below:

```
CITIFILE A.01.00
#NA VERSION HP8753C.04.13
NAME DATA
VAR FREQ MAG 11
DATA S[1,1] RI
SEG_LIST_BEGIN
SEG 100000000 200000000 11
SEG_LIST_END
BEGIN
8.30566E-1,-1.36749E-1
8.27392E-1,-1.43676E-1
8.26908E-1,-1.52069E-1
8.25653E-1,-1.60035E-1
8.26385E-1,-1.68029E-1
8.26507E-1,-1.77154E-1
8.26263E-1,-1.87316E-1
8.26721E-1,-1.97265E-1
8.2724E-1,-2.07611E-1
8.28552E-1,-2.19940E-1
8.29620E-1,-2.31109E-1
END
```

This data file can be stored from the analyzer by remote control or by front-panel operations. See “Saving Instrument States” in the Printing, Plotting, and the Saving Measurement Results chapter in the HP 8753D/Option 011 Network Analyzer User’s Guide for more details on manual operation.

This program stores a file in the same manner as an operator would store a file on to the analyzer’s internal disk drive from the front panel. If a PC is being used as the system controller, the disk format will have to be transformed from LIF to PC-DOS format. A file utility is available to perform this format translation. The utility is called “LIF2DOS.EXE”. This utility is available on the “HP BASIC Programming Examples” disk (HP part number 08753-10028) that was received with the analyzer. Once transformed, the PC will now have a DOS-format file to read and interpret.

This example explains the process of storing the data from the analyzer to a file on the internal disk drive. There is also a program to read the data from the file into a data array for further processing or reformattting to another file type. The internal drive will store in the same format that is present on the disk. A new disk may be formattted in either LIF or DOS. For the example, the assumption has been made that the format transformation has already taken place, and there is a file that can be read record by record, from which data can be retrieved.

The goal of this example is to recover an array of stimulus frequency along with the trace-data values. CITIFILES contain the real and imaginary values of each data point. Some further transformation will be required to obtain magnitude values, for example.
The disk file contents for this example are shown above. This file contains more information than will be used in this example. The file is accessed and the records read from the file and printed on the controller display to observe the actual file contents. The file pointer is reset and the records are then read and interpreted for their data contents.

The first six records are skipped for this example. The seventh record contains the stimulus-frequency values and the number of points in the trace. These values are read from the record. The frequency increment, or point spacing, is calculated and used later in the frequency-data calculations for a point. Two more records are skipped and the next is the first record representing data values. The data values are read in a loop until the values for the number of points have been recovered from the file. The data values are tabulated and printed out on the controller display.

The following is an outline of the program’s processing sequence:

- An I/O path is assigned to the analyzer.
- The system is initialized.
- A string is dimensioned to hold a file record.
- The analyzer operating state is set.
- The internal drive is selected for storage (only ASCII data is stored).
- A file name is entered and the data stored into it.
- The operator is prompted to move the disk to the controller disk drive.
- The disk file is read and the contents displayed.
- The file pointer is rewound.
- The file contents are converted to trace data.
- The frequency and complex-data pair is displayed for each point.
- The analyzer is restored to continuous-sweep mode.
- The analyzer is returned to local control and the program ends.

The program is written as follows:

```
10 ! This program shows how to store an ASCII data file in CITIFILE format
20 ! and retrieve the data with the controller. The disk is written in the
30 ! analyzer system and then moved to the controller disk and the data
40 ! accessed.
50 !
60 ! EXAMP7C
70 !
80 ASSIGN ONwa TO 716 ! Assign an I/O path for the analyzer
90 !
100 CLEAR SCREEN ! Generate an IFC (Interface Clear)
110 ABORT 7 ! SDC (Selected Device Clear)
120 CLEAR ONwa
130 OUTPUT ONwa;"OPC?;PRES;"
140 ENTER ONwa;Reply ! Preset the analyzer and wait
150 !
160 DIM Record$[80] ! Read the 1 when complete
170 ! String to read the disk records
180 ! Set up analyzer
190 OUTPUT ONwa;"STAR100MHZ;"
200 OUTPUT ONwa;"STOP 200MHZ" ! Start frequency 100 MHz
201 ! Stop frequency 200 MHz
```

3-76 HP BASIC Programming Examples
210 OUTPUT @Nwa:"P0IN11;" ! Trace length 11 points
220 OUTPUT @Nwa:"OPC?;SING;" ! Single sweep and wait
230 ENTER @Nwa;Reply ! Read in the 1 when complete
240 !
250 ! Program disk storage operation
260 !
270 OUTPUT @Nwa:"INTD;" ! Select internal disk file
280 OUTPUT @Nwa:"EXTMFORMON;" ! Store formatted data
290 OUTPUT @Nwa:"EXTMDATEON;" ! Store data file only
300 INPUT "Enter data file name (5 chars),File_name$" ! Get file name
310 File_name$=UPC$(File_name$) ! File names are uppercase
320 OUTPUT @Nwa:"TITF""C";File_name$,"C";" ! Title for save reg 1
330 OUTPUT @Nwa:"SAVUASCI;" ! Save as ASCII file
340 !
350 OUTPUT @Nwa:"STOR1;" ! Store data to disk file
360 OUTPUT @Nwa:"OPC?;WAIT;" ! Wait until store is complete
370 ENTER @Nwa;Reply
380 !
390 ! File storage is complete
400 !
410 INPUT "Place disk in controller disk drive, then press Return";A$
420 !
430 ! Read data file information
440 !
450 ASSIGN @File TO File_name$"D1:,1400" ! Open an I/O path for file
460 Record_cnt=1 ! Counter to count records
470 !
480 PRINT CHR$(12); ! Formfeed to clear display
490 PRINT "Contents of data file" ! Show contents of the data file
500 Readfile: !
510 ON END @File GOTO End_file ! Test for end of file and exit
520 ENTER @File;Record$ ! Read ASCII record
530 PRINT Record_cnt,Record$ ! print record on display
540 Record_cnt=Record_cnt+1 ! Increment record counter
550 GOTO Readfile ! Read next record
560 !
570 END_file: ! ! Reached the end of file
580 PRINT "End of File. ";Record_cnt-1;" Records found"
590 INPUT "Press Return to continue";A$
600 PRINT CHR$(12); ! Formfeed to clear display
610 !
620 ! Read file data into arrays
630 !
640 RESET @File ! Rewind file pointer to beginning
650 FOR I=1 TO 6
660 ENTER @File;Record$ ! Skip first six records
670 NEXT I
680 ENTER @File;Record$ ! Read frequency data record
690 Record$=Record$[POS(Record$," ")+1] ! skip SEG to first space + 1
700 Startf=VAL(Record$) ! Read start frequency
710 Record$=Record$[POS(Record$," ")+1] ! Skip to next space + 1
720 Stopf=VAL(Record$) ! Read stop frequency
730 Record$=Record$[POS(Record$," ")+1] ! Skip to next space + 1
740 Num_points=VAL(Record$) ! Read the number of points
750 PRINT " Number of points in file ";Num_points

HP BASIC Programming Examples 3-77
760 PRINT
770 !
780 Freq_inc=(Stopf-Startf)/(Num_points-1) ! Compute frequency increment
790 !
800 ALLOCATE Array(Num_points,2) ! Allocate array from Num_points
810 ENTER @File;Record$ ! Skip SEG_LIST_END record
820 ENTER @File;Record$ ! Skip BEGIN record
830 !
840 ! Read in the data array
850 PRINT "Freq (MHz) Data 1 Data 2" ! Table header for data array
860 FOR I=1 TO Num_points ! Read in array entries
870 ENTER @File;Record$ ! Read in the record of 2 entries
880 !
890 Array(I,1)=VAL(Record$) ! Read first data value
900 Data$=Record$[POS(Record$","")+1] ! Skip to comma and next value
910 Array(I,2)=VAL(Data$) ! Read second data value
920 !
930 Freq=Startf+(Freq_inc*(I-1)) ! Compute stimulus value for array
940 Freq=Freq/1.E+6 ! Convert frequency to MHz
950 !
960 PRINT Freq,Array(I,1),Array(I,2) ! Print data array values
970 NEXT I ! Read next array data points
980 !
990 OUTPUT @Nwa:"CONT;" ! Restore continuous sweep
1000 OUTPUT @Nwa:"CPC?;WAIT;" ! Wait for analyzer to finish
1010 ENTER @Nwa;Reply ! Read the 1 when complete
1020 LOCAL @Nwa ! Release HP-IB control
1030 END

Running the Program

The analyzer is initialized and the operating range re-defined to an 11-point trace from
100 to 200 MHz. This setup gives a restricted range to be evaluated when the ASCII data file
(CITIFILE) is read in from the controller. The operator is prompted for a 5-character filename
to use for storing the data. The analyzer is setup for external storage and stores the data file.
Once the “pass control/storage/return control” operation is complete, the operator is prompted
to place the disk in the controller disk drive and press (RETURN). The disk is then read and
the records contained in the file are printed on the controller display. A prompt appears,
Press return to continue, which allows viewing of the file contents. Once (RETURN) is
pressed, the data records are read and decoded and a table of the stimulus frequency and the
data values are printed.
Index

A
abort sequence, 3-8
additional information, 3-1
  BASIC 6.2, 3-1
analyzer operating modes, 3-3
  pass-control mode, 3-3, 3-61
  system-control mode, 3-3
  talker/listener, 3-3, 3-59
array-data formats, 3-22
  FORM 1, 3-22
  FORM 2, 3-22
  FORM 3, 3-20, 3-22
  FORM 4, 3-22
  FORM 5, 3-22
ASCII disk files, 3-75
  reading, 3-75
C
calibration data, 3-42
  inputting, 3-42
  outputting, 3-42
  reading, 3-42
  calibration kit, 3-2
clearing any messages waiting to be output, 3-8
clearing syntax errors, 3-8
clearing the input-command buffer, 3-8
clear sequence, 3-8
command structure, 3-4
command structure elements, 3-4
  appendage, 3-4
  BASIC command statement, 3-4
data, 3-4
  terminators, 3-4
  unit, 3-4
compatible peripherals, 3-2
connecting the test system, 3-2
D
data transfer, 3-20
  to a plotter, 3-65
  using floating-point numbers, 3-25
  using FORM 1, 3-29
  using FORM 4, 3-22
  using frequency-array information, 3-27
  using markers, 3-20
dump mode, 3-5
E
equipment
  optional, 3-2
  required, 3-1
error queue, 3-31
event-status-register B, 3-56
event-status registers, 3-31
example
  plotting plot files stored on disk, 3-68
  printing plot files stored on disk, 3-68
F
frequency calculation equation, 3-22
G
graphics applications, utilizing, 3-67
H
held commands, 3-7
HP 9000 Series 300 computer, 3-1
HP-IB interconnect cables, 3-1
I
input buffer, 3-7
input/output path, 3-9
instrument states, 3-40
  recalling, 3-40, 3-45
  saving, 3-40, 3-45
interrogating commands, 3-6
interrupts, generating, 3-33
L
limit lines, 3-53
  setting up, 3-53
limit-line testing, 3-48
  list-frequency table, creating, 3-48
  list-frequency table, selecting a single segment, 3-51
  performing PASS/FAIL tests, 3-53
  using list-frequency mode, 3-48
limit-test table, 3-53
  creating, 3-53
  transmitting, 3-53
list-frequency mode, 3-48
local lockout, 3-5
local mode, 3-5

M
marker positioning, 3-20
  by data point location, 3-20
  by frequency location, 3-20
  by trace-data value, 3-20
measurement parameters
  required order, 3-10
  setting, 3-10
  verifying, 3-12
measurement setup, 3-10
measurement specifications, 3-25
  group delay, 3-25
  magnitude, 3-25
  phase, 3-25
memory requirements, 3-1

O
ON TIMEOUT, 3-7
operation complete commands, 3-7

P
PASS/FAIL tests, 3-56
plotting
  to a file, 3-65, 3-67
plotting plot files stored on disk example program, 3-68
plotting, remote, 3-59, 3-61
power meter calibration, 3-36
preparing for remote operation, 3-8
presetting the instrument, 3-8
printing
  using the serial port, 3-63
printing plot files stored on disk example program, 3-68
printing, remote, 3-59, 3-61
program example
  plotting plot files stored on disk, 3-68
  printing plot files stored on disk, 3-68

R
recommended disk drives, 3-2
recommended plotters, 3-2
recommended printers, 3-2
remote mode, 3-5
report generation, 3-59

S
serial poll, 3-31
service request, 3-33
setting addresses, 3-2
setting the control mode, 3-2
setting up the system, 3-2
status byte, 3-31
status reporting, 3-31
synchronization, 3-31
system setups, 3-40
  reading calibration data, 3-42
  using the learn string, 3-40

T
test port return cables, 3-2
transferring plot-file data to a plotter example program, 3-68
transferring plot-file data to a printer example program, 3-68
troubleshooting, 3-3, 3-5

V
verifying HP-IB operation, 3-2
Index

Special characters

$ 1-8

1
1-port measurement calibration example
program, 2-37

2
2-port measurement calibration program
example, 2-41

A
A/B, 1-42
AB, 1-42
abort message (IFC), 2-8
abort sequence, 3-8
acceptor handshake, 1-2
additional information, 3-1
  BASIC 6.2, 3-1
ADDRCONT[D], 1-42
ADDRDISC[D], 1-42
address
  controller, 1-42
  disk drive, 1-42
  peripheral, 1-42
  plotter, 1-42
  power meter, 1-42
  printer, 1-42
address capability, 2-5
addresses for HP-IB, 2-8
ADDRPERI[D], 1-42
ADDRPLOT[D], 1-42
ADDRPOWM[D], 1-42
ADDRPRIN[D], 1-42
adjust brightness, 1-43
adjust color, 1-45
adjust tint, 1-69
AF, 1-36
AH1, 1-2
AH1 (full-acceptor handshake), 2-5
ALTAB, 1-42
alternate inputs, 1-42
ANAB, 1-42
ANA1, 1-42
analog bus, 1-42
analog input, 1-42
analyzer array-data formats, 2-18
analyzer bus mode, 2-7
analyzer command syntax, 2-10
analyzer control of peripherals, 2-7
analyzer data reading, 2-14
analyzer-debug mode, 2-31
analyzer features helpful in developing
programs, 2-31
analyzer HP-IB capabilities, 1-2
analyzer identification, 1-4
analyzer operating modes, 3-3
  pass-control mode, 3-3, 3-61
  system-control mode, 3-3
  talker/listener, 3-3, 3-59
analyzer operation, 2-13
analyzer single bus concept, 2-6
analyzer status reporting structure, 2-24
analyzer system setups example program,
  2-74
appendage, 1-5
appendage in syntax, 2-11
AR, 1-42
array-data formats, 2-18, 3-22
  FORM 1, 3-22
  FORM 2, 3-22
  FORM 3, 3-20, 3-22
  FORM 4, 3-22
  FORM 5, 3-22
arrays of data, 2-21
arrays related to frequency, 2-20
ASCII
  save format, 1-64
ASCII disk files, 3-75
  reading, 3-75
ASCII disk files read to the instrument
  controller’s disk file example program,
  2-108
ASCII transfer example program, 2-51
ASEG, 1-42
assert sequence, 1-42
ASSS, 1-42
ATN (attention) control line, 2-4
attention (ATN) control line, 2-4
ATTP1, 1-42
ATTP2, 1-42
AUTO, 1-42
auto feed
plotter, 1-60
printer, 1-62
auto scale, 1-42
averaging, 1-42
restart, 1-42
averaging factor, 1-42
AVERFACT[D], 1-42
AVERO, 1-42
AVERREST, 1-42

B
BACI[D], 1-42
background intensity, 1-42
BANDPASS, 1-42
basic listener, 1-2
basic talker, 1-2
basic talker (T6), 2-5
baud rate
plotter, 1-60
printer, 1-62
beep
emit, 1-48
BEEPDONE, 1-42
beeper on done, 1-42
beeper on warning, 1-42
BEEPFAIL, 1-42
BEEPWARN, 1-42
begin cal sequence, 1-43
bi-directional lines, 2-4
binary
save format, 1-64
BR, 1-42
bus device modes, 2-6
bus structure, 2-2, 2-3

C
C1, 1-2
C10, 1-2
C10 (pass control capabilities), 2-5
C1, C2, C3 (system controller capabilities), 2-5
C1[D], 1-43
C2, 1-2
C2[D], 1-43
C3, 1-2
C3[D], 1-43
CAL1, 1-43
CALFCALF[D], 1-43
CALFFREQ[D], 1-43
CALFSENA, 1-43
CALFSENSB, 1-43
calibrating the test setup, 2-28
calibration
power meter, 1-62
calibration arrays, 1-34
calibration/classes relationship, 1-34
calibration coefficients, 2-21
calibration command sequence, 1-34
calibration data, 3-42
inputting, 3-42
outputting, 3-42
reading, 3-42
calibration data reading example program, 2-76
calibration example program, 2-37
calibration kit, 3-2
calibration kits, 1-43, 2-37
calibration kit string and learn string, 2-23
calibration (power meter) example program, 2-69
calibration type off, 1-43
CALIFUL2, 1-43
CALIONE2, 1-43
CALIRAI, 1-43
CALIRESP, 1-43
CALIS111, 1-43
CALIS221, 1-43
CALITRL2, 1-43
CALK35MD, 1-43
CALK35MM, 1-43
CALK7MM, 1-43
cal kit done, 1-52
CALKN50, 1-43
CALKN75, 1-43
CALKUSED, 1-43
CALN, 1-43
cal power
set port 1, 1-62
cal sensor table
edit, 1-43
cal sequence
begin, 1-43
resume, 1-63
carriage returns, 1-6
case distinctions, 1-5
CBRI[D], 1-43
CENT[D], 1-43
center, 1-43
chain for data processing, 2-21
CHAN1, 1-43
CHAN2, 1-43
channels
coupled, 1-45
characters that are valid, 2-11
CHOPAB, 1-43
ctifile
save format, 1-64
CLAD, 1-44
CLASS11A, 1-44

Index-2
CLASS11B, 1-44
CLASS11C, 1-44
CLASS22A, 1-44
CLASS22B, 1-44
CLASS22C, 1-44
class done, 1-44
CLEABIT, 1-44
CLEA<>, 1-44
CLEAL, 1-44
CLEARALL, 1-44
clear device, 2-8
CLEAREG<>, 1-44
clearing any messages waiting to be output, 3-8
clearing syntax errors, 3-8
clearing the input-command buffer, 3-8
clear list, 1-44
clear register, 1-44
clear sequence, 1-44, 3-8
CLEASEQ<>, 1-44
CLEL, 1-44
CLES, 1-44
CLS, 1-44
COAX, 1-44
COD, 1-43
code naming conventions, 2-10
code syntax structure, 2-11
COLOCH1D, 1-45
COLOCH1M, 1-45
COLOCH2D, 1-45
COLOCH2M, 1-45
COLOGRAT, 1-45
color
data channel 1, 1-59
data channel 2, 1-59
graticule, 1-59
memory channel 1, 1-59
memory channel 2, 1-59
text, 1-59
warning, 1-59
COLOR[D], 1-45
colors, 1-59
COLOTEXT, 1-45
COLOWARN, 1-45
? command, 2-14
command formats, 2-11
command interrogate, 1-3, 2-14
command naming, 1-7
commands
HP-IB, 1-1
command structure, 1-5, 3-4
command structure elements, 3-4
appendage, 3-4
BASIC command statement, 3-4
data, 3-4
terminators, 3-4
unit, 3-4
command syntax, 2-10
command syntax structure, 2-11
compatible peripherals, 3-2
complete operation, 2-13
complete service request capabilities (SR1), 2-5
computer controllers, 2-3
connecting the device under test, 2-28
connecting the test system, 3-2
CONS, 1-45
CONT, 1-45
continue sequence, 1-45
controlled sweep, 2-31
controller
address, 1-42
controller Interface function, 2-3
control lines, 2-4
controlling peripherals using pass-control
mode example program, 2-88
CONV1DS, 1-45
conventions for code naming, 2-10
CONVOFF, 1-45
CONVREF, 1-45
CONVTRA, 1-45
CONVZTRA, 1-45
copy display, 1-58, 1-60, 1-61
COPYFRFT, 1-45
COPYFRRT, 1-45
CORI, 1-45
CORR, 1-45
correction, 1-45
interpolative, 1-45
correction of errors example program, 2-37
COUC, 1-45
COUP, 1-45
coupled channels, 1-45
CRT focus, 1-49
CRT intensity, 1-51
CRT title, 1-59
CS, 1-36
CSWI, 1-45
CW freq, 1-45
CWFREQ[D], 1-45
CW time, 1-45
CWTIME, 1-45
D
[D], 1-8
D1DIVD2, 1-46
data
include with disk files, 1-48
data-array formats, 2-18
data arrays, 2-21
data bus, 2-4
data channel 1
  color, 1-59
data channel 2
  color, 1-59
data for markers, 2-16
data formats and transfers, 2-48
data levels, 2-22
data only
  include with disk files, 1-48
data-processing chain, 2-21
data processing chain, 1-38
data rate, 2-5
data reading, 2-14
data taking, 2-29
data transfer, 2-4, 2-29, 3-20
  to a plotter, 3-65
  using floating-point numbers, 3-25
  using FORM 1, 3-29
  using FORM 4, 3-22
  using frequency-array information, 3-27
  using markers, 3-20
data-transfer character definitions, 2-15
data transfer example program, 2-48
data transfer for traces, 2-19
data transfer using floating-point numbers, 2-54
data transfer using form 11 example program, 2-60
data transfer using form 4 example program, 2-51
data transfer using frequency array
  information example program, 2-56
data units, 1-4
date, 1-65
DATI, 1-46
DC1, 1-2
DC1 (complete device clear), 2-5
DCONV, 1-46
DEBU, 1-46
debug, 1-46
debug mode, 2-31, 3-5
decibels terminator code, 2-11
decimal point, 1-6
decrement loop counter, 1-46
DECRLOOC, 1-46
default calibration kits, 1-43
default colors, 1-46
DEFC, 1-46
definitions of status bit, 2-24
DEFLPRINT, 1-46
DEFLCPFO, 1-46
DEFS[D], 1-46
DELA, 1-46
delay, 1-46, 1-48
set to mkr, 1-54
delete segment, 1-64
DEL&<I>, 1-47
DELO, 1-46
DELRFIXM, 1-47
delta limits, 1-53
delta reference, 1-46, 1-47
DEMOAMPL, 1-47
demodulation off, 1-47
DEMOFF, 1-47
DEMOPHAS, 1-47
DeskJet, 1-62
developing program features, 2-31
device clear, 1-2, 2-8
device clear (DC1), 2-5
device connection, 2-28
device trigger, 2-9
device types for HP-IB, 2-2
DF, 1-36
DFLT, 1-47
DI, 1-36
directory size
  LIF, 1-47
DIRS[D], 1-47
disabling the front panel, 2-9
DISCUNIT[D], 1-47
DISCOVOL[D], 1-47
disk
  load file, 1-53
disk drive
  address, 1-42
disk drive unit, 1-47
disk drive volume, 1-47
disk file names, 1-40
disk files read to the controller's disk file
  example program, 2-108
disk format, 1-49
DISM, 1-47
DISPDATA, 1-47
DISPDATM, 1-47
DISPDDM, 1-47
display A/B, 1-42
display A/R, 1-42
display B/R, 1-42
display data, 1-47
display data — mem, 1-47
display data & mem, 1-47
display data/mem, 1-47
display data to mem, 1-46
display format units, 2-16
display graphics, 1-36
display memory, 1-47
DISPMM, 1-47
DIVI, 1-47
does not respond to parallel poll (PPO), 2-5
done
  with class, 1-47
  with isolation, 1-51
  with reflection, 1-63
  with transmission, 1-70
DONE, 1-47
done modify sequence, 1-48
DONM, 1-48
DOSEQ<l>, 1-48
do sequence, 1-48
DOS format, 1-49
DOWN, 1-48
down converter, 1-46
DTI, 1-2
DTI (responds to a group execute trigger), 2-5
dT, 1-61
DUAC, 1-48
dual channels, 1-48
duplicate sequence, 1-48
DUPLSEQ<X>SEQ<X>, 1-48
E
E2, 1-2
E2 (tri-state drivers), 2-5
edt cal sensor table, 1-43
EDITDONE, 1-48
edit limit table, 1-48
EDITLIML, 1-48
EDITLIST, 1-48
edit power loss range, 1-61
edit power loss table, 1-61
edit segment, 1-65
ELED[D], 1-48
EMIB, 1-48
emit beep, 1-48
end or identify, 1-5
dend or identify (EOI) control line, 2-4
ENTO, 1-48
entry off, 1-48
EOI, 1-5
EOI (end or identify) control line, 2-4
Epson-P2, 1-62
equipment
  optional, 3-2
  required, 3-1
error-corrected data, 2-21
error-correction example program, 2-37,
  2-41
error output, 2-27
er style, 3-31
er error queue use example program, 2-63
er reporting, 2-24
ESB?, 1-48
ESE[D], 1-48
ESNB[D], 1-48
ESR?, 1-48
event-status register, 2-24, 2-27
event-status-register B, 3-56
event-status registers, 3-31
element
  1. measurement setup, 2-32
controlling peripherals using pass-control
  mode, 2-98
data transfer using floating-point numbers,
  2-54
data transfer using form 1, 2-60
data transfer using form 4 (ASCII transfer),
  2-51
data transfer using frequency array
  information, 2-56
data transfer using markers, 2-49
full 2-port measurement calibration, 2-41
  generating interrupts, 2-65
limit-line testing, 2-83
measurement calibration, 2-37
measurement data transfer, 2-48
measurement process synchronization,
  2-62
network analyzer system setups, 2-74
operation using talker/listener mode, 2-96
performing PASS/FAIL tests while tuning,
  2-93
plotting plot files stored on disk, 2-106,
  3-68
plotting to a file and transferring the file
data to a plotter, 2-103
power meter calibration, 2-69
printing plot files stored on disk, 2-106,
  3-68
printing with the serial port, 2-101
reading ASCII disk files to the instrument
  controller’s disk file, 2-108
reading calibration data, 2-76
report generation, 2-96
S11 1-port measurement calibration, 2-37
saving and restoring the analyzer
  instrument state, 2-79
selecting a single segment from a table of
  segments, 2-86
setting parameters, 2-32
setting up limit lines, 2-89
storing and recalling instrument states,
  2-74
system initialization, 2-32
using the error queue, 2-63
using the learn string, 2-74
verifying parameters, 2-35
examples of programs, 2-30
EXTD, 1-48
EXTDATA, 1-48
extended listener capabilities (LEO), 2-5
external source mode, 1-51
external trigger, 1-48
EXTMDATO, 1-48
EXTMFORM, 1-48
EXTMGRAP, 1-48
EXTMRAW, 1-48
EXTTHIGH, 1-48
EXTTLOW, 1-48
EXTTOFF, 1-48
EXTTON, 1-48
EXTTPOIN, 1-48

F
features helpful in developing programming routines, 2-31
femtoseconds terminator code, 2-11
file names
disk, 1-40
file titles
recall, 1-63
firmware revision identification, 1-4
FIXE, 1-48
fixed load, 1-48
fixed marker, 1-47
flat line type, 1-53
floating-point numbers example program, 2-54
FOCU[D], 1-49
FORM1, 1-49
form 1 format, 2-18
FORM2, 1-49
form 2 format, 2-18
FORM3, 1-49
form 3 format, 2-18
FORM4, 1-49
form 4 format, 2-18
FORM5, 1-49
form 5 format, 2-18
format
disk, 1-49
format display units, 2-16
FORMATDOS, 1-49
FORMATLIF, 1-49
formats and transfers of trace-data, 2-48
formats for array-data, 2-18
formats for commands, 2-11
formatted data, 2-21
include with disk files, 1-48
form feed
plotter, 1-60

printer, 1-62
forward calibration class, 1-49
FREO, 1-49
FREQOFFS, 1-49
frequency calculation equation, 3-22
frequency list mode example program, 2-83
frequency notation, 1-49
frequency offset, 1-49
frequency offset value, 1-71
frequency-related arrays, 2-20
full 2-port measurement calibration program example, 2-41
full-acceptor handshake (AH1), 2-5
full-source handshake (SH1), 2-5
FULP, 1-49
FWDI, 1-49
FWDM, 1-49
FWDT, 1-49

G
GATE, 1-49
GATECENT[D], 1-49
gate center time, 1-49
gate on/off, 1-49
gate shape, 1-49
  maximum, 1-49
  minimum, 1-49
  normal, 1-49
  wide, 1-49
GATESPAN[D], 1-49
gate span time, 1-49
GATESTAR[D], 1-49
gate start time, 1-49
GATESTOP[D], 1-49
gate stop time, 1-49
GATSMAXI, 1-49
GATSMINI, 1-49
GATSNORM, 1-49
GATSWIDE, 1-49
general structure of syntax, 2-11
generating interrupts example program, 2-65
generation of report example program, 2-96
gigahertz terminator code, 2-11
GOSUB, 1-49
gosub sequence, 1-49
GPIO, 1-59
GPIO input bit, 1-59
GPIO output bits, 1-59
graphics
  character size, 1-37
  default values, 1-36
  display off, 1-36
  display on, 1-37
  draw to x,y, 1-36
  erase display, 1-36, 1-37
label display, 1-36
line type, 1-36
output scaling limits, 1-36
ten down, 1-36
ten up, 1-37
plot relative, 1-37
select pen, 1-37
graphics applications, utilizing, 3-67
graphics commands, 1-36
graticule
color, 1-59
group execute trigger response (DT1), 2-5
guidelines for code naming, 2-10

H

halting all modes and functions, 2-8
handshake
plotter, 1-60
printer, 1-61
handshake lines, 2-4
HARMOFF, 1-50
harmonic mode off, 1-50
HARMSEC, 1-50
HARMTHIR, 1-50
held commands, 1-2, 2-13, 3-7
helpful features for developing programs,
2-31
hertz terminator code, 2-11
HOLD, 1-50
HP 9000 Series 300 computer, 3-1
HP-GL
character size, 1-37
commands accepted but ignored, 1-37
default values, 1-36
display off, 1-36
display on, 1-37
draw to x,y, 1-36
erase display, 1-36, 1-37
label display, 1-36
type, 1-36
output scaling limits, 1-36
ten down, 1-36
ten up, 1-37
plot relative, 1-37
select pen, 1-37
HP-GL subset, 1-36
HP-IB
acceptor handshake, 1-2
address capability, 2-5
addresses, 2-8
analyzer capabilities, 1-2
basic listener, 1-2
basic talker, 1-2
bus structure, 2-2, 2-3
command formats, 2-11
data rate, 2-5
device clear, 1-2
device types, 2-2
message transfer scheme, 2-5
meta-messages, 2-8
multiple-controller capability, 2-5
operation, 2-2
operational capabilities, 2-5
parallel poll, 1-2
programming reference, 2-1
requirements, 2-5
serial poll, 1-2
source handshake, 1-2
status indicators, 2-6
HP-IB commands, 1-1
HP-IB interconnect cables, 3-1
HP-IB only commands, 1-29

I

<I>, 1-8
identification
of analyzer, 1-4
of firmware revision, 1-4
IDN?, 1-4, 1-50
IEEE-488 universal commands, 2-8
IEEE standard codes, formats, protocols
information, 2-2
IEEE standard digital interface information,
2-2
IP bandwidth, 1-50
IFBIGHI, 1-50
IFBIL, 1-50
IFBW[D], 1-50
IFC (abort message), 2-8
IFC (interface clear) control line, 2-4
IFCLEZESEQ<i>, 1-50
IFLSEQ<i>, 1-50
IFLFAILSEQ<i>, 1-50
IFLPASSSEQ<i>, 1-50
IM, 1-37
IMAG, 1-50
imaginairy, 1-50
increment loop counter, 1-50
INCRLOOC, 1-50
information on programs, 2-31
INID, 1-50
INIE, 1-50
initialize disk, 1-50
initializing the system, 2-32
INPUCALC<i>[D], 1-50
INPUKALK[D], 1-51
INPUDATA[D], 1-51
INPUFORM[D], 1-51
INPLEAS[D], 1-51
INPUPMCAL<i>, 1-51

Index-7
INPURAW, 1-51
input buffer, 3-7
input/output path, 3-9
input syntax, 1-5
INSMEXSA, 1-51
INSMEXSM, 1-51
INSMNETA, 1-51
INSMTUNR, 1-51
instrument setup, 2-28
instrument states, 3-40
recalling, 3-40, 3-45
saving, 3-40, 3-45
instrument state saving and restoring example program, 2-79
instrument state summary, 2-23
INTD, 1-51
INTE[D], 1-51
intensity
  background, 1-42
interface addresses, 2-8
interface clear (IFC) control line, 2-4
interface functions
  controller, 2-3
  listener, 2-3
  talker, 2-2
interpolative correction, 1-45
interrogate command, 2-14
interrogate syntax, 1-6, 2-12
interrogating commands, 1-3, 3-6
interrupt generation example program, 2-65
interrupts, generating, 3-33
INTM, 1-51
IP, 1-37
ISOD, 1-51
ISOOP, 1-51
IW, 1-37
K
key codes, 1-41
KEY[D], 1-51
key select codes, 1-8
kilohertz terminator code, 2-11
KITT, 1-52
kit done, 1-52
kits of calibration standards, 2-37
L
L4, 1-2
LABEFWDM[$], 1-52
LABEFWDT[$], 1-52
label cal kit, 1-52
label class, 1-52
label standard, 1-52
LABERES[$], 1-52
LABERESP[$], 1-52
LABEREVM[$], 1-52
LABERVEVT[$], 1-52
LABES11A[$], 1-52
LABES11B[$], 1-52
LABES11C[$], 1-52
LABES22A[$], 1-52
LABES22B[$], 1-52
LABES22C[$], 1-52
LABETLIFM[$], 1-52
LABETLFT[$], 1-52
LABETLRM[$], 1-52
LABETLRT[$], 1-52
LABETRFM[$], 1-52
LABETRRM[$], 1-52
LABETTFM[$], 1-52
LABETTFT[$], 1-52
LABETTRM[$], 1-52
LABETTRT[$], 1-52
LABK[$], 1-52
LABS[$], 1-52
LaserJet, 1-62
LB, 1-36
LEO (no extended listener capabilities), 2-5
learn string and calibration kit string, 2-23
learn string use example program, 2-74
LEFL, 1-52
LEFU, 1-52
levels of data, 2-22
LIF
  directory size, 1-47
LIF format, 1-49
LIMD[D], 1-53
LIMIAPO[D], 1-52
LIMILINE, 1-52
LIMIMAOF[D], 1-52
LIMISTIO[D], 1-53
LIMITEST, 1-53
limit line, 1-52
limit line amplitude offset, 1-52
limit lines, 3-53
  setting up, 3-53
limit line setup example program, 2-89
limit line stimulus offset, 1-53
limit-line testing, 3-48
  list-frequency table, creating, 3-48
  list-frequency table, selecting a single segment, 3-51
  performing PASS/FAIL tests, 3-53
  using list-frequency mode, 3-48
limit-line testing example program, 2-83
limit table
  edit, 1-48
  limit test, 1-53
limit-test array used to read values example program, 2-56
limit test beeper, 1-42
limit test fall, 1-50
limit test pass, 1-50
limit-test table, 3-53
creating, 3-53
transmitting, 3-53
LIML[D], 1-53
LIMM[D], 1-53
LIMS[D], 1-53
LIMITFL, 1-53
LIMTSL, 1-53
LIMTSP, 1-53
LIMU[D], 1-53
linear sweep, 1-53
line feeds, 1-5, 1-6
lines for control, 2-4
lines for handshaking, 2-4
line type
data, 1-53
memory, 1-53
LINFREQ, 1-53
LINM, 1-53
lin mag, 1-53
LINTDATA[D], 1-53
LINTMEMO[D], 1-53
LISFREQ, 1-53
list
clear, 1-44
listener interface function, 2-3
listen mode (L), 2-6
list-frequency mode, 3-48
list-frequency mode example program, 2-83
list sweep, 1-53
list values, 1-53
print, 1-61
LISV, 1-53
L (listen mode), 2-6
LOAD<>, 1-53
LOADSEQ<>1, 1-54
local command (GTL), 2-8
local lockout, 1-2, 3-5
local lockout command (LLO), 2-9
local mode, 3-5
LOGFREQ, 1-54
LOGM, 1-54
log mag, 1-54
log sweep, 1-54
LOOC[D], 1-54
loop counter
decrement, 1-46
increment, 1-50
loop counter value, 1-54
lower case, 1-5, 1-6
lower limit
segment, 1-53
low pass frequency, 1-65
low pass impulse, 1-54
low pass step, 1-54
LOWPIMPU, 1-54
LOWPSTEP, 1-54
LITA, 1-36
M
MANTRIG, 1-54
MARKCENT, 1-54
MARKCONT, 1-54
MARKCOUP, 1-54
MARKCW, 1-54
MARKDELA, 1-54
MARKDISC, 1-54
marker bandwidth search, 1-72
marker data, 2-16
marker parameters
print, 1-61
marker positioning, 3-20
by data point location, 3-20
by frequency location, 3-20
by trace-data value, 3-20
marker range, 1-54
markers
continuous, 1-54
discrete, 1-54
displayed, 1-47
markers coupled, 1-54
marker search
left, 1-65
maximum, 1-65
minimum, 1-65
off, 1-65
right, 1-65
target, 1-65
tracking, 1-70
markers off, 1-55
marker statistics, 1-55
markers uncoupled, 1-55
markers used in data transfer example
program, 2-49
marker to CW frequency, 1-54
marker to limit offset, 1-52
marker to middle
segment, 1-55
marker to stimulus
segment, 1-55
marker width, 1-72
marker zero, 1-55
MARKFAUV[D], 1-54
MARKFSTI[D], 1-54
MARKFVAL[D], 1-55
MARK<>, 1-54
MARKMIDD, 1-55
MARKMINI, 1-55
MARKOFF, 1-55
MARKREF, 1-55
MARKSPAN, 1-55
MARKSTAR, 1-55
MARKSTIM, 1-55
MARKSTOP, 1-55
MARKUNCO, 1-55
MARKZERO, 1-55
MAXF[D], 1-55
MEASA, 1-55
MEASB, 1-55
MEASR, 1-55
MEASTAT, 1-55
measurement calibration, 1-34
measurement calibration example program, 2-37, 2-41
measurement data post-processing, 2-29
measurement data taking, 2-29
measurement data transfer example program, 2-48
measurement parameters
required order, 3-10
setting, 3-10
verifying, 3-12
measurement process, 2-28
measurement process synchronization example program, 2-62
measurement restart, 1-63
measurement setup, 2-32, 3-10
measurement specifications, 3-25
group delay, 3-25
magnitude, 3-25
phase, 3-25
megahertz terminator code, 2-11
memory channel 1
color, 1-59
memory channel 2
color, 1-59
memory requirements, 3-1
MENU, 1-55
MENUAVG, 1-55
MENUCAL, 1-55
MENUCOPY, 1-55
MENUDISP, 1-55
MENUFORM, 1-55
MENUMARK, 1-55
MENUMEAS, 1-55
MENUMRKF, 1-55
MENURECA, 1-55
MENUSCAL, 1-55
MENUSEQU, 1-55
MENUSTIM, 1-55
MENUSYST, 1-56
message transfer scheme, 2-5
meta-messages, 2-8
methods of HP-IB operation, 2-2
microseconds terminator code, 2-11
middle value
segment, 1-53
milliseconds terminator code, 2-11
MINF[D], 1-56
modes
analyzer bus, 2-7
debug, 2-31
pass-control, 2-7
system-controller, 2-6
talker/listener, 2-7
modes for bus device, 2-6
MODI1, 1-56
modify cal kit, 1-56
modify colors, 1-45
modify sequence, 1-56
multiple-controller capability, 2-5
N
naming conventions, 2-10
nanoseconds terminator code, 2-11
network analyzer mode, 1-51
network analyzer system setups example program, 2-74
NEWS[EQ<i>], 1-56
new sequence, 1-56
NEXP, 1-56
next page, 1-56
no extended talker capabilities (TEO), 2-5
NOOP, 1-56
number of HP-IB devices allowed, 2-2
number of listeners allowed, 2-3
number of readings, 1-56
NUMG[D], 1-56
NUMR[D], 1-56
O
OC, 1-37
OE, 1-37
OFSD[D], 1-56
OFSL[D], 1-56
OFSSZ[D], 1-56
OL, 1-37
OMII, 1-56
ON TIMEOUT, 3-7
OP, 1-36
OPC, 1-3, 1-56
OPC-compatible commands, 1-3, 2-13
open capacitance values, 1-43
OPEP, 1-56
operating parameters, 1-56
operational capabilities for HP-IB, 2-5
operation complete, 1-3, 2-13
operation complete commands, 3-7
operation of analyzer, 2-13
operation of HP-IB, 2-2
operation using talker/listener mode example program, 2-96
OS, 1-37
OUTPCAL<1>, 1-56
OUTPCALK, 1-57
OUTPDATA, 1-57
OUTPERRO, 1-57
OUTPFORM, 1-57
OUTPCAL<1>, 1-57
OUTPilton, 1-57
OUTPIMC<1>, 1-57
OUTPKEY, 1-57
OUTPLEAS, 1-58
OUTPLIMF, 1-58
OUTPLIML, 1-58
OUTPLIMM, 1-58
OUTPMARK, 1-58
OUTPMEMO, 1-58
OUTPMSTA, 1-58
OUTPMWID, 1-58
OUTPMWIL, 1-58
OUTPLOT, 1-58
OUTPPMCAL<1>, 1-58
OUTPRIN, 1-58
OUTPRRALL, 1-58
OUTPRAW<1>, 1-59
OUTPFRFR, 1-59
OUTPSEQ<1>, 1-59
OUTPSSTAT, 1-59
OUTPTTL, 1-59
output
plot string, 1-58
output-data command, 2-14
output of errors, 2-27
output queue, 1-4, 2-14
output syntax, 2-15
outputting trace-related data, 2-16

P
PaintJet, 1-62
PARAIN[D], 1-59
PARAL, 1-59
parallel poll, 1-2
parallel poll configure, 2-9
parallel poll non response (PPO), 2-5
parallel port configure, 1-59
parameter setting example program, 2-32
parameter verification example program, 2-32
PARAOUT[D], 1-59
pass control, 1-71
pass control capabilities (C10), 2-5
pass-control mode, 2-7
pass control mode, 2-9
pass-control mode use in peripheral control example program, 2-98
PASS/FAIL tests, 3-56
PASS/FAIL tests example program, 2-93
PAUS, 1-59
pause, 1-59
pause to select sequence, 1-62
PAx,y, 1-36
PCB[D], 1-59
PC-graphics applications example program, 2-105
PCOLDATA1, 1-59
PCOLDATA2, 1-59
PCOLGRAT, 1-59
PCOLMEMO1, 1-59
PCOLMEMO2, 1-59
PCOLTEXT, 1-59
PCOLWARN, 1-59
PD, 1-36
PDATA, 1-59
PENNDATA[D], 1-59
PENNGRAT[D], 1-59
PENNMARK[D], 1-59
PENNMEMO[D], 1-59
PENNTEXT[D], 1-59
pen number
data, 1-59
graticule, 1-59
markers, 1-59
memory, 1-59
text, 1-59
performing PASS/FAIL tests while tuning example program, 2-93
peripheral
address, 1-42
peripheral addresses, 2-8
peripheral control using pass-control example program, 2-98
PG, 1-37
PGRAT, 1-60
PHAO[D], 1-60
PHIAS, 1-60
phase, 1-60
phase offset, 1-60
picoseconds terminator code, 2-11
PLO, 1-60
PLOT, 1-60
plot data, 1-59
plot file and PC-graphics example program, 2-105
plot graticule, 1-60
plot markers, 1-60
plot memory, 1-60
plot quadrant, 1-52, 1-64
plot scale, 1-64
plot softkeys, 1-62
plot speed, 1-60
plot string
  output, 1-58
plotter
  address, 1-42
  auto feed, 1-60
  baud rate, 1-60
  disk, 1-60
  form feed, 1-60
  handshake, 1-60
plotter default setup, 1-47
plotter port
  HP-IB, 1-60
  parallel, 1-60
  serial, 1-60
plotter type, 1-60
plot text, 1-62
plotting
  to a file, 3-65, 3-67
plotting plot files stored on disk example
  program, 2-106, 3-68
plotting, remote, 3-59, 3-61
plotting to a file and transferring the file data
  to a plotter example program, 2-103
PLTHNDSHK, 1-60
PLTPRDISK, 1-60
PLTPRTHPIB, 1-60
PLTPRTPARA, 1-60
PLTPRTSERI, 1-60
PLTTRAUTF, 1-60
PLTTRBAUD[D], 1-60
PLTTRFORF, 1-60
PLTTYPHPLG, 1-60
PLTTYPPLTR, 1-60
plus sign, 1-6
PMEM, 1-60
PMKR, 1-60
PMTRTTIT, 1-60
POIN[D], 1-60
points
  specify, 1-60
POLA, 1-60
polar, 1-60
polar markers, 1-60
POLMLN, 1-60
POLMLOG, 1-60
POLMR, 1-60
PORC, 1-60
PORT 1 attenuator, 1-42
PORT1[D], 1-60
PORT 2 attenuator, 1-42
PORT2[D], 1-60
PORTA[D], 1-60
PORTB[D], 1-60
port extensions, 1-60
PORTP, 1-60
port power coupling, 1-60
PORTR, 1-61
PORTT, 1-61
post-processing the measurement data, 2-29
POWE[D], 1-61
power level, 1-61
power loss range
  edit, 1-61
power loss table, 1-62
  edit, 1-61
power meter
  address, 1-42
power meter cal factor, 1-43
power meter calibration, 1-62, 3-36
power meter calibration example program, 2-69
power meter into title string, 1-60
power meter type, 1-61
power ranges, 1-61
power slope, 1-66
power sweep, 1-61
power trip, 1-61
POWLFRQ[D], 1-61
POWLLIST, 1-61
POWLLOSS[D], 1-61
POWM, 1-61
POWS, 1-61
POWT, 1-61
PPO, 1-2
PPO (does not respond to parallel poll, 2-5
PRAN, 1-61
preparing for remote operation, 3-8
presetting the instrument, 3-8
PRIC, 1-61
PRINALL, 1-61
PRINSEQ<i>, 1-61
PRINTALL, 1-61
print color, 1-61
printer
  address, 1-42
  auto feed, 1-62
  baud rate, 1-62
  form feed, 1-62
  handshake, 1-61
printer default setup, 1-46
printer port
  HP-IB, 1-61
  parallel, 1-62
  serial, 1-62
printing
  using the serial port, 3-63
printing plot files stored on disk example
program, 2-106, 3-68
printing, remote, 3-59, 3-61
printing with the serial port example program,
2-101
print monochrome, 1-61
print sequence, 1-61
print softkeys, 1-62
PRIS, 1-61
PRNHDNSHK, 1-61
PRNPRTHPIB, 1-61
PRNPRTPARA, 1-62
PRNPRTSERI, 1-62
PRNTRAUTP, 1-62
PRNTRBAUD[D], 1-62
PRNTRFORF, 1-62
PRNTPDJI, 1-62
PRNTYPEP, 1-62
PRNTYPPLJ, 1-62
PRNTYPPJ, 1-62
PRNTYPTPJ, 1-62
processing after taking measurement data,
2-29
processing data chain, 2-21
process of measuring, 2-28
process synchronization example program,
2-62
program debugging, 2-31
program development features, 2-31
program example
1. measurement setup, 2-32
   controlling peripherals using pass-control
mode, 2-98
data transfer using floating-point numbers,
2-54
data transfer using form 1, 2-60
data transfer using form 4 (ASCII transfer),
2-51
data transfer using frequency array
information, 2-56
data transfer using markers, 2-49
full 2-port measurement calibration, 2-41
generating interrupts, 2-65
limit-line testing, 2-83
measurement calibration, 2-37
measurement data transfer, 2-48
measurement process synchronization,
2-62
network analyzer system setups, 2-74
operation using talker/listener mode, 2-96
performing PASS/FAIL tests while tuning,
2-93
plotting plot files stored on disk, 2-106,
3-68
plotting to a file and transferring the file
data to a plotter, 2-103
power meter calibration, 2-69
printing plot files stored on disk, 2-106,
3-68
printing with the serial port, 2-101
reading ASCII disk files to the instrument
controller's disk file, 2-108
reading calibration data, 2-76
report generation, 2-86
S11 1-port measurement calibration, 2-37
saving and restoring the analyzer
instrument state, 2-79
selecting a single segment from a table of
segments, 2-86
setting parameters, 2-32
setting up limit lines, 2-89
storing and recalling instrument states,
2-74
system initialization, 2-32
using the error queue, 2-63
using the learn string, 2-74
verifying parameters, 2-35
program information, 2-31
programming examples, 2-30
PRx.y, 1-37
PSOFT, 1-62
PTEXT, 1-62
PTOS, 1-62
PU, 1-37
purge file, 1-62
PURG<>, 1-62
PWMCEACS[D], 1-62
PWMCOFF[D], 1-62
PWMCOMES[D], 1-62
PWRLSS, 1-62
PWRMCAL[D], 1-62
PWRRL, 1-62
Q
quasi 2-port cal, 1-45
query, 1-3
question mark
using, 1-3
queue for output, 2-14
R
RAID, 1-62
RAISOL, 1-62
RAIRESP, 1-63
raw data
   include with disk files, 1-48
raw measured data, 2-21
reading analyzer data, 2-14
reading calibration data example program, 2-76
REAL, 1-63
RECA<1>, 1-63
recall colors, 1-63
recalling and storing example program, 2-74
recall register, 1-63
recall sequence, 1-54
RECAREG<1>, 1-63
RECO, 1-63
recommended disk drives, 3-2
recommended plotters, 3-2
recommended printers, 3-2
REFD, 1-63
reference line value, 1-63
reference of HP-IB programming, 2-1
reference position, 1-63
set to mkr, 1-55
REFL, 1-63
reflection, 1-44
REFOP, 1-63
REFP[D], 1-63
REFT, 1-63
REFV[D], 1-63
remote enable (REN) control line, 2-4
remote/local capability (RL1), 2-5
remote lockout, 1-2
remote mode, 2-9, 3-5
remote operation (R), 2-6
REN (remote enable) control line, 2-4
report generation, 3-59
report generation example generation, 2-96
reporting of errors, 2-24
reporting on status, 2-24
reporting status, 2-62
RESC, 1-63
RESID, 1-63
reset color, 1-64
RESPDONE, 1-63
response cal done, 1-63
REST, 1-63
restart averaging, 1-42
restore display, 1-63
restoring and saving the analyzer instrument
state example program, 2-79
resume cal sequence, 1-63
REV, 1-63
REV, 1-63
REVVM, 1-63
REV, 1-63
RFGTLO, 1-64
RF < LO, 1-64
RF > LO, 1-64
RFLTLO, 1-64
RIGL, 1-64
RIGU, 1-64

RL1, 1-2
RL1 (complete remote/local capability), 2-5
routing debugging, 2-31
R (remote operation), 2-6
RS, 1-37
RSCO, 1-64
RST, 1-64
rules for code naming, 2-10

S
S11, 1-64
S11, 1-port measurement calibration example
program, 2-37
S12, 1-64
S21, 1-64
S22, 1-64
SADD, 1-64
SAV1, 1-64
SAV2, 1-64
SAVC, 1-64
save cal kit, 1-64
save colors, 1-68
save format, 1-64
SAVE<1>, 1-64
SAVEREG<1>, 1-64
save register, 1-64
save sequence, 1-68
SAVEUSEK, 1-64
saving and restoring the analyzer instrument
state example program, 2-79
SAVUSCI, 1-64
SAVUBINA, 1-64
SCAL[D], 1-64
scale
auto, 1-42
SCAP, 1-64
SDEL, 1-64
SDON, 1-65
SEAL, 1-65
SEAMAX, 1-65
SEAMIN, 1-65
SEAOFF, 1-65
SEAR, 1-65
SEATARG[D], 1-65
second harmonic, 1-50
seconds terminator code, 2-11
SEDU[D], 1-65
segment
add, 1-64
delete, 1-64
edit, 1-65
segment edit done, 1-48
segment select, 1-67
segment selection from a table of segments
example program, 2-86
selecting a segment from a table of segments example program, 2-86
select sequence, 1-65
select standard, 1-67
semicolon, 1-6
sensor input selection, 1-71
SEQ<1>, 1-65
sequence wait, 1-65
SEQWAIT[D], 1-65
serial poll, 1-2, 2-9, 3-31
serial port interface for printing example program, 2-101
service request, 3-33
service request asserted by the analyzer (S), 2-6
service request (SRQ) control line, 2-4
set bandwidth, 1-50
SETBIT[D], 1-65
SETDATE[8], 1-65
SETF, 1-65
SETTIME[8], 1-65
setting addresses, 3-2
setting HP-IB addresses, 2-8
setting parameters example program, 2-32
setting the control mode, 3-2
setting up limit lines example program, 2-89
setting up the instrument, 2-28
setting up the system, 3-2
SETZ[D], 1-65
SH1, 1-2
SH1 (full-source handshake), 2-5
SHOM, 1-65
show menus, 1-65
SLh.w, 1-37
SING, 1-65
single bus concept, 2-6
single point type, 1-53
SL, 1-37
SLID, 1-66
sliding load, 1-66
done, 1-66
set, 1-66
SLIL, 1-66
SLIS, 1-66
SLOPE[D], 1-66
sloping line type, 1-53
SMIC, 1-66
SMIMGB, 1-66
SMIMLIN, 1-66
SMIMLOG, 1-66
SMIMRI, 1-66
SMIMRX, 1-66
Smith chart, 1-66
Smith markers, 1-66
SMOAPER[D], 1-66
SMOOG, 1-66
smoothing, 1-66
smoothing aperture, 1-66
SOFR, 1-66
SOFT<1>, 1-66
SOUP, 1-66
source handshake, 1-2
source power on/off, 1-66
spaces, 1-6
in commands, 1-5
SPAN[D], 1-66
S-parameters, 1-64
SPECFWDM[T], 1-66
SPECFWDM[T], 1-66
specify class, 1-66
specify gate menu, 1-66
specify points, 1-60
SPECRESI[T], 1-66
SPECRESP[T], 1-66
SPECREV[M[T], 1-67
SPECREV[T], 1-67
SPEC11A[T], 1-67
SPEC11B[T], 1-67
SPEC11C[T], 1-67
SPEC22A[T], 1-67
SPEC22B[T], 1-67
SPEC22C[T], 1-67
SPECTLFT[T], 1-67
SPECTRM[T], 1-67
SPECULRT[T], 1-67
SPECURM[T], 1-67
SPECUTFM[T], 1-67
SPECUTFT[T], 1-67
SPECUTTRM[T], 1-67
SPECUTTR[T], 1-67
SPECUTART[T], 1-67
SPECUTART[T], 1-67
SPGD, 1-66
SPLD, 1-67
split display, 1-67
SPn, 1-37
SR, 1-37
SR1, 1-2
SR1 (complete service request capabilities), 2-5
SRQ (service request) control line, 2-4
SEEG[D], 1-67
S (service request asserted by the analyzer), 2-6
STANA, 1-67
STANB, 1-67
STANC, 1-67
STAND, 1-67
standard defined, 1-68
standard definition, 1-46
standard labelling, 1-52
standard offsets, 1-56
standard type, 1-68
STANE, 1-67
STANF, 1-67
STANG, 1-67
STAR[D], 1-67
statistics
  marker, 1-55
status bit definitions, 2-24
status byte, 2-24, 2-26, 3-31
status indicators, 2-6
status reporting, 2-24, 2-62, 3-31
STB?, 1-67
STDD, 1-68
STDTRAI, 1-68
STDTDELA, 1-68
STDLOAD, 1-68
STDTOPEN, 1-68
STDTHOR, 1-68
step 1 of a measurement, 2-28
step 2 of a measurement, 2-28
step 3 of a measurement, 2-28
step 4 of a measurement, 2-29
step 5 of a measurement, 2-29
step 6 of a measurement, 2-29
step down, 1-48
step up, 1-71
stimulus value
  segment, 1-53
STOP[D], 1-68
storage
  disk, 1-48, 1-51
  internal memory, 1-51
store to disk, 1-68
STOR<>1>, 1-68
storing and recalling instrument states
  example program, 2-73
STORSEQ<>, 1-68
STPSEQ[D], 1-68
string for calibration kit, 2-23
structure
  of commands, 1-5
structure of command syntax, 2-11
structure of HP-IB bus, 2-3
structure of status reporting, 2-24
SVCO, 1-68
sweep user-controlled, 2-31
SWET[D], 1-68
SWR, 1-68
synchronization, 3-31
synchronization of process example program, 2-62
syntax for commands, 2-10
syntax for output, 2-15
syntax structure, 2-11
syntax types, 1-5, 1-6, 2-12
system controller capabilities (C1,C2,C3), 2-5
system-controller mode, 2-6, 2-7
system initialization, 2-32
system setups, 3-40
  reading calibration data, 3-42
  using the learn string, 3-40
system setups example program, 2-74
T
T6, 1-2
T6 (basic talker), 2-5
TAKCS, 1-68
take cal sweep, 1-68
take-control command, 2-9
taking the measurement data, 2-29
talker interface function, 2-2
talker/listener, 1-69
talker/listener mode, 2-7
talker/listener mode operation example program, 2-96
TALKLIST, 1-69
talk mode (T), 2-6
TE0 (no extended talker capabilities), 2-5
TERI[D], 1-69
terminal impedance, 1-69
terminators, 1-5
terminators and units, 2-11
TESS7, 1-69
testing with limit lines example program, 2-83
test port return cables, 3-2
test port selection, 1-71
test set switching, 1-45
test setup calibration, 2-28
tests that PASS/FAIL example program, 2-93
text
  color, 1-59
ThinkJet, 1-62
third harmonic, 1-50
TIMDTRAN, 1-69
time, 1-65
time domain bandpass, 1-42
time domain gate, 1-49
time specify, 1-68
TIMESTAMP, 1-69
time stamp, 1-69
TINT, 1-69
TITFO, 1-69
TITF<>>, 1-69
TITF<<>, 1-69
title
  CRT, 1-69
title disk file, 1-69
title register, 1-69
title sequence, 1-69
title string to trace memory, 1-69
title to peripheral, 1-69
title to printer, 1-69
TTTP, 1-69
TTTREG<1>, 1-69
TTTR<1>, 1-69
TTTSEQ<1>(8], 1-69
TTTMEM, 1-69
TTTPERI, 1-69
TTTPRIN, 1-69
trace-data formats and transfers, 2-48
trace-data transfers, 2-19
trace memory, 2-21
trace-related data, 2-16
TRACK, 1-70
TRAD, 1-70
TRAN, 1-70
transfer of data, 2-4
transfer of data example program, 2-48
transfer of data using floating-point numbers
example program, 2-54
transfer of data using form 11 example
program, 2-60
transfer of data using form 4 example
program, 2-51
transfer of data using frequency array
information example program, 2-56
transferring file data to a plotter example
program, 2-103
transferring plot-file data to a plotter example
program, 2-106, 3-68
transferring plot-file data to a printer example
program, 2-106, 3-68
transferring the measurement data, 2-29
transfers and formats of trace-data, 2-48
transfers of trace-data, 2-19
transform, 1-69
TRAOP, 1-70
TRIG, 1-70
trigger
continuous, 1-45
external, 1-48
hold, 1-50
number of groups, 1-56
single, 1-65
trigger device, 2-9
tri-state drivers, 1-2
tri-state drivers (E2), 2-5
troubleshooting, 3-3, 3-5
TST?, 1-70
TSTIOFW(1), 1-70
TSTIOREV(1), 1-71
TSTP, 1-71
T (talk mode), 2-6
TTLHPULS, 1-71
TTLLPULS, 1-71
TTL0H, 1-71
TTL0L, 1-71
TTL out high, 1-71
TTL out low, 1-71
tuned receiver mode, 1-51
types of syntax, 2-12
U
UNCONV, 1-71
units, 1-4
units and terminators, 2-11
units as a function of display format, 2-16
universal commands, 2-8
UP, 1-71
up converter, 1-71
upper case, 1-5, 1-6
upper limit
segment, 1-53
USEPASC, 1-71
user-controllable sweep, 2-31
user-defined cal kits, 1-43
user-defined kit
save, 1-64
user graphics
include with disk files, 1-48
USESENSA, 1-71
USESENSB, 1-71
use sensor A, 1-71
use sensor B, 1-71
V
valid characters, 2-11
velocity factor, 1-71
VELOFACT(D), 1-71
verifying HP-IB operation, 3-2
verifying parameters example program, 2-35
VIEM, 1-71
view measurement, 1-71
VOFF(D), 1-71
volts terminator code, 2-11
W
WAIT, 1-71
waiting-for-group-execute-trigger, 2-9
waiting-for-reverse-get bit, 2-9
warning
color, 1-59
warning beeper, 1-42
WAVE, 1-72
WIDT, 1-72
WIDV(D), 1-72
WINDMAXI, 1-72

Index-17
WINDMINI, 1-72
WINDNORM, 1-72
window
  maximum, 1-72
  minimum, 1-72
  normal, 1-72
  shape, 1-72
  value, 1-72
WINDOW[D], 1-72

WINDUSEM, 1-72
WRSK<1>[‰], 1-72

X
Xon, 1-61

Z
Z0, 1-65